

# GSplatLoc : Ultra-Precise Pose Optimization via 3D Gaussian Reprojection

<https://spla-tam.github.io>

Atticus Zhou, Atticus Zhou, Atticus Zhou, Atticus Zhou

July 20, 2024

## ABSTRACT

We present GSplatLoc, an innovative pose estimation method for RGB-D cameras that employs a volumetric representation of 3D Gaussians. This approach facilitates precise pose estimation by minimizing the loss based on the reprojection of 3D Gaussians from real depth maps captured from the estimated pose. Our method attains rotational errors close to zero and translational errors within 0.01mm, representing a substantial advancement in pose accuracy over existing point cloud registration algorithms, as well as explicit volumetric and implicit neural representation-based SLAM methods. Comprehensive evaluations demonstrate that GSplatLoc significantly improves pose estimation accuracy, which contributes to increased robustness and fidelity in real-time 3D scene reconstruction, setting a new standard for localization techniques in dense mapping SLAM.

## 1 Introduction

We present GSplatLoc, an innovative pose estimation method for RGB-D cameras that employs a volumetric representation of 3D Gaussians. This approach facilitates precise pose estimation by minimizing the loss based on the reprojection of 3D Gaussians from real depth maps captured from the estimated pose. Our method attains rotational errors close to zero and translational errors within 0.01mm, representing a substantial advancement in pose accuracy over existing point cloud registration algorithms, as well as explicit volumetric and implicit neural representation-based SLAM methods. Comprehensive evaluations demonstrate that GSplatLoc significantly improves pose estimation accuracy, which contributes to increased robustness and fidelity in real-time 3D scene reconstruction, setting a new standard for localization techniques in dense mapping SLAM.

## 2 Related Work

We present GSplatLoc, an innovative pose estimation method for RGB-D cameras that employs a volumetric representation of 3D Gaussians. This approach facilitates precise pose estimation by minimizing the loss based on the reprojection of 3D Gaussians from real depth maps captured from the estimated pose. Our method attains rotational errors close to zero and translational errors within 0.01mm, representing a substantial advancement in pose accuracy over existing point cloud registration algorithms, as well as explicit volumetric and implicit neural representation-based SLAM methods. Compre-

hensive evaluations demonstrate that GSplatLoc significantly improves pose estimation accuracy, which contributes to increased robustness and fidelity in real-time 3D scene reconstruction, setting a new standard for localization techniques in dense mapping SLAM.

## 3 Method

Depth-only Gaussian splatting is a highly effective method for modeling 3D scenes and producing depth maps. In our methodology, we initiate 3D Gaussians from a dense point cloud acquired via a RGB-D camera.

### 3.1 Gaussian Splatting

Let  $\mathcal{G} = \{G_i\}_{i=1}^N$  denote a set of  $N$  3D Gaussians, where each Gaussian  $G_i$  is parameterized as:

$$G_i = (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, o_i)$$

Here,  $\boldsymbol{\mu}_i \in \mathbb{R}^3$  represents the 3D mean,  $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3 \times 3}$  is the 3D covariance matrix, and  $o_i \in \mathbb{R}$  denotes the opacity. Initially, we set  $o_i = 1$  for all Gaussians to ensure full opacity.

The projection of a 3D Gaussian onto the 2D image plane is computed through a series of transformations:

$$\boldsymbol{\mu}_I = \pi(P(T_{wc}\boldsymbol{\mu}_{\text{homogeneous}}))$$

where  $T_{wc} \in SE(3)$  represents the world-to-camera transformation,  $P \in \mathbb{R}^{4 \times 4}$  is the projection matrix, and  $\pi : \mathbb{R}^4 \rightarrow \mathbb{R}^2$  maps to pixel coordinates.

The 2D covariance  $\Sigma_I \in \mathbb{R}^{2 \times 2}$  of a projected Gaussian is derived as:

$$\Sigma_I = J R_{wc} \Sigma R_{wc}^T J^T$$

where  $R_{wc}$  represents the rotation component of  $T_{wc}$ , and  $J$  is the affine transform as described in [1].

### 3.2 Depth Compositing

For depth map generation, we employ a front-to-back compositing scheme. This approach allows for accurate depth estimation and edge alignment. Let  $d_n$  represent the depth value associated with the  $n$ -th Gaussian, which is the z-coordinate of the Gaussian's mean in the camera coordinate system. The depth  $D(p)$  at pixel  $p$  is then computed as:

$$D(p) = \sum_{n \leq N} d_n \cdot \alpha_n \cdot T_n, \quad \text{where } T_n = \prod_{m < n} (1 - \alpha_m)$$

In this equation,  $\alpha_n$  represents the opacity of the  $n$ -th Gaussian at pixel  $p$ , which is computed as:

$$\alpha_n = o_n \cdot \exp(-\sigma_n), \quad \sigma_n = \frac{1}{2} \Delta_n^T \Sigma_I^{-1} \Delta_n$$

where  $\Delta_n$  is the offset between the pixel center and the 2D Gaussian center  $\mu_I$ , and  $o_n$  is the opacity parameter of the Gaussian.  $T_n$  denotes the cumulative transparency product of all Gaussians preceding  $n$ , accounting for the occlusion effects of previous Gaussians.

To normalize the depth values and ensure consistent representation across the image, we first calculate the total accumulated opacity  $\alpha(p)$  for each pixel:

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

The normalized depth  $\text{Norm}_D(p)$  is then defined as:

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

This normalization process ensures that the depth values are properly scaled and comparable across different regions of the image, regardless of the varying densities of Gaussians in the scene.

### 3.3 Camera Tracking

We define the camera pose as

$$\mathbf{T}_{cw} = \begin{pmatrix} \mathbf{R}_{cw} & \mathbf{t}_{cw} \\ \mathbf{0} & 1 \end{pmatrix} \in SE(3)$$

where  $\mathbf{T}_{cw}$  represents the camera-to-world transformation matrix. Notably, we parameterize the rotation  $\mathbf{R}_{cw} \in SO(3)$  using a quaternion  $\mathbf{q}_{cw}$ . This choice of parameterization is motivated by several key advantages that quaternions offer in the context of camera pose estimation and optimization. Quaternions provide a compact and efficient representation, requiring only four parameters, while maintaining numerical stability and avoiding singularities such as gimbal lock. Their continuous and non-redundant nature is particularly advantageous for gradient-based optimization algorithms, allowing for unconstrained optimization and simplifying the optimization landscape.

Based on these considerations, we design our optimization variables to separately optimize the normalized quaternion and the translation. The loss function is designed to ensure accurate depth estimations and edge alignment, incorporating both depth magnitude and contour accuracy. It can be defined as:

$$L = \lambda_1 \cdot L_{\text{depth}} + \lambda_2 \cdot L_{\text{contour}}$$

where  $L_{\text{depth}}$  represents the L1 loss for depth accuracy, and  $L_{\text{contour}}$  focuses on the alignment of depth contours or edges. Specifically:

$$L_{\text{depth}} = \sum_{i \in M} |D_i^{\text{predicted}} - D_i^{\text{observed}}|$$

$$L_{\text{contour}} = \sum_{j \in M} |\nabla D_j^{\text{predicted}} - \nabla D_j^{\text{observed}}|$$

Here,  $M$  denotes the reprojection mask, indicating which pixels are valid for reprojection. Both  $L_{\text{depth}}$  and  $L_{\text{contour}}$  are computed only over the masked regions.  $\lambda_1$  and  $\lambda_2$  are weights that balance the two parts of the loss function, tailored to the specific requirements of the application.

The optimization objective can be formulated as:

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

where  $\lambda_q$  and  $\lambda_t$  are regularization terms for the quaternion and translation parameters, respectively.

We employ the Adam optimizer for both quaternion and translation optimization, with different learning rates and weight decay values for each. The learning rates are set to  $5 \times 10^{-4}$

for quaternion optimization and  $10^{-3}$  for translation optimization, based on experimental results. The weight decay values are set to  $10^{-3}$  for both quaternion and translation parameters, serving as regularization to prevent overfitting.

## 4 Experiments

We present GSplatLoc, an innovative pose estimation method for RGB-D cameras that employs a volumetric representation of 3D Gaussians. This approach facilitates precise pose estimation by minimizing the loss based on the reprojection of 3D Gaussians from real depth maps captured from the estimated pose. Our method attains rotational errors close to zero and translational errors within 0.01mm, representing a substantial advancement in pose accuracy over existing point cloud registration algorithms, as well as explicit volumetric and implicit neural representation-based SLAM methods. Comprehensive evaluations demonstrate that GSplatLoc significantly improves pose estimation accuracy, which contributes to increased robustness and fidelity in real-time 3D scene reconstruction, setting a new standard for localization techniques in dense mapping SLAM.

## 5 Conclusion

We present GSplatLoc, an innovative pose estimation method for RGB-D cameras that employs a volumetric representation of 3D Gaussians. This approach facilitates precise pose estimation by minimizing the loss based on the reprojection of 3D Gaussians from real depth maps captured from the estimated pose. Our method attains rotational errors close to zero and translational errors within 0.01mm, representing a substantial advancement in pose accuracy over existing point cloud registration algorithms, as well as explicit volumetric and implicit neural representation-based SLAM methods. Comprehensive evaluations demonstrate that GSplatLoc significantly improves pose estimation accuracy, which contributes to increased robustness and fidelity in real-time 3D scene reconstruction, setting a new standard for localization techniques in dense mapping SLAM.

- [1] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, “EWA splatting,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 3, pp. 223–238, 2002, doi: 10.1109/TVCG.2002.1021576.