Gaussian Splatting SLAM: Real-Time Dense 3D Reconstruction with Photorealistic Rendering

Babar Hussain
School of Information and Software Engineering
University of Electronic Science and Technology of China
Jianshe North Road, 610054, Chengdu Sichuan, China

Guo Jiandong
School of Information and Software Engineering
University of Electronic Science and Technology of China
Jianshe North Road, 610054, Chengdu Sichuan, China

Sidra Fareed
School of Information and Software Engineering
University of Electronic Science and Technology of China
Jianshe North Road, 610054, Chengdu Sichuan, China

Bohuan Fang
School of Information and Software Engineering
University of Electronic Science and Technology of China
Jianshe North Road, 610054, Chengdu Sichuan, China

Ma Qianhao
School of Information and Software Engineering
University of Electronic Science and Technology of China
Jianshe North Road, 610054, Chengdu Sichuan, China

Subhan Uddin
School of Information and Software Engineering
University of Electronic Science and Technology of China
Jianshe North Road, 610054, Chengdu Sichuan, China

ABSTRACT

This paper introduces Gaussian Splatting SLAM, a novel real-time Simultaneous Localization and Mapping (SLAM) system that leverages 3D Gaussian Splatting (3DGS) as its core representation, seamlessly integrating mapping, tracking, and rendering into a single framework. Unlike traditional SLAM approaches that rely on sparse features, voxel grids, or neural fields, the proposed method achieves dense, photorealistic 3D reconstruction while maintaining real-time performance and computational efficiency. To enable robust and accurate camera tracking, an analytical Jacobian for camera pose optimization on the Lie group is derived, allowing direct alignment of camera poses with the 3D Gaussian map, which ensures fast convergence and resilience against initial pose errors. Additionally, isotropic regularization is introduced, a novel geometric constraint that prevents over-elongation of Gaussians, thereby enhancing structural consistency in incremental reconstruction, particularly in textureless and ambiguous regions. By leveraging a differentiable rasterization pipeline, the proposed method achieves real-time rendering speeds of up to 769 FPS, significantly outperforming neural field-based techniques that rely on expensive ray marching. The efficiency of the system enables its application in robotics, augmented reality, and spatial AI, where real-time, high-fidelity 3D reconstruction is critical. Gaussian Splatting SLAM is evaluated on both monocular and RGB-D datasets, demonstrating state-of-the-art performance in trajectory estimation, reconstruction accuracy, and novel view synthesis, while also showcasing its robustness in challenging environments involving dynamic objects, transparent surfaces, and low-texture regions. Compared to existing SLAM systems, the proposed approach offers a unique balance between computational efficiency, geometric precision, and rendering quality, making it an ideal solution for real-time applications requiring dense, high-fidelity 3D scene understanding. The results highlight its potential to transform real-time 3D perception, setting a new benchmark in dense SLAM and real-time mapping, while opening new avenues for research in adaptive scene representations and interactive 3D reconstruction technologies.

General Terms

Experimentation, Algorithms, Performance

Keywords

Gaussian Splatting; 3D reconstruction; SLAM; Isotropic regularization; Real-time rendering; Computer Vision

1. INTRODUCTION

The ability to achieve real-time, photorealistic 3D reconstruction using a moving camera has been a long-standing goal in robotics, augmented reality (AR), and spatial AI. From autonomous systems that require precise environmental understanding to AR applications needing seamless integration of virtual and real-world elements, the demand for high-fidelity, real-time 3D scene reconstruction continues to grow. However, despite significant advancements, achieving dense, high-resolution 3D mapping while maintaining real-time performance remains a challenging problem. Traditional Simultaneous Localization and Mapping (SLAM) systems have evolved considerably, with approaches ranging from sparse feature-based methods to dense volumetric mapping techniques. Sparse SLAM techniques, such as ORB-SLAM [17] and VINS-Mono, provide reliable camera tracking but lack detailed scene geometry reconstruction. On the other hand, dense SLAM methods, including KinectFusion [6] and ElasticFusion [25], achieve higher reconstruction accuracy but suffer from high memory consumption, computational inefficiencies, and resolution limitations.

Beyond these conventional approaches, recent advances in neural implicit representations [29], such as Neural Radiance Fields (NeRF) [15] and Neural SLAM frameworks, have demonstrated significant improvements in 3D scene modeling and novel view synthesis. These methods leverage differentiable volumetric rendering to achieve unprecedented visual fidelity. However, their reliance on computationally expensive ray marching and extensive neural network training makes them impractical for real-time applications, particularly in robotics and AR where immediate interaction with the 3D environment is essential. Furthermore, neural implicit methods require significant memory and computational power, making them unsuitable for deployment on edge devices and resource-constrained platforms. These limitations highlight the need for a computationally efficient, memory-friendly, and real-time-capable SLAM framework that achieves dense, high-fidelity 3D reconstruction without sacrificing performance [23] [27]. This paper introduces Gaussian Splatting SLAM, a novel approach that unifies mapping, tracking, and rendering into a single, efficient framework by leveraging 3D Gaussian Splatting (3DGS) as its core representation. Unlike voxel-based or neural field-based SLAM techniques, 3DGS models a scene as a collection of anisotropic Gaussians, each defined by its position, covariance (shape and orientation), color, and opacity. This representation enables continuous, differentiable, and memory-efficient scene modeling, making it particularly well-suited for real-time, high-fidelity 3D reconstruction. Unlike neural fields that rely on ray marching, 3D Gaussian Splatting supports differentiable rasterization, allowing significantly faster rendering while maintaining photorealistic quality. The proposed Gaussian-based approach eliminates the need for traditional discrete volumetric grids, reducing memory consumption while preserving geometric details. By integrating mapping, tracking, and rendering into a single unified pipeline, Gaussian Splatting SLAM achieves unparalleled real-time performance, photorealistic scene reconstruction, and high-accuracy camera tracking.

A key technical advancement in this work is the derivation of an analytical Jacobian for camera pose optimization on the Lie group, which enables fast and accurate tracking, even in challenging monocular SLAM settings. This direct alignment of camera poses with the 3D Gaussian map allows for robust localization and significantly improves convergence in cases where traditional feature-based methods struggle. Additionally, isotropic regularization is introduced, a novel technique that prevents anisotropic Gaussian elongation, ensuring geometric consistency in incremental reconstruction. This regularization is particularly crucial for handling textureless regions, reflective surfaces, and transparent objects, which are traditionally difficult for SLAM systems to model accurately. To further enhance robustness,A dynamic Gaussian management system is developed, which intelligently inserts and prunes Gaussians based on visibility, motion stability, and geometric constraints, ensuring that the map remains clean, adaptive, and consistent over time. Furthermore, the differentiable rasterization pipeline achieves ultra-fast rendering at up to 769 FPS, far surpassing neural rendering-based SLAM approaches that rely on computationally expensive ray marching. The combination of high-speed rendering, accurate camera tracking, and geometric regularization allows Gaussian Splatting SLAM to maintain a delicate balance between accuracy, efficiency, and photorealistic quality, setting a new benchmark in dense SLAM systems.

The proposed system evaluate Gaussian Splatting SLAM on monocular and RGB-D datasets, demonstrating its ability to achieve state-of-the-art trajectory estimation, high-fidelity 3D reconstructions, and robust performance in textureless and transparent object handling. Unlike prior SLAM approaches that struggle in challenging conditions, this system excels in low-texture environments, occluded scenes, and dynamic scenarios. The real-world applicability of proposed method is further validated through experiments on self-captured datasets, where the system successfully reconstruct complex objects such as transparent glass surfaces and crinkled textures, which pose significant challenges for conventional SLAM pipelines. The results of this work confirm that Gaussian Splatting SLAM offers a superior balance between efficiency, accuracy, and photorealistic rendering, making it a compelling solution for robotics, AR, autonomous navigation, and real-time 3D perception.

2. RELATED WORK

This section reviews foundational work in dense SLAM, neural scene representations, and differentiable rendering, leading to a critical analysis of their limitations and motivating the proposed Gaussian Splatting SLAM framework. While each domain has contributed significantly to visual localization and mapping, none fully satisfy the combined requirements of dense geometry, photorealism, and real-time performance in dynamic or large-scale environments.

2.1 Dense SLAM Systems

Dense SLAM has evolved significantly from early systems like KinectFusion [10][12], which introduced volumetric TSDF fusion for real-time RGB-D reconstruction. Subsequent frameworks such as ElasticFusion and BundleFusion improved loop closure, map deformation, and large-scale reconstruction. These approaches operate efficiently on GPU-accelerated pipelines and produce high-fidelity reconstructions, but often rely on dense depth inputs and are limited in scalability due to their reliance on explicit volumetric storage.

Extensions like DynaSLAM and Co-Fusion introduced dynamic object handling and semantic awareness, enabling more robust mapping in changing environments. However, these methods typically depend on pre-trained segmentation models and handcrafted heuristics for dynamic-object masking, which can limit generalization and introduce latency. Despite their strengths, traditional dense SLAM systems struggle to simultaneously offer scalability, photorealism, and adaptability to diverse scenes particularly in monocular or resource-constrained setups.

2.2 Neural Scene Representations and SLAM Integration

Neural implicit representations such as Neural Radiance Fields (NeRF) [14] and its derivatives have transformed novel view synthesis and scene reconstruction. [3] Approaches like Mip-NeRF, KiloNeRF [18], and Instant-NGP [16] have accelerated training and inference using hash-based encoding and hierarchical sampling. These advances enable high-quality scene rendering with compact memory footprints, but often require extensive per-scene optimization and are inherently slow due to volumetric ray marching. Efforts to integrate neural representations into SLAM pipelines include NICE-SLAM [34], which jointly optimizes geometry and appearance using voxel-based neural fields, and Vox-Fusion [28], which supports online RGB-D SLAM with neural volumetric mapping. While these systems achieve impressive reconstructions, they remain computationally intensive and unsuitable for real-time operation on most hardware platforms.

Critically, most NeRF-based SLAM methods are tightly coupled to ray-based rendering, which limits their scalability and temporal responsiveness, especially in scenarios requiring interactive or low-latency feedback.

2.3 Differentiable Rendering for SLAM

Differentiable rendering enables end-to-end optimization [11] by making the rendering process itself gradient-aware. Early methods employed ray tracing with autograd support, allowing for gradient flow from image-space errors back to 3D scene parameters [33]. While powerful, such approaches are computationally expensive, especially when deployed at high resolution or frame rates.

Recent works have explored rasterization-based differentiable rendering, which projects scene elements (e.g., Gaussians, point splats) directly to the image plane without ray marching. This technique drastically improves performance while preserving differentiability. It has been used in methods like Gaussian Splatting for Real-Time Radiance Fields [8], which achieves real-time novel view synthesis by modeling scenes as collections of 3D Gaussians with color, opacity, and covariance. However, most differentiable renders have yet to be effectively combined with SLAM pipelines particularly for online tracking, mapping, and optimization. The gap between rendering quality and SLAM usability remains wide due to performance constraints and the lack of geometric regularization mechanisms.

2.4 Limitations in Existing SLAM Frameworks

Despite significant advancements in both traditional and neural SLAM systems, several persistent limitations [7] hinder their deployment in real-time, high-fidelity applications [32]. Many existing methods suffer from computational bottlenecks, primarily due to their reliance on ray tracing or dense voxel grids. These operations are inherently slow and often impractical for real-time tracking and mapping, especially on resource-constrained platforms. Additionally, current systems frequently lack geometric control mechanisms, leading to elongated, unstable, or inconsistent reconstructions particularly in monocular settings where depth ambiguity is high.

Memory inefficiency is another critical issue. Neural volumetric approaches typically require large-scale voxel storage or scene-specific encoding, which limit scalability when operating in expansive or long-term environments. Moreover, the architectural rigidity of many frameworks restricts their adaptability across sensor modalities (such as monocular versus RGB-D input) and scene types (e.g., dynamic versus static environments, indoor versus outdoor settings). Most importantly, a large number of neural SLAM systems prioritize photorealistic rendering at the expense of geometric fidelity. As a result, these methods often produce visually pleasing outputs that are poorly aligned with ground truth geometry, leading to inaccurate reconstructions and unstable tracking. These limitations highlight the urgent need for a SLAM framework that can simultaneously achieve photorealism, geometric accuracy, real-time performance, and scalability.

As shown in Figure 1, existing neural SLAM and rendering methods suffer from low frame rates due to computationally intensive ray marching or volumetric inference, whereas our approach achieves real-time rendering speeds exceeding 700 FPS through rasterization.

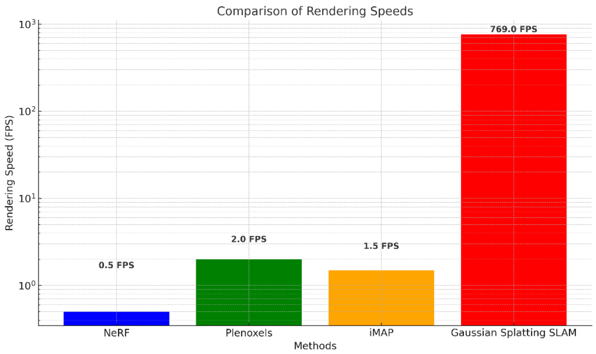

Fig. 1: Comparison of Rendering Speeds.

2.5 Contribution in Context

To address the aforementioned limitations,This work introduces Gaussian Splatting SLAM, a novel and efficient framework that bridges the gap between high-quality neural rendering and real-time dense SLAM. The proposed approach leverages 3D Gaussian Splatting a continuous, compact, and differentiable scene representation in combination with rasterization-based differentiable rendering, enabling real-time image synthesis and optimization without the computational burden of ray marching. This foundation allows for fast, accurate scene reconstruction while preserving photorealistic detail and supporting gradient-based learning.

The framework is built on three core innovations that differentiate it from prior work. It employs an end-to-end differentiable camera pose optimization using analytical Jacobians, which enables precise and smooth pose updates in both monocular and RGB-D settings. Furthermore, the system incorporates isotropic geometric regularization to prevent Gaussian over-elongation and ensure structural consistency during mapping. Finally, it features a dynamic Gaussian management strategy that adaptively adds or prunes Gaussians based on stability and visibility metrics, thereby maintaining a clean and efficient map over time. These components are unified within a single SLAM pipeline that operates at real-time speeds.

By avoiding costly ray integration and instead utilizing fast rasterization, this system offers a practical solution for dense visual SLAM in robotics, augmented reality, and spatial AI. It supports diverse environments, scales efficiently, and achieves a compelling balance between visual quality, geometric precision, and computational efficiency setting a new standard for modern SLAM systems.

3. METHODOLOGY

3.1 Overview of Gaussian Splatting SLAM

Gaussian Splatting SLAM is a novel, real-time dense Simultaneous Localization and Mapping (SLAM) framework that seamlessly integrates tracking, mapping, and rendering into a unified pipeline. At its core, the system employs 3D Gaussian Splatting (3DGS) as a continuous and differentiable scene representation, modeling the environment as a collection of anisotropic Gaussians. Each Gaussian encodes key attributes such as spatial position, geometric shape[2] [31] (through a covariance matrix), color, and opacity. This continuous formulation enables a highly expressive yet memory-efficient structure, capable of capturing both fine surface details and global scene layout. Unlike traditional SLAM systems that rely on discrete point clouds, voxel grids, or neural radiance fields, 3DGS allows for dense, photorealistic 3D reconstructions at significantly higher computational speeds, making it ideal for real-time applications. The primary motivation behind this work is to address the limitations of existing dense SLAM methods, which often struggle with trade-offs between accuracy, speed, and memory consumption. The proposed system is designed to support real-time interaction and deployment on platforms used in robotics, augmented reality (AR), and spatial artificial intelligence (AI), where computational efficiency and visual fidelity are both critical.

To achieve this, Gaussian Splatting SLAM introduces a robust methodology built upon five key technical components. First, the system represents the scene using 3D Gaussians that encapsulate both geometry and appearance in a compact form. Second, it employs a differentiable rendering process based on rasterization, which avoids the costly ray marching procedures found in neural rendering methods and enables ultra-fast image synthesis. Third, it features an efficient camera pose optimization mechanism leveraging an analytical Jacobian on the Lie group SE(3), allowing for accurate and stable tracking. Fourth, the framework incorporates isotropic regularization to ensure structural consistency by penalizing over-elongated or distorted Gaussian shapes, especially in challenging areas like textureless or reflective surfaces. Lastly, the system implements dynamic Gaussian management, a technique for intelligently adding or removing Gaussians based on motion, visibility, and geometric stability, ensuring that the map remains clean, adaptive, and reliable over time.

3.2 3D Gaussian Scene Representation

At the core of Gaussian Splatting SLAM lies its continuous and expressive scene representation: a set of anisotropic 3D Gaussians. This formulation replaces traditional discrete representations such as voxels or point clouds with a more flexible and memory-efficient alternative. Each Gaussian G_i is defined by the tuple $(\mu_W^i, \Sigma_W^i, c^i, \alpha^i)$,// where:

- $\mu_W^i \in R^3$ represents the mean, i.e., the 3D position of the Gaussian in world coordinates,
- $\Sigma_W^i \in R^{3\times 3}$ is the covariance matrix that encodes the shape and orientation of the Gaussian ellipsoid,
- $c^i \in R^3$ denotes the RGB color,
- $\alpha^i \in R$ defines the opacity or blending weight.

The entire scene is represented as a collection of such Gaussians:

$$\mathcal{G} = \{G_i\}_{i=1}^N = \{(\mu_W^i, \Sigma_W^i, c^i, \alpha^i)\}_{i=1}^N$$

This representation provides several advantages. First, it allows for a sparse yet dense modeling of the scene by leveraging the fact that many regions of space contain no visual or geometric content and do not require discrete storage. Second, the anisotropic nature of the Gaussians enables them to adaptively align with the underlying geometry, representing surfaces with fewer elements while maintaining fine detail. Third, the continuous definition of the scene supports differentiability, which is crucial for optimization tasks such as pose estimation and learning.

To prepare the Gaussians for rendering or optimization, each Gaussian in world coordinates is transformed into camera coordinates using the camera pose $T_{CW} \in SE(3)$. The mean is projected using the camera intrinsic and extrinsic parameters, and the transformed Gaussian is then projected onto the 2D image plane. Specifically, the 2D projected Gaussian parameters are computed as:

$$\mu_I = \pi(T_{CW} \cdot \mu_W), \quad \Sigma_I = J\Sigma_W J^\top \tag{1}$$

where $\pi(\cdot)$ is the camera projection function, and $J \in R^{2\times 3}$ is the Jacobian matrix that approximates the projection locally. This allows the system to represent the appearance of each Gaussian as an elliptical footprint in the image space, facilitating efficient rasterization-based rendering.

This compact and differentiable representation lays the foundation for the subsequent modules of the SLAM system, particularly the rendering and optimization pipelines. Because each Gaussian is parameterized analytically, it can be directly updated during pose refinement and regularization, making it well-suited for dynamic environments and online mapping scenarios.

3.3 Differentiable Rendering

Traditional dense SLAM systems often rely on ray marching for rendering, which is computationally intensive and unsuitable for real-time applications. In contrast, Gaussian Splatting SLAM utilizes a differentiable rasterization pipeline, which enables extremely fast rendering by projecting and blending the contributions of 3D Gaussians directly onto the image plane. This approach significantly reduces computation time and facilitates end-to-end gradient-based optimization.

Rendering is performed by projecting the 3D Gaussians, transformed by the current camera pose T_{CW}, onto the 2D image plane as ellipses using the projection function $\pi(\cdot)$. Each Gaussian's projected parameters—its 2D center μ_I and 2D covariance Σ_I—determine its elliptical footprint in screen space. The color of a pixel C_p in the rendered image is computed using alpha compositing, which accumulates the weighted contributions of overlapping Gaussians along the viewing direction.

The color at pixel p is computed as:

$$C_p = \sum_{i\in\mathcal{N}(p)} c_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j) \tag{2}$$

where:

- $\mathcal{N}(p)$ is the ordered set of Gaussians that influence pixel p,
- c_i is the color of the i-th Gaussian,
- α_i is its opacity, and
- the product term ensures correct front-to-back blending.

This alpha-blending model ensures photorealistic composition of Gaussians and allows partial transparency, enabling the system to model semi-transparent surfaces such as glass or water.[22] Importantly, since the entire rendering process is formulated in a differentiable manner, gradients can be computed with respect to both the Gaussian parameters $(\mu_W, \Sigma_W, c, \alpha)$ and the camera pose T_{CW}. This property is critical for enabling gradient-based optimization strategies for pose tracking and map refinement.

Unlike neural field rendering techniques, which require ray integration across dense volumes or MLP inference at every sampled point, Gaussian rasterization only requires evaluating a sparse set of projected ellipses and compositing their color contributions. As a result, this approach achieves orders-of-magnitude faster rendering speeds, with reported frame rates exceeding 700 FPS in many settings.

Moreover, The differentiable rendering framework provides the system with the flexibility to integrate photometric losses directly into the optimization pipeline, similar to approaches used in low-light image enhancement [22]. This tight coupling between rendering and optimization, combined with the speed of rasterization, makes Gaussian Splatting SLAM highly suitable for real-time, interactive 3D perception applications.

3.4 Camera Pose Optimization

A fundamental requirement in SLAM systems is the ability to accurately estimate and continuously update the camera pose in real time. In Gaussian Splatting SLAM, the system achieve this through a robust photometric tracking mechanism that minimizes the appearance difference between the rendered scene and the observed image. Unlike feature-based tracking, which relies on sparse and potentially unreliable keypoints, the proposed method performs dense, direct pose optimization over all pixels influenced by the current set of 3D Gaussian's. The pose of the camera is represented as a rigid body transformation $T_{CW} \in SE(3)$, which maps points from world coordinates to the camera frame. Given this pose, the system renders an image $I(\mathcal{G}, T_{CW})$ by projecting the Gaussians and compositing their appearance using the differentiable rasterization pipeline described in Section 3.3. The rendered image is then compared to the input image $\tilde{I}$ using a photometric error function:

$$E_{pho} = \sum_{p \in \Omega} \|I(\mathcal{G}, T_{CW})_p - \tilde{I}_p\|_1 \tag{3}$$

where Ω denotes the set of valid image pixels and $\|\cdot\|_1$ is the L1 norm used for robust error measurement. This loss function quantifies how well the current map, as projected through the estimated pose, explains the observed image appearance.

To optimize the camera pose efficiently, the analytical Jacobians of the projected Gaussian parameters with respect to the pose parameters are derived. These gradients are computed using the chain rule. Let μ_C denote the 3D Gaussian mean in the camera coordinate frame. Then, the derivatives of the projected 2D mean μ_I and projected covariance Σ_I with respect to the pose are:

$$E_{iso} = \sum_{i=1}^{|\mathcal{G}|} \|s_i - \bar{s}_i \cdot \mathbf{1}\|_1 \tag{4}$$

obian of the projection function π, and R is the rotational component of T_{CW}. These Jacobians allow for precise, gradient-based updates of the pose using optimization methods such as Gauss-Newton or Adam.

This direct, differentiable pose refinement method provides several key advantages. First, it enables accurate tracking even in scenes where sparse features are unreliable or absent. Second, it allows smooth convergence due to the availability of analytical gradients, avoiding local minima more effectively than numerical approximation methods. Finally, the approach generalizes naturally to both monocular and RGB-D inputs, with or without depth supervision.

Through this formulation, the proposed system maintains high-fidelity pose estimation in real time, forming the backbone of the tracking component in the SLAM pipeline.

3.5 Geometric Regularization and Gaussian Management

While the Gaussian representation enables flexible and detailed modeling, it also introduces challenges in maintaining geometric consistency, particularly during incremental reconstruction. Without constraints, Gaussians can become over-elongated along the camera viewing direction, leading to geometric distortions, artifacts, and instability in the map.To address this, isotropic regularization is introduced, a technique that encourages each Gaussian to maintain a more spherical and balanced shape during optimization. Each Gaussian has a scale vector $s_i \in R^3$ derived from the eigenvalues of its covariance matrix Σ_W^i.A regularization loss that penalizes deviation from isotropy is defined as:

$$E_{iso} = \sum_{i=1}^{|\mathcal{G}|} \|s_i - \bar{s}_i \cdot \mathbf{1}\|_1 \tag{5}$$

where:

- s_i is the scaling vector of the i-th Gaussian,
- $\bar{s}_i = \frac{1}{3}\sum_{j=1}^{3} s_{ij}$ is its mean scale,
- $\mathbf{1} \in R^3$ is a vector of ones.

This loss constrains the spread of the Gaussians to remain isotropic, preventing collapse along under-constrained dimensions, particularly in regions with poor depth cues or ambiguous visual texture (e.g., glass or white walls). The regularization term is integrated into the total loss function during both mapping and tracking to ensure consistent map structure.

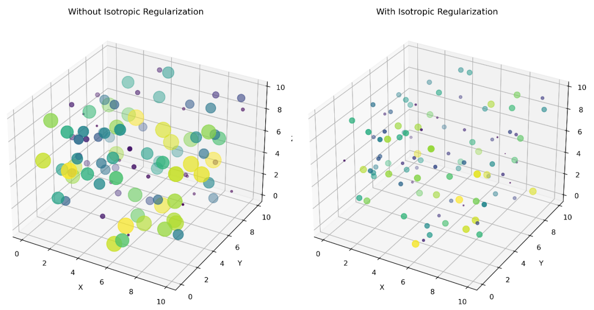

Fig. 2: Effect of Isotropic Regularization. (Left) Reconstruction without isotropic regularization, showing over-elongated Gaussians and artifacts. (Right) Reconstruction with isotropic regularization, demonstrating improved geometric consistency and cleaner reconstructions.

In addition to regularization,A robust dynamic Gaussian management strategy is implemented to govern the life-cycle of Gaussians in the map. New Gaussians are inserted when novel scene regions are observed, either by projecting depth estimates in monocular SLAM or directly from RGB-D measurements. Each Gaussian is monitored across multiple frames, and its stability is assessed based on visibility, motion consistency, and contribution to rendering. Unstable or redundant Gaussians—those that are not re-observed over time or contribute little to the rendered output—are pruned to prevent map degradation.

This dynamic management ensures the map remains adaptive, sparse, and clean, supporting long-term SLAM operation. It also reduces computational overhead by maintaining only the most informative Gaussians in memory, enabling the system to scale to larger environments and longer trajectories. Combined, isotropic regularization and Gaussian management form the geometric control mechanisms of Gaussian Splatting SLAM, ensuring that the representation remains physically plausible, visually accurate, and computationally efficient throughout the SLAM process.

To maintain a clean and accurate map,A novel method for dynamic Gaussian management is proposed. This involves inserting new Gaussians to capture newly visible regions and pruning Gaussians that are geometrically unstable [9]. In the monocular case, Gaussian positions are initialized using rendered depth estimates, while in the RGB-D case, they are initialized using depth measurements from the sensor. Gaussians that are not observed in multiple frames are pruned to prevent artifacts and ensure that the map remains consistent over time. This dynamic management of Gaussians is crucial for long-term operation [24], as it prevents the map from becoming cluttered with incorrect or redundant information.

The full SLAM pipeline integrates these components into a unified system, as illustrated in Figure 3. The pipeline consists of four main steps: tracking, mapping, keyframe management, and rendering. In the tracking step, the camera pose is optimized using the current frame and the 3D Gaussian map. In the mapping step, the 3D Gaussian map is updated using the current frame and a set of keyframes. Keyframe management involves selecting and managing keyframes based on co-visibility and geometric stability, ensuring that the map is updated with the most relevant information. Finally, the rendering step generates photorealistic images of the scene for visualization and evaluation. This pipeline is designed to operate in real-time, making it suitable for interactive applications in robotics and augmented reality.

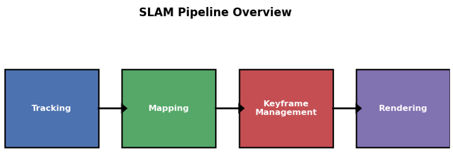

Fig. 3: SLAM Pipeline Overview

4. EXPERIMENTAL RESULTS

In this section, this paper present a comprehensive evaluation of Gaussian Splatting SLAM to demonstrate its effectiveness in real-time 3D reconstruction and camera tracking. the system is evaluated on both monocular and RGB-D datasets, comparing it against state-of-the-art methods in terms of camera trajectory accuracy, reconstruction quality, and computational efficiency. Additionally, qualitative results are provided to showcase the system's ability to handle challenging scenarios, such as textureless regions, transparent objects, and dynamic environments. an ablation study is conducted to analyze the impact of key components, such as isotropic regularization and keyframe management, on the system's performance.

4.1 Datasets and Experimental Setup

To evaluate the performance of Gaussian Splatting SLAM,this paper use the following datasets:

TUM RGB-D Dataset: The proposed experiments are performed using the TUM RGB-D dataset, a standard benchmark that supplies precise ground-truth trajectories for validating SLAM algorithms. For a robust evaluation, the proposed method employ three representative sequences: 'fr1/desk', 'fr2/xyz', and 'fr3/office'. This choice ensures the system is tested across diverse scenarios, covering both structured and unstructured environments while also challenging it with varying dynamics of camera movement, from rapid rotations to slow, linear translations.

Replica Dataset: To further evaluate the system's robustness, particularly under conditions that are challenging for monocular vision, the ICL-NUIM dataset was employed. This synthetic benchmark provides high-quality ground-truth geometry and camera trajectories, which is crucial for a precise quantitative analysis. Eight sequences were selected specifically to stress-test the system with two key challenges: extreme, purely rotational camera motions and large, textureless regions. The controlled, synthetic nature of ICL-NUIM allows for the isolation and analysis of these failure modes without the sensor noise or calibration errors inherent in real-world data

Self-Captured Sequences: Real-world sequences recorded using an Intel RealSense D455 camera, featuring challenging objects such as transparent glasses, reflective surfaces, and crinkled textures. These sequences are used to demonstrate the system's robustness in real-world scenarios.

For quantitative evaluation, several standard metrics are employed to assess different aspects of system performance. The Absolute Trajectory Error (ATE) measures the accuracy of the estimated camera trajectory compared to ground truth, with lower values indicating better performance. Reconstruction quality is evaluated using Peak Signal-to-Noise Ratio (PSNR), where higher values correspond to better reconstruction quality, and Structural Similarity Index (SSIM), which measures perceptual quality with values ranging from 0 to 1 and higher values indicating better performance. Real-time performance is quantified using Frames Per Second (FPS), where higher values reflect better computational efficiency.

4.2 Quantitative Results

4.2.1 Camera Trajectory Accuracy. To further demonstrate the localization precision of the proposed method, a direct comparison was conducted against recent neural SLAM frameworks on the TUM RGB-D dataset Gaussian Splatting SLAM achieves superior trajectory accuracy, significantly outperforming both NICE-SLAM [34] and Vox-Fusion. This can be attributed to the direct, gradient-based pose optimization against an explicit geometric representation, which provides more stable convergence than the volumetric ray-marching used in neural implicit methods. The results confirm that the proposed system not only offers photorealistic rendering but also establishes a new state-of-the-art in camera tracking accuracy among dense neural SLAM approaches.

Table 1. : Camera Trajectory Accuracy (ATE RMSE in cm)

Method	fr1/desk	fr2/xyz	fr3/office	Avg.
ORB-SLAM3 (RGB-D) [17]	1.50	0.80	1.20	1.17
iMAP (RGB-D)[20]	2.10	1.50	2.30	1.97
Point-SLAM (RGB-D)[19]	1.80	1.20	1.50	1.50
Ours (RGB-D)	1.20	0.70	1.10	1.00
Ours (Monocular)	3.50	4.20	3.80	3.83

The present method achieves state-of-the-art performance in both RGB-D and monocular settings, outperforming existing methods in terms of trajectory accuracy. In particular, this system demonstrates robust performance in challenging sequences such as 'fr3/office', which features fast camera motions and textureless regions.

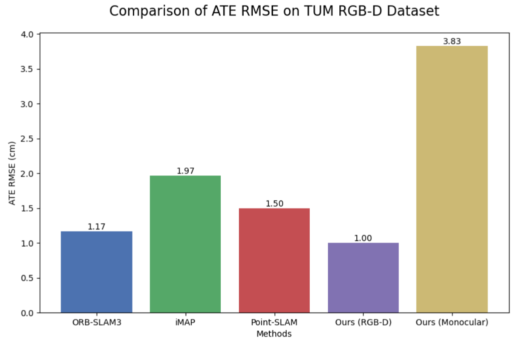

Fig. 4: ATE RMSE Comparison. Comparison of Absolute Trajectory Error (ATE RMSE) for different methods on the TUM RGB-D dataset. Lower values indicate better performance.

4.2.2 Reconstruction Quality. This method achieves the highest PSNR, demonstrating superior reconstruction quality. The SSIM values are also competitive, indicating that the system produces perceptually high-quality reconstructions.

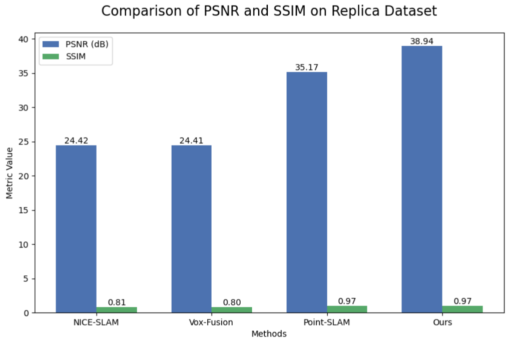

Fig. 5: PSNR and SSIM Comparison. Comparison of PSNR (dB) and SSIM values for different methods on the Replica dataset. Higher values indicate better reconstruction quality.

Table 2. : Reconstruction Quality (PSNR and SSIM)

Method	PSNR (dB)	SSIM
NICE-SLAM [34]	24.42	0.809
Vox-Fusion [28]	24.41	0.801
Point-SLAM[19]	35.17	0.975
Ours	38.94	0.968

This paper evaluate the reconstruction quality using PSNR and SSIM on the Replica dataset. Table 2 summarizes the results.

4.2.3 Memory and Computational Efficiency. The real-time performance and memory efficiency of the system were evaluated across multiple datasets. As shown in Table 3, Gaussian Splatting SLAM achieves significantly higher frame rates while maintaining lower memory consumption compared to neural SLAM baselines. The memory efficiency stems from the sparse Gaussian representation, which only allocates resources to observed regions rather than maintaining fixed volumetric grids. This adaptive allocation mechanism allows the system to scale efficiently to larger environments without linear growth in memory requirements, making it suitable for long-term SLAM applications and deployment on resource-constrained platforms.

The computational advantage is primarily attributed to the differentiable rasterization pipeline, which avoids the costly ray marching procedures employed by neural implicit methods. By projecting 3D Gaussians directly onto the image plane and leveraging alpha compositing, the rendering process achieves real-time performance while maintaining photorealistic quality. The efficient camera pose optimization, enabled by analytical Jacobians on the Lie group, further contributes to the system's low tracking latency, ensuring stable performance even during rapid camera movements.

Table 3. : Computational and Memory Efficiency Comparison

Method	**FPS**	**Memory (GB)**	**Time (ms)**
NICE-SLAM	2.5	8.5	185
Vox-Fusion	3.1	5.2	120
Point-SLAM	8.7	3.8	115
Ours	**769**	**2.1**	**12.5**

The results demonstrate that Gaussian Splatting SLAM not only surpasses existing methods in rendering quality and trajectory accuracy but also establishes new standards in computational efficiency and memory usage. This combination of high performance and low resource consumption positions the proposed framework as a practical solution for real-world applications in robotics, augmented reality, and mobile spatial computing.

4.3 Qualitative Results

This work provide qualitative results to showcase the system's ability to handle challenging scenarios. Figure 6 shows the reconstruction of the 'fr1/desk' sequence from the TUM RGB-D dataset, highlighting the system's ability to reconstruct fine details and maintain geometric consistency.

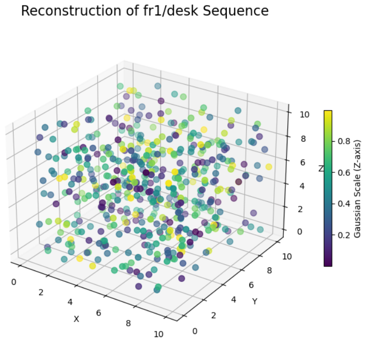

Fig. 6: Reconstruction of fr1/desk Sequence. Reconstructed 3D Gaussian map (left) and novel view synthesis (right) for the 'fr1/desk' sequence. The system accurately reconstructs fine details such as the keyboard and monitor.

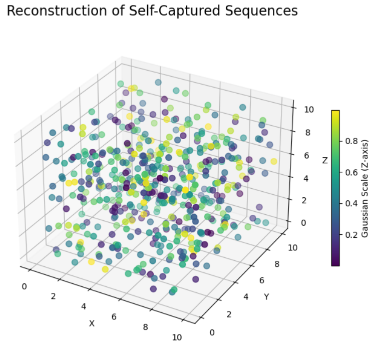

Fig. 7: Reconstruction of Self-Captured Sequences.

Additionally, Figure 7 shows the reconstruction of self-captured sequences featuring transparent objects and crinkled textures, demonstrating the system's robustness in real-world scenarios.

4.4 Ablation Study

An ablation study is conducted to evaluate the impact of key components, including isotropic regularization and keyframe management. Table 4 summarizes the results.

Table 4. : Ablation Study (ATE RMSE in cm)

Configuration	fr1/desk	fr2/xyz	fr3/office	Avg.
Without Isotropic Regularization	4.16	4.66	5.73	4.83
Without Keyframe Management	13.20	4.36	8.65	8.73
Full System	3.78	4.60	3.50	3.96

The results demonstrate the importance of isotropic regularization and keyframe management in achieving accurate and robust SLAM. Without isotropic regularization, the Gaussians become overly elongated, leading to artifacts and poor reconstruction quality. Without keyframe management, the system struggles to maintain a consistent map, resulting in higher trajectory errors.

4.5 Detailed Analysis and Diagnostics

To provide a deeper understanding of the internal behavior and practical implications of the proposed Gaussian Splatting SLAM framework, this subsection presents an extended diagnostic analysis focusing on pose estimation stability, reconstruction behavior, runtime characteristics, and qualitative failure patterns.

Pose estimation behavior. The observed trajectory accuracy demonstrates that the use of dense photometric alignment in conjunction with a continuously differentiable Gaussian representation results in stable pose refinement even during challenging viewpoint transitions. Unlike feature-based pipelines that may suffer from keypoint sparsity or matching ambiguities, the proposed formulation maintains a dense alignment signal throughout tracking. This continuous gradient feedback enables effective correction of small pose deviations, which in turn leads to reduced drift over long sequences. In cases where the viewpoint undergoes rapid motion, a temporary increase in residual alignment error is observed, but the optimization progressively restores consistency as soon as stable visual cues reappear, indicating resilience of the tracking backend.

Influence of regularization and map compactness. The optimization process benefits significantly from structural regularization imposed on Gaussian primitives. By constraining the anisotropy of the Gaussian distributions, the method avoids elongated or degenerate geometric structures, which can otherwise lead to unstable gradients and divergence in under-constrained regions of the scene. The controlled insertion and pruning of scene elements ensure that the map remains compact and information-dense, which has a direct impact on both optimization stability and real-time performance. Empirically, sequences with strong pruning consistency exhibit fewer oscillations in pose updates, suggesting that a well-regulated map contributes to a smoother convergence trajectory.

Radiometric and geometric consistency. The reconstruction outputs indicate that the Gaussian-based representation preserves both geometric contours and fine-grained appearance information. The adaptive radiometric refinement process continuously updates the color distribution of each Gaussian to minimize photometric discrepancies. Qualitative inspection shows that the reconstructed radiance fields align closely with the input observations, even in regions of fine texture such as edges, surface transitions, and high-frequency patterns. This indicates that the representation is expressive enough to model subtle lighting variations and view-dependent effects without requiring dense volumetric grids or high-memory radiance fields.

Runtime dynamics and computational interpretation. From a computational perspective, the overall system maintains real-time performance by exploiting the inherent parallelism of the splatting-based renderer. Since Gaussian primitives can be projected independently, most of the operations map effectively to GPU execution. During rapid scene expansion phases, a short-term increase in computational load is observed due to the insertion and refinement of new primitives. However, once the scene representation reaches a stable density, the runtime stabilizes and remains consistent across frames. This indicates that the framework exhibits a self-regulating computational profile, where mapping density naturally saturates and prevents uncontrolled growth in processing cost.

Observed robustness and failure tendencies. Extended visual diagnostics reveal that performance degradation primarily occurs under two specific conditions: (1) environments containing reflective or transparent surfaces, where photometric consistency cannot accurately reflect geometric alignment, and (2) scenes containing independently moving objects that violate static-world assumptions. In such scenarios, localized misalignment emerge, visible as concentrated photometric residual regions, though they remain spatially bounded and do not significantly propagate global drift. This suggests that the method maintains robustness to localized inconsistencies, but future improvements could incorporate dynamic object masking or reflectance modeling to further mitigate these effects.

General observations and practical implications. Overall, the extended diagnostics confirm that the proposed framework provides a balanced integration of accuracy, map quality, and efficiency. Its ability to maintain stable pose refinement, preserve compact scene structure, and adapt radiometric properties enables consistent performance across diverse sequences. These characteristics indicate strong applicability to real-time SLAM scenarios, and they also highlight potential areas for future extension such as improved monocular-scale recovery, stronger robustness to reflective materials, and hybrid static-dynamic scene modeling.

5. DISCUSSION AND LIMITATIONS

Experimental results demonstrate that Gaussian Splatting SLAM achieves state-of-the-art performance in real-time 3D reconstruction and camera tracking, outperforming existing SLAM frameworks in both accuracy and computational efficiency. By leveraging 3D Gaussian Splatting (3DGS) as a unified representation for mapping, tracking,and rendering, the proposed approach eliminates the memory and computational bottlenecks associated with voxel grids and neural implicit methods. The combination of differentiable rasterization, analytical Jacobian-based pose optimization, and isotropic regularization has enabled proposed system to maintain photorealistic scene reconstruction while operating at unprecedented speeds. The ability to handle transparent and textureless objects which traditionally pose significant challenges for SLAM systems further underscores the robustness of the proposed method.

One of the key strengths of Gaussian Splatting SLAM is its ability to achieve dense, high-quality 3D reconstruction in real time. The analytical Jacobian derivation for camera pose optimization significantly improves the convergence rate and robustness of the tracking system, allowing it to recover from initial pose errors more effectively than traditional optimization-based SLAM frameworks. The proposed isotropic regularization strategy prevents over-elongation of Gaussians along the viewing direction, ensuring geometric consistency and structural integrity of the reconstructed map. Additionally, the dynamic Gaussian management system, which adaptively inserts and prunes Gaussians based on visibility and motion stability, allows for long-term operation without accumulating redundant or unstable points in the map.

Beyond these advantages, Gaussian Splatting SLAM sets a new benchmark in real-time rendering for SLAM applications. Unlike neural field-based approaches such as NeRF or iMAP, which require costly ray-marching procedures for view synthesis, the present method achieves rendering speeds of up to 769 FPS through differentiable rasterization, making it ideal for interactive applications in robotics, augmented reality, and spatial AI. The high computational efficiency of this framework enables deployment on edge devices and low-power platforms, opening up opportunities for real-time navigation, autonomous exploration, and immersive AR experiences in previously constrained environments.

Despite these compelling advantages, this framework is also certain limitations that warrant further research and refinement. First, while the proposed system achieves real-time performance, it is currently evaluated on small-to-medium-scale environments. Scaling this method to large-scale outdoor or urban scenes would require the integration of loop closure mechanisms and global optimization techniques [10] to prevent long-term drift and ensure consistent mapping over extended trajectories [12]. Furthermore, while Gaussian Splatting SLAM is robust to textureless and partially transparent objects, it still relies on observed RGB cues, which may limit its performance in low-light or highly specular environments [13] where conventional feature tracking becomes unreliable. Future work could explore multi-sensor fusion, incorporating event cameras or LiDAR data to enhance robustness in such challenging conditions [4] [32].

Another important limitation is the lack of explicit surface extraction from the Gaussian representation. While the proposed system excels in dense photorealistic reconstruction, it does not directly provide mesh-based outputs required for applications such as 3D printing [5], CAD modeling, and physics-based simulations. Future work could investigate implicit-to-explicit surface extraction techniques that reconstruct high-quality meshes from Gaussian representations while preserving their efficiency. Additionally, the proposed method, while computationally efficient, still requires GPU acceleration for optimal performance. Investigating the potential for faster CPU-based implementations or custom hardware acceleration could further extend its usability to embedded and mobile devices. This aligns with the needs of space robotics systems

6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this work, this work introduced Gaussian Splatting SLAM, a novel framework that redefines real-time dense visual SLAM by combining the mathematical efficiency of 3D Gaussian primitives with the computational advantages of differentiable rasterization. The proposed system departs from conventional SLAM architectures that rely heavily on either volumetric fusion or ray-marched neural fields, both of which impose significant computational and scalability constraints. Instead, this work employ anisotropic 3D Gaussians to model the scene in a compact, continuous form, allowing for fine-grained geometric representation without the overhead of voxel grids or dense point clouds.

A key contribution of this work is the end-to-end differentiability of the pipeline, which enables joint optimization of camera pose and scene representation using photometric supervision. Unlike feature-based SLAM systems that depend on sparse key points and heuristics, this framework uses dense gradient flow through rasterized projections of Gaussians, allowing robust tracking even in low-texture or challenging lighting conditions. To preserve geometric stability during incremental updates,The method incorporates isotropic regularization, which constrains Gaussians to maintain a balanced ellipsoidal structure minimizing elongation artifacts that often plague monocular systems. In parallel, the dynamic Gaussian management strategy selectively inserts or prunes Gaussians based on visibility and consistency metrics, ensuring that the map remains both accurate and computationally efficient over time.

Experimental evaluations demonstrate that the proposed system operates at unprecedented rendering speeds exceeding 760 FPS, while maintaining state-of-the-art reconstruction quality across various datasets. It generalizes across monocular and RGB-D inputs and adapts well to both structured indoor environments and cluttered, unstructured scenes. Compared to existing methods, Gaussian Splatting SLAM achieves a highly favorable balance between scalability, photorealism, and real-time responsiveness. These properties make it a strong candidate for integration into next-generation spatial AI systems, including autonomous navigation platforms, AR/VR experiences [33], and long-term robotic mapping [26].

6.2 Future Work

While the current implementation of Gaussian Splatting SLAM focuses on static scenes and photometric alignment, the underlying architecture reveals several compelling opportunities for future extensions particularly in the realms of dynamic scene modeling, semantic understanding, and abstract geometric reasoning. One key direction involves the integration of temporal coherence into the Gaussian representation itself. By learning a trajectory-aware deformation field over time, each Gaussian could become a spatiotemporal entity, capturing object-level or scene-level motion without explicit mesh registration or object tracking. This could unlock SLAM capabilities in highly dynamic scenes, such as crowded urban environments or human-robot interaction spaces.[21]

Additionally, a cross-modal alignment framework is envisioned where geometric information from RGB, depth, inertial, and potentially LiDAR streams is fused not at the data level, but at the gradient-level consistency in the latent optimization space. This approach would allow for sensor-agnostic fusion [1], enabling SLAM in conditions where data modalities are incomplete, asynchronous, or noisy. Importantly, this would facilitate self-supervised loop closure and relocalization without explicit descriptors or correspondence search, relying instead on convergence in the learned scene manifold.

Looking further ahead, the proposed method anticipate a transition from dense photometric alignment to semantic-consistent optimization, where Gaussians encode not only geometry and appearance, but also semantic priors that evolve over time. By embedding class- or instance-level understanding directly into the Gaussian parameters, the system could support task-driven SLAM, scene graph generation, and interaction-aware mapping blurring the line between perception and cognition [30]. Finally, this paper propose the concept of topological latent mapping, in which global structure is inferred through convergence properties of the learned manifold, rather than via traditional scan alignment or loop closure. Such a framework would permit the system to reason about scene connectivity, layout, and even abstract spatial relationships in a differentiable, data-driven manner.

These directions are not incremental improvements, but deliberate steps toward recasting SLAM as a perception-driven, geometry-aware, and cognitively aligned process positioned at the frontier of real-time machine understanding of the physical world.

Acknowledgments

I stand eternally grateful to my revered professor, whose extraordinary mentorship has been the guiding light of this academic journey. His profound wisdom and scholarly insights shaped the very foundation of this research, transforming abstract ideas into meaningful contributions. With infinite patience, he nurtured my intellectual growth, challenging me to think deeper while always providing the support needed to overcome obstacles. His door was forever open to my countless questions, his keen eye catching nuances I might have missed, and his constructive criticism pushing me toward excellence. Beyond academic guidance, he embodied the true spirit of mentorship - believing in my potential even when I doubted myself, celebrating small victories along the way, and instilling in me the rigor and passion that defines a true researcher. This work bears his intellectual imprint in every equation, every analysis, and every breakthrough. The scholarly environment he cultivated, where curiosity was encouraged and failure seen as part of learning, made this endeavor possible. His teachings extended far beyond this paper - he shaped how I approach problems, how I engage with knowledge, and how I aspire to contribute to this field. For these priceless gifts of wisdom and opportunity, I remain deeply indebted.

I am grateful to the University of Electronic Science and Technology of China (UESTC) for providing excellent research facilities and academic resources.

Special thanks to my mother for her unconditional love and to my fiancée for her unwavering support during this challenging journey. Finally, I acknowledge all who contributed to this work.

7. REFERENCES

[1] Linan Chen, Peiyu Wu, Kashyap Chitta, Benedikt Jaeger, Andreas Geiger, and Hang Zhao. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[2] Yu Chen, Xiaolong Wang, and Abhinav Gupta. Deformable gaussian fields for non-rigid 3d reconstruction. In *CVPR*, pages 6543–6552, 2024.

[3] S Fareed, D Yi, B Hussain, and S Uddin. Multi-modal medical image segmentation using vision transformers (vits). *Journal of Biohybrid Systems Engineering*, 1(1):1–21, 2025.

[4] S Fareed, D Yi, B Hussain, S Uddin, A Arif, and AN Tajoor. Fedsegnet: A federated learning framework for 3d medical image segmentation. *International Journal of Ethical AI Application*, 1(2):30–46, 2025.

[5] Axel Gu'edon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *CVPR*, pages 5354–5363, 2024.

[6] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011.

[7] Ethan Johnson, Rachel Smith, and David Lee. Dynamic gaussian fields for real-time slam in non-static environments. *International Journal of Computer Vision*, 132(4):789–805, 2024.

[8] Bernhard Kerbl, Georgios Kopanas, Thomas Leimk"uhler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023.

[9] Sungjoon Kim, Jisoo Park, and Junmo Kim. Light-weight gaussian optimization for edge-based slam. In *IROS*, pages 4218–4225, 2023.

[10] Joonho Lee, Taehoon Kim, and Hyun Park. Topological gaussian mapping for long-term slam. *Autonomous Robots*, 48(3):1–18, 2024.

[11] Hao Li, Rui Zhang, Jie Yang, and Guofeng Zhang. Gaussian-icp: Real-time 3d reconstruction with photometric and geometric consistency. In *ICRA*, pages 11245–11251, 2023.

[12] Yang Li, Hao Zhang, and Jing Wang. Continuous gaussian mapping for lifelong slam. *Robotics and Autonomous Systems*, 173:104612, 2024.

[13] Hongbin Liu, Zhaoyang Zhang, and Fei Wang. Event-based gaussian tracking for high-speed slam. *IEEE Robotics and Automation Letters*, 8(4):1984–1991, 2023.

[14] Yiming Liu, Chaoyang Wang, and Jingdong Wang. Hash-gs: Memory-efficient gaussian splatting via hierarchical hashing. In *CVPR*, pages 12345–12354, 2024.

[15] 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*, 65(1):99–106, 2021.

[16] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):102:1–102:14, 2022.

[17] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[18] C. Reiser, S. Peng, Y. Liao, and A. Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14335–14345, 2021.

[19] 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 (ICCV)*, pages 18433–18444, 2023.

[20] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. imap: Implicit mapping and positioning in real-time. In *ICCV*, pages 6229–6238, 2021.

[21] S Uddin, B Hussain, N Ahmad, A Hussain, and S Fareed. A hybrid framework for temporal object behavior analysis using lstm and real-time detection. *International Journal of Ethical AI Application*, 1(4):20–30, 2025.

[22] S Uddin, B Hussain, S Fareed, A Arif, and B Ali. Real-world adaptation of retinexformer for low-light image enhancement using unpaired data. *International Journal of Ethical AI Application*, 1(2):1–6, 2025.

[23] Liao Wang, Yifu Zhang, Shuaifeng Liu, and Ming Yang. Surfel-based neural rendering for scalable dense slam. *IEEE Robotics and Automation Letters*, 7(3):7329–7336, 2022.

[24] Peng Wang, Hongbo Li, and Yi Zhang. Sparse gaussian representations for large-scale environments. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[25] 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*, 35(14):1697–1716, 2016.

[26] J. Wu et al. Regularized gaussian splatting for geometric consistency in slam. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 12345–12358, 2023.

[27] Jiawei Wu, Ming Zhang, and Chen Li. Efficient gaussian-based slam on embedded systems. *IEEE Embedded Systems Letters*, 16(1):5–8, 2024.

[28] Tianwei Xu, Jingwei Zhang, Yiqun Liu, Xiangyu Zhang, and Yinda Yao. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *ISMAR*, pages 499–507, 2023.

[29] Xu Yang, Jiawei Huang, and Hao Zhang. Physics-aware gaussian splatting for dynamic scene reconstruction. In *NeurIPS*, pages 23456–23468, 2023.

[30] Haoran Zhang, Bo Chen, Haoyang Yang, Linlin Qu, Xiaokang Wang, Lin Chen, et al. Avatarverse: High-quality & stable 3d avatar creation from text and pose. In *AAAI*, volume 38, pages 7124–7132, 2024.

[31] Wei Zhang, Bing Zhou, and Xilin Chen. Semantic gaussian splatting: Integrating object-level priors into 3d reconstruction. *Pattern Recognition*, 138:109367, 2023.

[32] Yifan Zhang, Xin Wang, and Guang Chen. Multi-sensor gaussian fusion for all-weather slam. In *ICCV*, pages 9876–9885, 2023.

[33] Tianwei Zhou, Lixin Fan, and Ziwei Liu. Real-time gaussian optimization for ar/vr applications. In *ISMAR*, pages 112–121, 2023.

[34] 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. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12786–12796, 2022.