GSplatLoc : Ultra-Precise Camera Localization via 3D Gaussian Splatting

Atticus J. Zeller
zhouge1831@gmail.com
Southeast University Chengxian College
Nanjing, China

Jane Doe*
jane.doe@another.edu
Southeast University Chengxian College
Nanjing, China

October 15, 2024

Abstract

We present **GSplatLoc**, a camera localization method that leverages the differentiable rendering capabilities of 3D Gaussian splatting for ultra-precise pose estimation. By formulating pose estimation as a gradient-based optimization problem that minimizes discrepancies between rendered depth maps from a pre-existing 3D Gaussian scene and observed depth images, GSplatLoc achieves translational errors within **0.01 cm** and near-zero rotational errors on the Replica dataset—significantly outperforming existing methods. Evaluations on the Replica and TUM RGB-D datasets demonstrate the method's robustness in challenging indoor environments with complex camera motions. GSplatLoc sets a new benchmark for localization in dense mapping, with important implications for applications requiring accurate real-time localization, such as robotics and augmented reality.

1 Introduction

Visual localization[1], specifically the task of estimating camera position and orientation (pose estimation) for a given image within a known scene, is a fundamental challenge in computer vision. Accurate pose estimation is crucial for applications like autonomous robotics (e.g., self-driving cars), as well as Augmented and Virtual Reality systems. Although Visual Simultaneous Localization and Mapping (Visual SLAM)[2, 3]combines both mapping and pose estimation, this paper focuses specifically on the localization component, which is essential for real-time tracking in dynamic environments.

Traditional SLAM systems [4] have demonstrated accurate pose estimation across diverse environments. However, their underlying 3D representations (e.g., point clouds, meshes, and surfels) exhibit limitations[5, 6] in flexibility for tasks like photorealistic scene exploration and fine-grained map updates. Recent methods utilizing Neural Radiance Fields (NeRF) [7] for surface reconstruction and view rendering have inspired novel SLAM approaches [8], which show promising[9, 10] results in tracking and scene modeling. Despite these advances[11], existing NeRF-based methods rely on computationally expensive volume rendering pipelines, limiting their ability to perform real-time **pose estimation** effectively.

The development of **3D Gaussian Splatting** [12] for efficient novel view synthesis presents a promising solution to these limitations. Its rasterization-based rendering pipeline enables faster image-level rendering, making it more suitable for real-time applications. However, integrating 3D Gaussian fields into SLAM systems still faces challenges, such as overfitting to input images due to anisotropic Gaussian fields and a lack of explicit multi-view constraints.

Current SLAM methods using 3D Gaussian Splatting, such as RTG-SLAM [13] and GS-ICP-SLAM [14], rely primarily on ICP-based techniques for pose estimation. Other approaches, like Gaussian-SLAM [15], adapt traditional RGB-D odometry methods. While these methods have shown potential, they often do not fully exploit the differentiable nature of the Gaussian Splatting representation, particularly for real-time and efficient **pose estimation**.

In this paper, we introduce **GSplatLoc**, a novel camera localization method that leverages the differentiable properties of 3D Gaussian Splatting specifically for efficient and accurate **pose estimation**. Rather than addressing the full SLAM pipeline, our approach is designed to focus solely on the localization aspect, allowing for more efficient use of the scene representation and camera pose estimation. By developing a fully differentiable pipeline, GSplatLoc can be seamlessly integrated into existing Gaussian Splatting SLAM frameworks or other deep learning tasks focused on localization.

Our main contributions are as follows:

1. We present a GPU-accelerated framework for real-time camera localization, based on a comprehensive theoretical analysis of camera pose derivatives in 3D Gaussian Splatting.
2. We propose a novel optimization approach that focuses on camera pose estimation given a 3D Gaussian scene, fully exploiting the differentiable nature of the rendering process.
3. We demonstrate the effectiveness of our method through extensive experiments, showing competitive or superior pose estimation results compared to state-of-the-art SLAM approaches utilizing advanced scene representations.

By focusing specifically on the challenges of localization in Gaussian Splatting-based scenes, GSplatLoc opens new avenues for high-precision **camera pose estimation** in complex environments. Our work contributes to the ongoing advancement of visual localization systems, pushing the boundaries of accuracy and real-time performance in 3D scene understanding and navigation.

2 Related Work

Camera localization is a fundamental problem in computer vision and robotics, crucial for applications such as autonomous navigation, augmented reality, and 3D reconstruction. Accurate and efficient pose estimation remains challenging, especially in complex and dynamic environments. In this section, we review the evolution of camera localization methodologies, focusing on classical RGB-D localization methods, NeRF-based approaches, and recent advancements in Gaussian-based techniques that utilize Iterative Closest Point (ICP) algorithms. We highlight their contributions and limitations, which motivate the development of our proposed method.

2.1 Classical RGB-D Localization

Traditional RGB-D localization methods leverage both color and depth information to estimate camera poses. These methods can be broadly categorized into feature-based, direct, and hybrid approaches.

Feature-Based Methods involve extracting and matching keypoints across frames to estimate camera motion. Notable systems such as ORB-SLAM2 [16] , ORB-SLAM3 [17] and [18] rely on sparse feature descriptors like ORB features. These systems have demonstrated robust performance in various environments, benefiting from the maturity of feature detection and matching algorithms. However, their reliance on distinct visual features makes them less effective in textureless or repetitive scenes. While they can utilize depth information for scale estimation and map refinement, they primarily depend on RGB data for pose estimation, making them susceptible to lighting changes and appearance variations.

Direct Methods[19] estimate camera motion by minimizing the photometric error between consecutive frames, utilizing all available pixel information. Methods such as Dense Visual Odometry (DVO) [4, 20] and DTAM[21] incorporate depth data to enhance pose estimation accuracy. These methods can achieve high precision in well-lit, textured environments but are sensitive to illumination changes and require good initialization to avoid local minima. The computational cost of processing all pixel data poses challenges for real-time applications. Additionally, the reliance on photometric consistency makes them vulnerable to lighting variations and dynamic scenes.

Hybrid Approaches combine the strengths of feature-based and direct methods. ElasticFusion [22] integrates surfel-based mapping with real-time camera tracking, using both photometric and geometric information. DVO-SLAM [20] combines geometric and photometric alignment for improved robustness. However, these methods often involve complex pipelines and can be computationally intensive due to dense map representations and intricate data association processes.

Despite their successes, classical methods face challenges in balancing computational efficiency with pose estimation accuracy, particularly in dynamic or low-texture environments. They may not fully exploit the potential of depth information for robust pose estimation, especially under lighting variations. Moreover, the lack of leveraging differentiable rendering techniques limits their ability to perform efficient gradient-based optimization for pose estimation.

2.2 NeRF-Based Localization

The advent of Neural Radiance Fields (NeRF) [7] has revolutionized novel view synthesis by representing scenes as continuous volumetric functions learned from images. NeRF has inspired new approaches to camera localization by leveraging its differentiable rendering capabilities.

Pose Estimation with NeRF involves inverting a pre-trained NeRF model to recover camera poses by minimizing the photometric error between rendered images and observed images. iNeRF [23] formulates pose estimation as an optimization problem, using gradient-based methods to refine camera parameters. While iNeRF achieves impressive accuracy, it suffers from high computational costs due to the per-pixel ray marching required in NeRF's volumetric rendering pipeline. This limitation hampers its applicability in real-time localization tasks.

Accelerated NeRF Variants aim to address computational inefficiency by introducing explicit data structures. Instant-NGP [24] uses hash maps to accelerate training and rendering, achieving interactive frame rates. PlenOctrees [25] and

Plenoxels [26] employ sparse voxel grids to represent the scene, significantly reducing computation time. However, even with these optimizations, rendering speeds may still not meet the demands of real-time localization in dynamic environments.

Furthermore, NeRF-based localization methods rely heavily on photometric consistency, making them sensitive to lighting variations, dynamic objects, and non-Lambertian surfaces. This reliance on RGB data can reduce robustness in real-world conditions where lighting can change dramatically. Additionally, the extensive training required on specific scenes limits their adaptability to new or changing environments.

2.3 Gaussian-Based Localization

Recent advancements in scene representation have introduced 3D Gaussian splatting as an efficient alternative to NeRF. **3D Gaussian Splatting** [12] represents scenes using a set of 3D Gaussian primitives and employs rasterization-based rendering, offering significant computational advantages over volumetric rendering.

Gaussian Splatting in Localization has been explored in methods such as SplaTAM [27], CG-SLAM [28], RTG-SLAM [13], and GS-ICP-SLAM [14]. SplaTAM introduces a SLAM system that uses gradient-based optimization to refine both the map and camera poses, utilizing RGB-D data and 3D Gaussians for dense mapping. CG-SLAM focuses on an uncertainty-aware 3D Gaussian field to improve tracking and mapping performance, incorporating depth uncertainty modeling.

Pose estimation approaches in these methods often rely on traditional point cloud registration techniques, such as Iterative Closest Point (ICP) algorithms [29]. **RTG-SLAM** employs ICP for pose estimation within a 3D Gaussian splatting framework, demonstrating real-time performance in 3D reconstruction tasks. Similarly, **GS-ICP-SLAM** utilizes Generalized ICP [30] for alignment, effectively handling the variability in point cloud density and improving robustness.

Gaussian-SLAM [15] adapts traditional RGB-D odometry methods, combining colored point cloud alignment [31] with an energy-based visual odometry approach [32]. These methods integrate ICP-based techniques within Gaussian-based representations to estimate camera poses.

While effective in certain scenarios, the reliance on ICP-based methods introduces limitations[33]. ICP algorithms require good initial alignment and can be sensitive to local minima, often necessitating careful initialization to ensure convergence. Additionally, ICP can be computationally intensive, especially with large point sets, hindering real-time performance. These methods may not fully exploit the differentiable rendering capabilities of 3D Gaussian representations for pose optimization.

Optimizing camera poses using depth information within a differentiable rendering framework offers several advantages, particularly in environments with challenging lighting or low-texture surfaces. Depth data provides direct geometric information about the scene, which is invariant to illumination changes, leading to more robust pose estimation. By focusing on depth-only optimization, methods can achieve robustness to lighting variations and improve computational efficiency by avoiding the processing of color data.

However, existing Gaussian-based localization techniques have not fully exploited depth-only optimization within a differentiable rendering framework. Challenges such as sensor noise, incomplete depth data due to occlusions, and the need for accurate initial pose estimates remain. Furthermore, many approaches tightly couple mapping and localization, introducing unnecessary computational overhead and complexity when the primary goal is pose estimation.

These limitations motivate the development of our proposed method. By leveraging the differentiable rendering capabilities of 3D Gaussian splatting specifically for depth-only pose optimization, we aim to overcome the challenges faced by existing methods. Our approach eliminates reliance on photometric data, enhancing robustness to lighting variations and reducing computational overhead. By decoupling localization from mapping, we simplify the optimization process, making it more suitable for real-time applications. Additionally, using quaternions for rotation parameterization [34] and careful initialization strategies improves the stability and convergence of the optimization, addressing challenges associated with sensor noise and incomplete data.

Our method fully exploits the differentiable rendering pipeline to perform efficient gradient-based optimization for pose estimation, setting it apart from ICP-based approaches. By focusing on depth information and leveraging the strengths of 3D Gaussian splatting, we provide a robust and computationally efficient solution for camera localization in complex environments.

3 Method

Overview. We propose **GSplatLoc**, a novel camera localization method that leverages the differentiable rendering capabilities of 3D Gaussian splatting for efficient and accurate pose estimation. By formulating pose estimation as a gradient-based optimization problem within a fully differentiable framework, GSplatLoc enables direct optimization of camera poses using depth information rendered from a pre-existing 3D Gaussian scene representation. This approach allows us to achieve high-precision localization suitable for real-time applications.

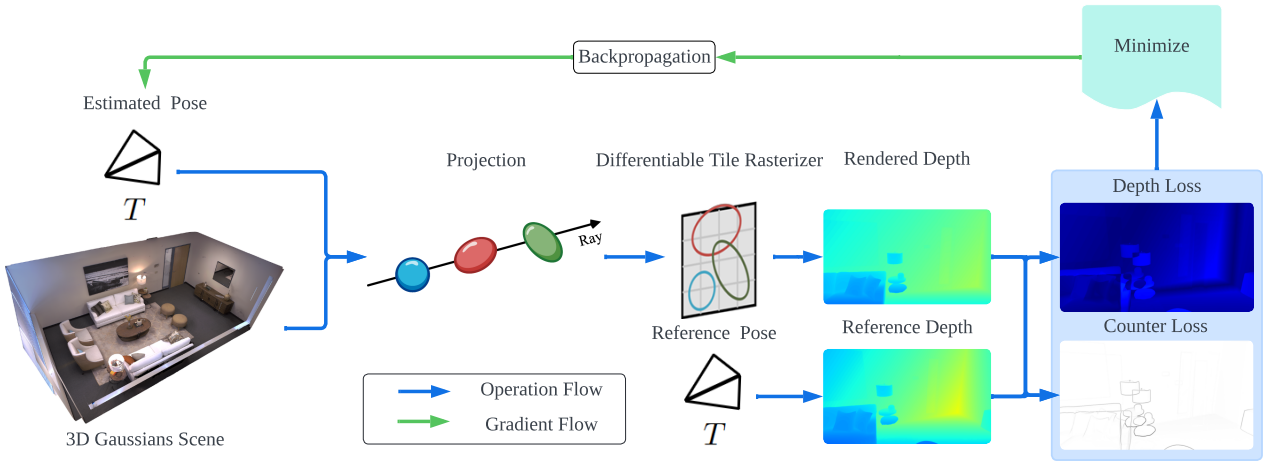

Figure 1: We propose **GSplatLoc**, a novel camera localization method that leverages the differentiable rendering capabilities of 3D Gaussian splatting for efficient and accurate pose estimation.

Motivation. Traditional SLAM systems that use point clouds, meshes, or surfels for 3D representation often face limitations in rendering quality and computational efficiency, hindering their ability to provide photorealistic scene exploration and fine-grained map updates. Neural Radiance Fields (NeRF) [7] have demonstrated exceptional rendering quality but suffer from computational inefficiencies due to per-pixel ray marching in volume rendering, making real-time applications challenging.

The recent development of **3D Gaussian Splatting** [12] offers a promising alternative by employing a rasterization-based rendering pipeline. In this method, scenes are represented using a set of 3D Gaussians, which can be efficiently projected onto the image plane and rasterized to produce high-quality renderings at interactive frame rates. The differentiable nature of this rendering process enables gradient computation with respect to both the scene parameters and the camera pose.

By leveraging these properties, we aim to develop a localization method that fully utilizes the differentiable rendering capabilities of 3D Gaussian splatting. Our approach focuses on optimizing the camera pose by minimizing the difference between the rendered depth map and the observed query depth image, thus enabling accurate and efficient pose estimation suitable for real-time SLAM systems.

Problem Formulation. Our objective is to estimate the 6-DoF pose $(\mathbf{R}, \mathbf{t}) \in SE(3)$ of a query depth image D_q, where $\mathbf{R}$ is the rotation matrix and $\mathbf{t}$ is the translation vector in the camera coordinate system. Given a 3D representation of the environment in the form of 3D Gaussians, let $\mathcal{G} = \{G_i\}_{i=1}^{N}$ denote a set of N Gaussians, and posed reference depth images $\{D_k\}$, which together constitute the reference data.

In the following sections, we detail the components of our method, including the scene representation, the differentiable depth rendering process, the formulation of the optimization problem, and the overall pipeline for camera localization.

3.1 Scene Representation

Building upon the Gaussian splatting method [12], we adapt the scene representation to focus on the differentiable depth rendering process, which is crucial for our localization task. Our approach utilizes the efficiency and quality of Gaussian splatting while tailoring it specifically for depth-based localization.

3D Gaussians. Each Gaussian G_i is characterized by its 3D mean $\boldsymbol{\mu}_i \in \mathbb{R}^3$, 3D covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$, opacity $o_i \in \mathbb{R}$, and scale $\mathbf{s}_i \in \mathbb{R}^3$. To represent the orientation of each Gaussian, we use a rotation quaternion $\mathbf{q}_i \in \mathbb{R}^4$.

The 3D covariance matrix $\boldsymbol{\Sigma}_i$ is parameterized using $\mathbf{s}_i$ and $\mathbf{q}_i$:

$$\boldsymbol{\Sigma}_i = \mathbf{R}(\mathbf{q}_i)\mathbf{S}(\mathbf{s}_i)\mathbf{S}(\mathbf{s}_i)^\top\mathbf{R}(\mathbf{q}_i)^\top$$

where $\mathbf{R}(\mathbf{q}_i)$ is the rotation matrix derived from $\mathbf{q}_i$, and $\mathbf{S}(\mathbf{s}_i) = \mathrm{diag}(\mathbf{s}_i)$ is a diagonal matrix of scales.

Projecting 3D to 2D. For the projection of 3D Gaussians onto the 2D image plane, we follow the approach described by [12]. The 3D mean $\boldsymbol{\mu}_i$ is first transformed into the camera coordinate frame using the world-to-camera transformation $\mathbf{T}_{wc} \in SE(3)$. Then, it is projected using the projection matrix $\mathbf{P} \in \mathbb{R}^{4\times4}$ and mapped to pixel coordinates via the function $\pi : \mathbb{R}^4 \to \mathbb{R}^2$:

$$\boldsymbol{\mu}_{I,i} = \pi\left(\mathbf{P}\mathbf{T}_{wc}\boldsymbol{\mu}_{i,\text{homogeneous}}\right)$$

Similarly, the 2D covariance $\boldsymbol{\Sigma}_{I,i} \in \mathbb{R}^{2\times2}$ of the projected Gaussian is obtained by transforming the 3D covariance $\boldsymbol{\Sigma}_i$ into the image plane:

$$\boldsymbol{\Sigma}_{I,i} = \mathbf{J}\mathbf{R}_{wc}\boldsymbol{\Sigma}_i\mathbf{R}_{wc}^\top\mathbf{J}^\top$$

where $\mathbf{R}_{wc}$ represents the rotation component of $\mathbf{T}_{wc}$, and $\mathbf{J}$ is the Jacobian of the projection function, accounting for the affine transformation from 3D to 2D as described by [35].

3.2 Depth Rendering

We implement a differentiable depth rendering process, which is crucial for our localization method as it allows for gradient computation throughout the rendering pipeline. This differentiability enables us to optimize camera poses directly based on rendered depth maps.

Compositing Depth. For depth map generation, we employ a front-to-back compositing scheme, which allows for accurate depth estimation and proper handling of occlusions. Let d_n denote the depth value of the n-th Gaussian, corresponding to the z-coordinate of its mean in the camera coordinate system. The depth at pixel $\mathbf{p}$, denoted $D(\mathbf{p})$, is computed as [12]:

$$D(\mathbf{p}) = \sum_{n \leq N} d_n \cdot \alpha_n \cdot T_n,$$

where T_n is the cumulative transparency up to the $(n-1)$-th Gaussian:

$$T_n = \prod_{m<n} (1 - \alpha_m).$$

In this formulation, α_n represents the opacity contribution of the n-th Gaussian at pixel $\mathbf{p}$, calculated as:

$$\alpha_n = o_n \cdot \exp(-\sigma_n),$$

with

$$\sigma_n = \frac{1}{2} \mathbf{\Delta}_n^\top \mathbf{\Sigma}_I^{-1} \mathbf{\Delta}_n.$$

Here, $\mathbf{\Delta}_n$ is the offset between the pixel center and the center of the 2D Gaussian $\boldsymbol{\mu}_{I,i}$, and $\mathbf{\Sigma}_I$ is the 2D covariance of the projected Gaussian. The opacity parameter o_n controls the overall opacity of the Gaussian.

Normalization of Depth. To ensure consistent depth representation across the image, we normalize the accumulated depth values. We first compute the total accumulated opacity at each pixel $\mathbf{p}$:

$$\alpha(\mathbf{p}) = \sum_{n \leq N} \alpha_n \cdot T_n$$

The normalized depth at pixel $\mathbf{p}$ is then defined as:

$$\text{Norm}_D(\mathbf{p}) = \frac{D(\mathbf{p})}{\alpha(\mathbf{p})}$$

This normalization ensures that the depth values are properly scaled, making them comparable across different regions of the image, even when the density of Gaussians varies.

The differentiable nature of this depth rendering process is key to our localization method. It allows us to compute gradients with respect to the Gaussian parameters and camera pose, enabling direct optimization based on the rendered depth maps. This differentiability facilitates efficient gradient-based optimization, forming the foundation for our subsequent localization algorithm.

3.3 Localization as Image Alignment

Assuming we have an existing map represented by a set of 3D Gaussians, our localization task focuses on estimating the 6-DoF pose of a query depth image D_q within this map. This process essentially becomes an image alignment problem between the rendered depth map from our Gaussian representation and the query depth image.

Rotating with Quaternions.[34] We parameterize the camera pose using a quaternion $\mathbf{q}_{cw}$ for rotation and a vector $\mathbf{t}_{cw}$ for translation. This choice of parameterization is particularly advantageous in our differential rendering context. Quaternions provide a continuous and singularity-free representation of rotation, which is crucial for gradient-based optimization. Moreover, their compact four-parameter form aligns well with our differentiable rendering pipeline, allowing for efficient computation of gradients with respect to rotation parameters.

Loss Function. Our optimization strategy leverages the differentiable nature of our depth rendering process. We define a loss function that incorporates both depth accuracy and edge alignment:

$$\mathscr{L} = \lambda_1 \mathscr{L}_d + \lambda_2 \mathscr{L}_c$$

where λ_1 and λ_2 are weighting factors (typically set to 0.8 and 0.2, respectively) that balance the contributions of the depth and contour losses. The depth loss $\mathscr{L}_d$ measures the L1 difference between the rendered depth map and the observed depth image:

$$\mathscr{L}_d = \sum_{i \in \mathscr{M}} \left| D_i^{\text{rendered}} - D_i^{\text{observed}} \right|$$

The contour loss $\mathscr{L}_c$ focuses on aligning the depth gradients (edges) between the rendered and observed depth images:

$$\mathscr{L}_c = \sum_{j \in \mathscr{M}} \left| \nabla D_j^{\text{rendered}} - \nabla D_j^{\text{observed}} \right|$$

Here, ∇D represents the gradient of the depth image, computed using the Sobel operator [36], and $\mathscr{M}$ is the mask of valid pixels determined by the rendered alpha mask.

The contour loss $\mathscr{L}_c$ serves several crucial purposes. It ensures that depth discontinuities in the rendered image align well with those in the observed depth image, thereby improving the overall accuracy of the pose estimation. By explicitly considering edge information, we preserve important structural features of the scene during optimization. Furthermore, the contour loss is less sensitive to absolute depth values and more focused on relative depth changes, making it robust to global depth scale differences.

The optimization objective can be formulated as:

$$\min_{\mathbf{q}_{cw},\mathbf{t}_{cw}} \mathscr{L} + \lambda_q \|\mathbf{q}_{cw}\|_2^2 + \lambda_t \|\mathbf{t}_{cw}\|_2^2$$

where λ_q and λ_t are regularization coefficients for the quaternion and translation parameters, respectively.

Masking Uncertainty. The rendered alpha mask plays a crucial role in our optimization process. It effectively captures the epistemic uncertainty of our map, allowing us to focus the optimization on well-represented parts of the scene. By utilizing this mask, we avoid optimizing based on unreliable or non-existent data, which could otherwise lead to erroneous pose estimates.

Optimization Parameters. We perform the optimization using the Adam optimizer, with distinct learning rates and weight decay values for the rotation and translation parameters. Specifically, we set the learning rate for quaternion optimization to 5×10^{-4} and for translation optimization to 10^{-3}, based on empirical tuning. Weight decay, set to 10^{-3} for both parameters, acts as a regularization term to prevent overfitting. These settings balance the trade-off between convergence speed and optimization stability.

3.4 Pipeline

Our GSplatLoc method streamlines the localization process by utilizing only the posed reference depth images $\{D_k\}$ and the query depth image D_q. The differentiable rendering of 3D Gaussians enables efficient and smooth convergence during optimization.

Evaluation Scene. For consistent evaluation, we initialize the 3D Gaussians from point clouds derived from the posed reference depth images $\{D_k\}$. Each point in the point cloud corresponds to the mean $\boldsymbol{\mu}_i$ of a Gaussian G_i. After filtering out outliers, we set the opacity $o_i = 1$ for all Gaussians to ensure full contribution in rendering. The scale $\mathbf{s}_i$ is initialized isotropically based on the local point density:

$$\mathbf{s}_i = (\sigma_i, \sigma_i, \sigma_i), \quad \text{with } \sigma_i = \sqrt{\frac{1}{3}\sum_{j=1}^{3} d_{ij}^2}$$

where d_{ij} is the distance to the j-th nearest neighbor of point i, computed using k-nearest neighbors (with $k = 4$). This initialization balances the representation of local geometry. The rotation quaternion $\mathbf{q}_i$ is initially set to $(1, 0, 0, 0)$ for all Gaussians, corresponding to no rotation.

To further enhance optimization stability, we apply standard Principal Component Analysis (PCA) to align the principal axes of the point cloud with the coordinate axes. By centering the point cloud at its mean and aligning its principal axes, we normalize the overall scene orientation. This provides a more uniform starting point for optimization across diverse datasets, significantly improving the stability of the loss reduction during optimization and facilitating the attainment of lower final loss values, especially in the depth loss component of our objective function.

Optimization. We employ the Adam[37] optimizer for optimizing both the quaternion and translation parameters, using the distinct learning rates and weight decay values as previously described. The optimization process greatly benefits from the real-time rendering capabilities of 3D Gaussian splatting. Since rendering is extremely fast, each iteration of the optimizer is limited mainly by the rendering speed, allowing for rapid convergence of our pose estimation algorithm and making it suitable for real-time applications.

Convergence. To determine convergence, we implement an early stopping mechanism based on the stabilization of the total loss. Our experiments show that the total loss usually stabilizes after approximately 100 iterations. We employ a patience parameter: after 100 iterations, if the total loss does not decrease for a predefined number of consecutive iterations, the optimization loop is terminated. We then select the pose estimate corresponding to the minimum total loss as the optimal pose.

In summary, our pipeline effectively combines the efficiency of Gaussian splatting with a robust optimization strategy, resulting in a fast and accurate camera localization method suitable for real-time applications.

4 Evaluation

We conducted extensive experiments to evaluate the performance of our proposed method, **GSplatLoc**, in comparison with state-of-the-art SLAM systems that utilize advanced scene representations. The evaluation focuses on assessing the accuracy of camera pose estimation in challenging indoor environments, emphasizing both the translational and rotational components of the estimated poses.

4.1 Experimental Setup

Implementation Details. Our localization pipeline was implemented on a system equipped with an Intel Core i7-13620H CPU, 16 GB of RAM, and an NVIDIA RTX 4060 GPU with 8 GB of memory. The algorithm was developed using Python and PyTorch, utilizing custom CUDA kernels to accelerate the rasterization and backpropagation processes inherent in our differentiable rendering approach. This setup ensures that our method achieves real-time performance, which is crucial for practical applications in SLAM systems.

Datasets. We evaluated our method on two widely recognized datasets for SLAM benchmarking: the **Replica** dataset [38] and the **TUM RGB-D** dataset [39]. The Replica dataset provides high-fidelity synthetic indoor environments, ideal for controlled evaluations of localization algorithms. We utilized data collected by Sucar et al. [9], which includes trajectories from an RGB-D sensor with ground-truth poses. The TUM RGB-D dataset offers real-world sequences captured in various indoor settings, providing a diverse range of scenarios to test the robustness of our method.

Metrics. Localization accuracy was assessed using two standard metrics: the **Absolute Trajectory Error (ATE RMSE)**, measured in centimeters, and the **Absolute Angular Error (AAE RMSE)**, measured in degrees. The ATE RMSE quantifies the root mean square error between the estimated and ground-truth camera positions, while the AAE RMSE measures the accuracy of the estimated camera orientations.

Baselines. To provide a comprehensive comparison, we evaluated our method against several state-of-the-art SLAM systems that leverage advanced scene representations. Specifically, we compared against RTG-SLAM (ICP) [13], which utilizes Iterative Closest Point (ICP) for pose estimation within a 3D Gaussian splatting framework. We also included GS-ICP-SLAM (GICP) [14], which employs Generalized ICP for alignment in a Gaussian-based representation. Additionally, we considered Gaussian-SLAM [15], evaluating both its PLANE ICP and HYBRID variants, which adapt traditional RGB-D odometry methods by incorporating plane-based ICP and a hybrid approach combining photometric and geometric information. These baselines were selected because they represent the current state of the art in SLAM systems utilizing advanced scene representations and focus on the localization component, which aligns with the scope of our work.

4.2 Localization Evaluation

We first evaluated our method on the Replica dataset, which provides a controlled environment to assess the accuracy of pose estimation algorithms.

Table 1. presents the ATE RMSE results in centimeters for various methods across different sequences in the Replica dataset. Our method significantly outperforms the baselines, achieving an average ATE RMSE of **0.00925 cm**, which is an order of magnitude better than the closest competitor. This substantial improvement demonstrates the effectiveness of our approach in accurately estimating the camera's position. The low translational errors indicate that our method can precisely align the observed depth images with the rendered depth from the 3D Gaussian scene.

Table 1: Replica[38] (ATE RMSE ↓[cm]).

Methods	Avg.	R0	R1	R2	Of0	Of1	Of2	Of3	Of4
RTG-SLAM(ICP)[13]	0.471	0.429	0.690	0.544	0.640	0.336	0.434	0.281	0.419
GS-ICP-SLAM(GICP)[14]	0.593	0.465	0.772	0.723	0.681	0.522	0.582	0.438	0.558
Gaussian-SLAM(PLANE ICP)[15]	0.633	0.476	0.812	0.781	0.709	0.541	0.667	0.449	0.625
Gaussian-SLAM(HYBRID)[15]	0.631	0.476	0.812	0.781	0.709	0.537	0.662	0.446	0.624
Ours	**0.009**	**0.007**	**0.008**	**0.010**	**0.009**	**0.009**	**0.011**	**0.009**	**0.011**

Table 2: Replica[38] (AAE RMSE ↓[°]).

Methods	Avg.	R0	R1	R2	Of0	Of1	Of2	Of3	Of4
RTG-SLAM(ICP)[13]	**0.576**	**0.720**	**0.826**	0.744	**0.054**	**0.537**	0.360	0.330	0.430
GS-ICP-SLAM(GICP)[14]	1.279	1.659	1.951	1.607	0.281	0.895	2.580	1.110	2.940
Gaussian-SLAM(PLANE ICP)[15]	1.287	1.834	1.880	1.398	0.305	1.019	1.060	1.100	1.130
Gaussian-SLAM(HYBRID)[15]	1.955	2.265	3.493	2.783	0.287	0.945	0.580	0.720	0.630
Ours	0.810	0.931	1.006	**0.666**	0.248	1.197	**0.011**	**0.009**	**0.011**

Table 2. presents the Absolute Angular Error (AAE) RMSE in degrees for various methods on the Replica dataset. Our method achieves a competitive average AAE RMSE of **0.80982**°, indicating superior rotational accuracy in most sequences. In sequences with significant rotational movements, such as Of2, Of3, and Of4, our approach consistently outperforms the baselines. For instance, in sequence Of3, our method achieves an AAE RMSE of **0.00930**°, compared to **0.33000**° by RTG-SLAM and higher errors by other methods. This exceptional performance can be attributed to the effective utilization of the differentiable rendering pipeline and the optimization strategy that precisely aligns the depth gradients between the rendered and observed images.

To evaluate the robustness of our method in real-world scenarios, we conducted experiments on the TUM RGB-D dataset, which presents challenges such as sensor noise and dynamic environments.

Table 3: TUM[39] (ATE RMSE ↓[cm]).

Methods	Avg.	fr1/desk	fr1/desk2	fr1/room	fr2/xyz	fr3/off.
RTG-SLAM(ICP)[13]	**0.576**	**0.720**	**0.826**	0.744	**0.054**	**0.537**
GS-ICP-SLAM(GICP)[14]	1.279	1.659	1.951	1.607	0.281	0.895
Gaussian-SLAM(PLANE ICP)[15]	1.287	1.834	1.880	1.398	0.305	1.019
Gaussian-SLAM(HYBRID)[15]	1.955	2.265	3.493	2.783	0.287	0.945
Ours	0.810	0.931	1.006	**0.666**	0.248	1.197

Table 3. presents the ATE RMSE in centimeters for various methods on the TUM-RGBD dataset [39]. Our method achieves competitive results with an average ATE RMSE of **8.0982 cm**, outperforming GS-ICP-SLAM[14] and Gaussian-SLAM[15] in most sequences. While RTG-SLAM[13] shows lower errors in some sequences, our method consistently provides accurate pose estimates across different environments.

The increased error compared to the Replica dataset is expected due to the real-world challenges present in the TUM RGB-D dataset, such as sensor noise and environmental variability. Despite these challenges, our method demonstrates robustness and maintains reasonable localization accuracy.

Table 4: TUM[39] (AAE RMSE ↓[°]).

Methods	Avg.	fr1/desk	fr1/desk2	fr1/room	fr2/xyz	fr3/off.
RTG-SLAM(ICP)[13]	**0.916**	1.181	1.557	1.355	0.138	**0.347**
GS-ICP-SLAM(GICP)[14]	0.959	1.288	1.618	1.363	0.147	0.381
Gaussian-SLAM(PLANE ICP)[15]	1.090	1.388	1.791	1.564	0.182	0.525
Gaussian-SLAM(HYBRID)[15]	1.117	1.426	2.098	1.594	**0.114**	0.355
Ours	0.979	**1.126**	**1.265**	**0.907**	0.789	0.808

Table 4. presents the AAE RMSE results in degrees for the TUM RGB-D dataset. Our method achieves an average AAE RMSE of **0.97928**°, which is competitive with the other methods. In sequences such as fr1/room, our method demonstrates superior rotational accuracy with an AAE RMSE of **0.90722**°, compared to higher errors by the baselines. The slightly higher rotational errors in the TUM RGB-D dataset, compared to the Replica dataset, can be attributed to the complexities of real-world data, including sensor inaccuracies and dynamic elements in the environment. Nonetheless, our method maintains reliable performance across various sequences.

4.3 Discussion

The experimental results indicate that our method consistently achieves high localization accuracy, particularly in terms of translational error, where we significantly outperform existing approaches on the Replica dataset. The rotational accuracy of our method is also competitive, often surpassing other methods in challenging sequences. These outcomes demonstrate the effectiveness of our approach in leveraging the differentiable rendering capabilities of 3D Gaussian splatting for pose estimation.

Several factors contribute to the superior performance of our method. By utilizing a fully differentiable depth rendering process, our method allows for efficient gradient-based optimization of camera poses, leading to precise alignment between the rendered and observed depth images. The combination of depth loss and contour loss in our optimization objective enables the method to capture both absolute depth differences and structural features, enhancing the robustness and accuracy of pose estimation. Additionally, employing quaternions for rotation representation provides a continuous and singularity-free parameter space, improving the stability and convergence of the optimization process.

While our method shows excellent performance on the Replica dataset, the increased errors on the TUM RGB-D dataset highlight areas for potential improvement. Real-world datasets introduce challenges such as sensor noise, dynamic objects, and incomplete depth data due to occlusions. Addressing these challenges in future work could further enhance the robustness of our method.

4.4 Limitations

Despite the promising results, our method has certain limitations. The reliance on accurate depth data means that performance may degrade in environments where the depth sensor data is noisy or incomplete. Additionally, our current implementation focuses on frame-to-frame pose estimation with initialization from the previous frame's ground-truth pose. In practical applications, this assumption may not hold, and integrating our method into a full SLAM system with robust initialization and loop closure capabilities would be necessary. Furthermore, handling dynamic scenes and improving computational efficiency for large-scale environments remain areas for future exploration.

5 Conclusion

In this paper, we introduced **GSplatLoc**, a novel method for ultra-precise camera localization that leverages the differentiable rendering capabilities of 3D Gaussian splatting. By formulating pose estimation as a gradient-based optimization problem within a fully differentiable framework, our approach enables efficient and accurate alignment between rendered depth maps from a pre-existing 3D Gaussian scene and observed depth images.

Extensive experiments on the Replica and TUM RGB-D datasets demonstrate that GSplatLoc significantly outperforms state-of-the-art SLAM systems in terms of both translational and rotational accuracy. On the Replica dataset, our method achieves an average Absolute Trajectory Error (ATE RMSE) of 0.00925 cm, surpassing existing approaches by an order of magnitude. The method also maintains competitive performance on the TUM RGB-D dataset, exhibiting robustness in real-world scenarios despite challenges such as sensor noise and dynamic elements.

The superior performance of GSplatLoc can be attributed to several key factors. The utilization of a fully differentiable depth rendering process allows for efficient gradient-based optimization of camera poses. The combination of depth and contour losses in our optimization objective captures both absolute depth differences and structural features, enhancing the accuracy of pose estimation. Moreover, employing quaternions for rotation representation provides a continuous and singularity-free parameter space, improving the stability and convergence of the optimization process.

While the results are promising, there are limitations to address in future work. The reliance on accurate depth data implies that performance may degrade with noisy or incomplete sensor information. Integrating GSplatLoc into a full

SLAM system with robust initialization, loop closure, and the capability to handle dynamic scenes would enhance its applicability. Additionally, exploring methods to improve computational efficiency for large-scale environments remains an important direction for future research.

In conclusion, GSplatLoc represents a significant advancement in camera localization accuracy for SLAM systems, setting a new standard for localization techniques in dense mapping. The method's ability to achieve ultra-precise pose estimation has substantial implications for applications in robotics and augmented reality, where accurate and efficient localization is critical.

1 Scaramuzza, D., Fraundorfer, F.: 'Visual odometry [tutorial]'*IEEE robotics & automation magazine*, 2011, **18**, (4), pp. 80–92.

2 Durrant-Whyte, H., Bailey, T.: 'Simultaneous localization and mapping: Part I'*IEEE robotics & automation magazine*, 2006, **13**, (2), pp. 99–110.

3 Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: 'MonoSLAM: Real-time single camera SLAM'*IEEE transactions on pattern analysis and machine intelligence*, 2007, **29**, (6), pp. 1052–1067.

4 Kerl, C., Sturm, J., Cremers, D.: 'Dense visual SLAM for RGB-D cameras', in '2013 IEEE/RSJ International Conference on Intelligent Robots and Systems' (IEEE, 2013), pp. 2100–2106

5 Newcombe, R.A., Izadi, S., Hilliges, O., *et al.*: 'Kinectfusion: Real-time dense surface mapping and tracking', in '2011 10th IEEE international symposium on mixed and augmented reality' (Ieee, 2011), pp. 127–136

6 Rusinkiewicz, S., Levoy, M.: 'Efficient variants of the ICP algorithm', in 'Proceedings third international conference on 3-D digital imaging and modeling' (IEEE, 2001), pp. 145–152

7 Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: 'NeRF: Representing scenes as neural radiance fields for view synthesis'*Commun. ACM*, 2022, **65**, (1), pp. 99–106.

8 Sandström, E., Li, Y., Van Gool, L., Oswald, M.R.: 'Point-slam: Dense neural point cloud-based slam', in 'Proceedings of the IEEE/CVF International Conference on Computer Vision' (2023), pp. 18433–18444

9 Sucar, E., Liu, S., Ortiz, J., Davison, A.J.: 'Imap: Implicit mapping and positioning in real-time', in 'Proceedings of the IEEE/CVF international conference on computer vision' (2021), pp. 6229–6238

10 Zhu, Z., Peng, S., Larsson, V., *et al.*: 'Nice-slam: Neural implicit scalable encoding for slam', in 'Proceedings of the IEEE/CVF conference on computer vision and pattern recognition' (2022), pp. 12786–12796

11 Garbin, S.J., Kowalski, M., Johnson, M., Shotton, J., Valentin, J.: 'Fastnerf: High-fidelity neural rendering at 200fps', in 'Proceedings of the IEEE/CVF international conference on computer vision' (2021), pp. 14346–14355

12 Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: '3D Gaussian Splatting for Real-Time Radiance Field Rendering'*ACM Transactions on Graphics*, 2023, **42**, (4), pp. 1–14.

13 Peng, Z., Shao, T., Liu, Y., *et al.*: 'RTG-SLAM: Real-time 3D Reconstruction at Scale using Gaussian Splatting', `http://arxiv.org/abs/2404.19706`, accessed July 2024

14 Ha, S., Yeon, J., Yu, H.: 'RGBD GS-ICP SLAM', `http://arxiv.org/abs/2403.12550`, accessed May 2024

15 Yugay, V., Li, Y., Gevers, T., Oswald, M.R.: 'Gaussian-SLAM: Photo-realistic Dense SLAM with Gaussian Splatting', `http://arxiv.org/abs/2312.10070`, accessed June 2024

16 Mur-Artal, R., Tardós, J.D.: 'Orb-Slam2: An open-source slam system for monocular, stereo, and rgb-d cameras'*IEEE transactions on robotics*, 2017, **33**, (5), pp. 1255–1262.

17 Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: 'Orb-Slam3: An accurate open-source library for visual, visual–inertial, and multimap slam'*IEEE Transactions on Robotics*, 2021, **37**, (6), pp. 1874–1890.

18 Gauglitz, S., Höllerer, T., Turk, M.: 'Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking'*Int J Comput Vis*, 2011, **94**, (3), pp. 335–360.

19 Engel, J., Koltun, V., Cremers, D.: 'Direct sparse odometry'*IEEE transactions on pattern analysis and machine intelligence*, 2017, **40**, (3), pp. 611–625.

20 Kerl, C., Sturm, J., Cremers, D.: 'Robust odometry estimation for RGB-D cameras', in '2013 IEEE international conference on robotics and automation' (IEEE, 2013), pp. 3748–3754

21 Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: 'DTAM: Dense tracking and mapping in real-time', in '2011 international conference on computer vision' (IEEE, 2011), pp. 2320–2327

22 Whelan, T., Salas-Moreno, R.F., Glocker, B., Davison, A.J., Leutenegger, S.: 'ElasticFusion: Real-time dense SLAM and light source estimation'*The International Journal of Robotics Research*, 2016, **35**, (14), pp. 1697–1716.

23 Yen-Chen, L., Florence, P., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.-Y.: 'Inerf: Inverting neural radiance fields for pose estimation', in '2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)' (IEEE, 2021), pp. 1323–1330

24 Müller, T., Evans, A., Schied, C., Keller, A.: 'Instant neural graphics primitives with a multiresolution hash encoding'*ACM Trans. Graph.*, 2022, **41**, (4), pp. 1–15.

25 Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: 'Plenoctrees for real-time rendering of neural radiance fields', in 'Proceedings of the IEEE/CVF International Conference on Computer Vision' (2021), pp. 5752–5761

26 Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: 'Plenoxels: Radiance fields without neural networks', in 'Proceedings of the IEEE/CVF conference on computer vision and pattern recognition' (2022), pp. 5501–5510

27 Keetha, N., Karhade, J., Jatavallabhula, K.M., *et al.*: 'SplaTAM: Splat, Track & Map 3D Gaussians for Dense RGB-D SLAM', http://arxiv.org/abs/2312.02126, accessed June 2024

28 Hu, J., Chen, X., Feng, B., *et al.*: 'CG-SLAM: Efficient Dense RGB-D SLAM in a Consistent Uncertainty-aware 3D Gaussian Field', http://arxiv.org/abs/2403.16095, accessed July 2024

29 Besl, P.J., McKay, N.D.: 'Method for registration of 3-D shapes', in 'Sensor fusion IV: Control paradigms and data structures' (Spie, 1992), pp. 586–606

30 Segal, A., Haehnel, D., Thrun, S.: 'Generalized-icp.', in 'Robotics: Science and systems' (Seattle, WA, 2009), p. 435

31 Park, J., Zhou, Q.-Y., Koltun, V.: 'Colored point cloud registration revisited', in 'Proceedings of the IEEE international conference on computer vision' (2017), pp. 143–152

32 Steinbrücker, F., Sturm, J., Cremers, D.: 'Real-time visual odometry from dense RGB-D images', in '2011 IEEE international conference on computer vision workshops (ICCV Workshops)' (IEEE, 2011), pp. 719–722

33 Pomerleau, F., Colas, F., Siegwart, R., Magnenat, S.: 'Comparing ICP variants on real-world data sets: Open-source library and experimental protocol'*Auton Robot*, 2013, **34**, (3), pp. 133–148.

34 Kuipers, J.B.: 'Quaternions and rotation sequences: A primer with applications to orbits, aerospace, and virtual reality' (Princeton university press, 1999)

35 Zwicker, M., Pfister, H., Van Baar, J., Gross, M.: 'EWA splatting'*IEEE Transactions on Visualization and Computer Graphics*, 2002, **8**, (3), pp. 223–238.

36 Kanopoulos, N., Vasanthavada, N., Baker, R.L.: 'Design of an image edge detection filter using the Sobel operator'*IEEE Journal of solid-state circuits*, 1988, **23**, (2), pp. 358–367.

37 Kingma, D.P.: 'Adam: A method for stochastic optimization' (2014)

38 Straub, J., Whelan, T., Ma, L., *et al.*: 'The Replica Dataset: A Digital Replica of Indoor Spaces', http://arxiv.org/abs/1906.05797, accessed August 2024

39 Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: 'A benchmark for the evaluation of RGB-D SLAM systems', in '2012 IEEE/RSJ international conference on intelligent robots and systems' (IEEE, 2012), pp. 573–580