The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-1/W5-2025
13th International Conference on Mobile Mapping Technology (MMT 2025)
"Mobile Mapping for Autonomous Systems and Spatial Intelligence", 20–22 June 2025, Xiamen, China

OneStep-GSPE: an Efficient 3D Gaussian Splatting Based Image Pose Estimation

Yuhao Li[1, 2], Yipeng Lu[2], Jianping Li[3], Zhen Dong[2]*, Bisheng Yang[2]

[1] The School of Mechanical Engineering,
Chongqing Technology and Business University, Chongqing, China - yhaoli@whu.edu.cn
[2] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing,
Wuhan University, Wuhan, China - (luyipeng, dongzhenwhu, bshyang)@whu.edu.cn
[3] School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore - jianping.li@ntu.edu.sg

Keywords: Pose Estimation, 3D Gaussian Splatting, Dimension Lifting, Feature Matching.

Abstract

Accurate image pose estimation within a predefined map is critical for applications such as autonomous driving and urban infrastructure management. Conventional methods predominantly rely on feature correspondences, which often require the presence of specific object categories or involve computationally intensive feature learning processes. Recently, 3D Gaussian Splatting (3DGS) has emerged as a promising scene representation technique, offering high-fidelity novel view synthesis while preserving geometric accuracy. However, existing 3DGS-based pose estimation approaches are mainly tailored to small-scale indoor environments with limited lighting variation. Moreover, they typically rely on iterative optimization, which is computationally demanding and often fails to converge when the initial pose error is significant. This paper introduces OneStep-GSPE, a novel and efficient image pose estimation framework designed for outdoor environments with coarse initial poses. By integrating dense LiDAR priors into the 3DGS pipeline, the accuracy of Gaussian initialization is substantially improved, resulting in enhanced scene geometry reconstruction. Furthermore, rendered depth maps are utilized to lift 2D correspondences into 3D space, establishing 2D-3D matches for absolute pose estimation. The proposed method is category-agnostic and eliminates the need for iterative refinement, enabling fast and precise pose estimation. Experiments conducted on the KITTI-360 dataset demonstrate the effectiveness and robustness. OneStep-GSPE achieves a single-image pose estimation time of approximately 1.81 seconds, yielding over a 90% improvement in computational efficiency compared to the baseline. The project page is publicly available.

1. Introduction

In the domains of autonomous driving, infrastructure management, and high-definition (HD) map change detection, image pose estimation within pre-existing maps has remained a persistent research focus (Lambert and Hays, 2021; Zhanabatyrova et al., 2023; Li et al., 2025), as illustrated in Figure 1. Current image pose estimation methodologies can be broadly categorized into three paradigms: Structure-from-Motion (SfM) point cloud-based approaches, LiDAR point cloud-based approaches, and emerging techniques employing implicit or hybrid scene representations.

SfM-based methods predominantly depend on extensive image collections with strong co-visibility constraints to reconstruct 3D point clouds for camera pose estimation (Sarlin et al., 2019). However, images collected by mobile devices are often front-facing or sparsely sampled omnidirectional views, which exhibit uneven spatial distribution and short baselines. These limitations result in sparse and inaccurate reconstructions, ultimately degrading pose estimation performance. In contrast, LiDAR point clouds provide broader scene coverage and higher geometric fidelity (Li et al., 2023), making them a valuable reference for image localization. LiDAR-based pose estimation typically necessitates precise pixel-to-point correspondences (Li and Hee Lee, 2021; Wang et al., 2022), driving the development of dual-encoder architectures for cross-modal feature alignment. Nevertheless, the inherent modality discrepancy between photometric images and geometric point clouds persists as a fundamental challenge. Current solutions often require computationally intensive training procedures and demonstrate limited generalization capabilities across diverse data distributions (Kang et al., 2024).

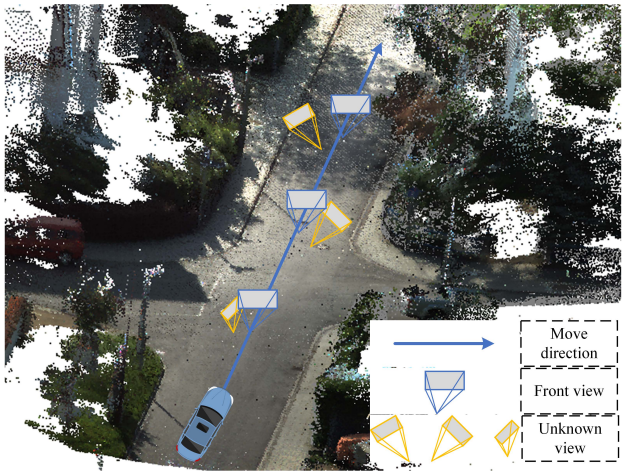


Figure 1. The illustration of the image pose estimation when the camera is mounted on a vehicle.

Implicit and hybrid scene representation methods, such as Neural Radiance Fields (NeRF) (Mildenhall et al., 2020) and 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023), have made significant progress in scene reconstruction. 3DGS integrates 3D scene geometry with 2D image, offering high-fidelity novel view synthesis capabilities and preserving rich geometric details. Con-

* Corresponding author

sequently, image pose estimation based on 3DGS scene representation has emerged as a promising approach. Currently, most 3DGS-based image pose estimation methods are designed for small-scale indoor environments with controlled illumination conditions. iComMa (Sun et al., 2024) incorporated image matching to recover the relative pose between images and optimizes image poses iteratively through joint rendering and comparison. However, existing approaches impose requirements on scene reconstruction quality and rely on iterative pose optimization to incrementally refine errors. When the initial pose error is large, the iterative process often struggles to converge and incurs high computational costs.

In this paper, OneStep-GSPE, a novel image pose estimation method, is designed for outdoor scenarios, which efficiently estimates image poses from an initial coarse pose without iterative refinement. By leveraging LiDAR point clouds and rendered depth, this approach enhances scene reconstruction while improving the robustness and accuracy of pose estimation. The main contributions of this paper can be summarized as:

1. The LiDAR point cloud priors are integrated into 3D Gaussian Splatting. The LiDAR priors ensure the position accuracy, providing a foundation for robust image pose estimation.

2. The 2D matching points are transformed into 3D space, and the image pose is estimated by converting 2D-2D correspondences into 2D-3D correspondences, enabling efficient and accurate pose estimation without iterative updates.

The remainder of this paper is organized as follows. Section 2 presents a review of image pose estimation based on implicit or hybrid scene representations. Section 3 elaborates on the proposed method in detail. The description of the dataset, implementation details, qualitative and quantitative results, and ablation studies are given in Section 4. Finally, the conclusions and future work are summarized in Section 5.

2. Related Works

Image pose estimation based on implicit representations originates from iNeRF (Yen-Chen et al., 2021), which fully leverages NeRF for rendering novel images. iNeRF introduced an object-level pose estimation approach based on inverse NeRF. However, due to the extensive iteration requirement, Chen et al. (2024) incorporated image matching, utilizing NeRF to render depth maps and lifting 2D correspondences between rendered and reference images into 3D space. The pose is then estimated by PnP and RANSAC. To improve efficiency, Sarlin et al. (2024) proposed a novel map representation that integrates multi-view street-level and aerial images to construct a 2D neural field map in a Bird's Eye View (BEV) perspective. They further designed a lightweight network that transforms monocular images into BEV feature maps and predicts image pose by comparing these features with the neural field map. Zhang et al. (2024) converted point clouds and images into NeRF representations, rendering images at predefined locations and constructing a feature database that includes both local and global descriptors for hierarchical pose estimation.

GaussianSplatting SLAM (Matsuki et al., 2024) is the first indoor visual SLAM system built upon the 3DGS scene representation. Given an input RGB-D stream, the method first represents the scene using 3DGS. By deriving the partial derivatives of 3DGS with respect to the image pose, it jointly optimizes the scene representation and camera poses, thereby achieving simultaneous 3DGS reconstruction and visual localization. Subsequent works such as GS-SLAM (Yan et al., 2024) and RTG-SLAM (Peng et al., 2024) introduced pruning and refinement strategies for the 3D Gaussians to improve rendering efficiency while maintaining real-time performance. DROID-Splat (Homeyer et al., 2024) integrated depth prediction and camera calibration modules to better balance robustness, speed, and accuracy. CG-SLAM (Hu et al., 2025) proposed an uncertainty-aware 3DGS by incorporating depth uncertainty, enabling the selection of valuable Gaussians and further enhancing computational efficiency. RGB-D GS-ICP SLAM (Ha et al., 2025) demonstrated that 3DGS and Generalized-ICP (GICP) (Segal et al., 2009) can share the same Gaussians during both tracking and mapping, thereby reducing redundant computation. iComMa (Sun et al., 2024) proposed an indoor image pose estimation method robust to large initial errors. It jointly optimizes the root mean square error and the feature matching loss between rendered and reference images via a gradient-based optimization strategy. GSLoc (Botashev et al., 2024) also adopted gradient backpropagation through the rendering pipeline and utilizes a coarse-to-fine optimization strategy to improve convergence. However, most of these methods are tailored for small-scale indoor RGB-D image pose estimation tasks and lack extensive experimental validation in outdoor environments.

3DGS-ReLoc (Jiang et al., 2024) proposed a memory-efficient 3DGS representation by omitting spherical harmonic coefficients. During outdoor visual localization, they retrieved the nearest Gaussian submaps using a KD-Tree and renders a set of images. The most similar image is found via normalized cross-correlation, after which a feature matching module estimates 2D correspondences. The rendered depth enables projecting 2D correspondences into 3D space. Finally, the image pose is recovered using PnP. GS-CPR (Liu et al., 2024) estimated 2D correspondences via a large visual foundation model and leveraged the 3DGS to refine the coarse image pose generated by the regression model. GigaSLAM (Deng et al., 2025) integrated the hierarchical sparse voxels and depth estimation model into Gaussian-based visual SLAM framework, achieving robust large-scale outdoor monocular pose estimation.

3. Methodology

The proposed OneStep-GSPE framework consists of two components: offline 3DGS construction and online image pose estimation, as illustrated in Figure 2. In the offline stage, a 3DGS model is constructed using LiDAR point clouds and viewpoint-limited images to represent the scene, enabling novel view synthesis. During online image pose estimation, given an initial pose with potential errors, the method renders RGB images and depth maps from the current viewpoint. Feature matching is then performed between the rendered and reference images to establish 2D correspondences. With the rendered depth, the matched 2D keypoints are lifted into 3D space, forming 2D-3D correspondences. This enables efficient and accurate image pose estimation.

3.1 LiDAR Point Cloud Assisted 3DGS Construction

The 3D Gaussian ellipsoids (Zwicker et al., 2001), defined as $\mathcal{G}_{\mathbf{N}} = \{\mathbf{G}_n \mid n = 1, ..., N\}$, serve as the fundamental primit-

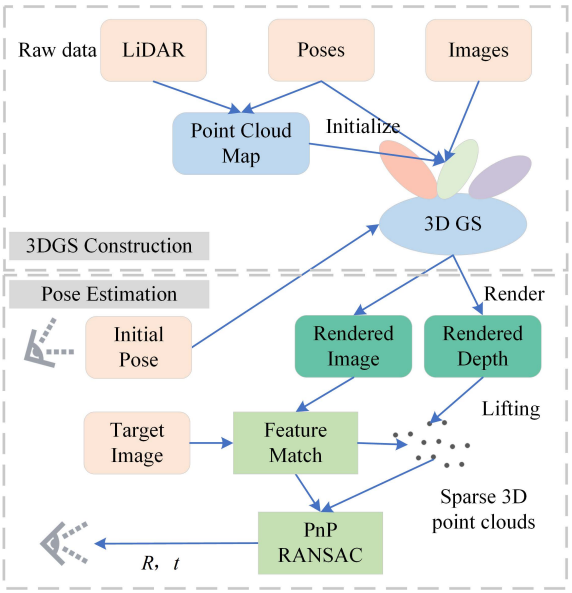


Figure 2. The pipeline of the OneStep-GSPE.

ives in 3DGS. The nth 3D Gaussian is defined as:

$$\mathbf{G}_n = \frac{1}{(2\pi)^{\frac{3}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_n^w)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_n^w)} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^3$ represents the position of a 3D point in the world coordinate system, $\boldsymbol{\mu}_n^w \in \mathbb{R}^3$ is the mean, indicating the position of the 3D Gaussian, and $\mathbf{\Sigma} \in \mathbb{R}^{3\times3}$ is the covariance matrix of the 3D Gaussian.

Both the means $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$ are set as learnable parameters to fit the scene adaptively. However, it is hard to optimize the covariance matrix through random initialization directly. Therefore, 3DGS re-parameterizes the covariance matrix into a scaling matrix $\mathbf{S} = \text{diag}(s_1, s_2, s_3) \in \mathbb{R}^{3\times3}$ and a rotation matrix $\mathbf{R} \in \text{SO}(3)$, such that $\mathbf{\Sigma} = \mathbf{RSS}^T\mathbf{R}^T$. Additionally, each 3D Gaussian includes an opacity $\alpha_n \in [0, 1]$ to represent occlusion relationships between ellipsoids and anisotropic spherical harmonic functions $\mathbf{f}_n \in \mathbb{R}^{3\times16}$ to express color, which allows for rendering viewpoint-dependent colors $\mathbf{c}_n \in \mathbb{R}^3$. In this paper, a third-order spherical harmonic function is used.

3DGS obtains images through the forward rendering of ellipsoid splatting. Specifically, the 3D Gaussians in the world coordinate system are first transformed into the camera coordinate system through the viewing transformation, represented as $\varphi : \mathbb{R}^3 \to \mathbb{R}^3$. The viewing transformation is linear and can be expressed as:

$$\mathbf{t} = \varphi(\boldsymbol{\mu}) = \mathbf{W}\boldsymbol{\mu} + \mathbf{d} \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{3\times3}$ is the rotation matrix in the viewing transformation, $\mathbf{d}$ is the translation, and $\mathbf{t}$ represents the position of the Gaussian in the camera coordinate system.

Subsequently, the 3D Gaussian in the camera coordinate system is projected onto the image plane as a 2D Gaussian through the projection transformation, formulated as $\phi : \mathbb{R}^3 \to \mathbb{R}^2$. The process of obtaining the 2D Gaussians $\mathbf{G}_n^{2D}$ is expressed as:

$$\mathbf{G}_n^{2D} = \varphi(\phi(\mathbf{G}_n)). \tag{3}$$

The projection transformation is nonlinear. Therefore, to facilitate optimization, it is approximated using an affine transformation via Taylor expansion. This involves computing the Jacobian matrix $\mathbf{J} \in \mathbb{R}^{2\times3}$, which represents the first-order partial derivatives of the projection function $\mathbf{P} \in \mathbb{R}^{2\times3}$. For the nth Gaussian, its Taylor expansion at $\mathbf{t}_n$ is given by:

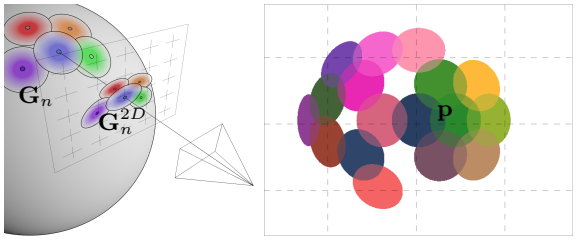


Figure 3. Schematic diagram of the 3D Gaussian splitting. (Left) the mapping from 3D Gaussians to 2D Gaussians. (Right) 2D Gaussians blended onto the image plane.

$$\phi_n(\mathbf{t}) = \mathbf{p}_n + \mathbf{J}_n \cdot (\mathbf{t} - \mathbf{t}_n), \tag{4}$$

where $\mathbf{p}_n = \phi(\mathbf{t}_n)$ denotes the coordinates of $\mathbf{t}_n$ on the image plane.

The position of the Gaussian in ray space can be expressed as:

$$\boldsymbol{\mu}_n = \phi_n(\varphi(\boldsymbol{\mu}_n^w)) = \mathbf{J}_n\mathbf{W}\boldsymbol{\mu}_n^w + \mathbf{x}_n + \mathbf{J}_n(\mathbf{d} - \mathbf{t}_n). \tag{5}$$

The covariance matrix of the 2D Gaussian $\mathbf{\Sigma}' \in \mathbb{R}^{2\times2}$ is given by:

$$\mathbf{\Sigma}' = \mathbf{JW\Sigma W}^T\mathbf{J}^T. \tag{6}$$

Finally, images are generated using α-blending, as shown in Figure 3. The entire Gaussian splatting process is differentiable and can be parallelized on the GPU, making the rendering highly efficient.

By leveraging the photometric error between the rendered image I_r and the reference image I_{gt}, defined as $L_1 = \|I_r - I_{gt}\|_1$, along with the structural similarity loss L_{D-SSIM}, these learnable parameters are optimized through gradient backpropagation. The overall loss function L is formulated as follows:

$$L = (1 - \lambda)L_1 + \lambda L_{D-SSIM}. \tag{7}$$

where λ is the balance factor.

In vanilla 3DGS, the Structure-from-Motion (SfM) is employed to convert sequential images into sparse 3D point clouds, which serve as the initial positions of the 3D Gaussians. However, due to the sparsity of onboard images and their predominantly forward-facing viewpoints, SfM fails to accurately reconstruct sparse 3D points. To address this limitation, this paper proposes integrating dense LiDAR point clouds into the 3DGS to jointly initialize the positions of the 3D Gaussians, thereby improving reconstruction quality and scene representation.

During 3DGS optimization, the Gaussian positions are continuously refined through expansion and pruning. For instance, when the gradient of the position becomes excessively large, an existing Gaussian splits into two; conversely, if the opacity of a Gaussian approaches zero, it is removed. By integrating dense LiDAR point clouds, the 3DGS not only captures anisotropic color but also achieves more accurate spatial positioning.

3.2 Depth Lifting based One-Step Pose Estimation

The estimation of absolute pose in this paper involves rendering the image I_r and depth D_r from the initial pose. First, image matching is used to establish the 2D correspondence between the rendered image I_r and the reference image I_t. Thanks to the aforementioned 3DGS representation, which integrates LiDAR point clouds, it can not only render RGB images but also generate accurate depth maps. Then, based on the rendered depth, the 2D correspondences are lifted to 3D space, establishing the 2D-3D correspondences. The image poses in the world coordinate system, $\hat{\mathbf{T}} \in \mathrm{SE}(3)$, is finally solved using the PnP. The pose estimation consists of the following stages: 1) image matching, 2) 2D-3D lifting, and 3) pose solving. The image pose estimation based on the 3DGS can be represented as Equation 8:

$$\hat{\mathbf{T}} = \mathrm{PnP}(I_t, \mathrm{Lift}(\mathrm{Render}(\mathbf{T}, \mathbf{K}, \Theta), D)), \tag{8}$$

where Lift() is the process of lifting from 2D to 3D, Render() represents rendering process, $\mathbf{T}$ is the initial pose, $\mathbf{K}$ is the camera intrinsic, Θ represents parameters of the 3DGS, and D is the depth corresponding to the 2D feature points.

(1) Image Matching: Based on the 3DGS constructed in Section 3.1, the image at the initial pose $\mathbf{T}$ could be rendered. The relative pose between the rendered image I_r and the reference target I_t images could be estimated by feature matching. And the 2D correspondences could be represented as $[\mathbf{p}_i, \mathbf{q}_i], i \in M$, as shown in Equation 9:

$$[\mathbf{p}_i, \mathbf{q}_i] = \mathrm{Matcher}(I_t, I_r), \tag{9}$$

where Matcher() represents any image matching method, $\mathbf{p}_i$ is the keypoint in the reference image, and the corresponding keypoint $\mathbf{q}_i$ is from the rendered image. M is the number of matching points.

(2) 2D-3D Lifting: Analogous to the rendering of RGB images, the rendering of the depth D_r can be expressed as:

$$D_r = \sum_{n=1}^{\mathbf{N}} \alpha_n d_i \mathbf{G}_n^{2D}(\mathbf{x}) \prod_{t=1}^{n-1} \left(1 - \alpha_t \mathbf{G}_t^{2D}(\mathbf{x})\right), \tag{10}$$

where d_i represents the depth corresponding to each 3D Gaussian in the camera coordinate system.

The process of lifting 2D points to 3D space is represented by equation 11:

$$\mathbf{P}_i = \mathbf{T}^{-1}(\hat{z}_i \mathbf{K}^{-1} \mathbf{q}_i), \tag{11}$$

where $\hat{z}_i \in D_r$ is the depth corresponding to the point $\mathbf{q}_i$, and $\mathbf{T}^{-1}()$ indicates the transformation of a point from the camera coordinates to the world coordinates.

(3) Pose Solving: After establishing the 2D-3D correspondences, the optimal image pose is estimated using PnP (Lepetit et al., 2009) and RANSAC, and the whole process can be expressed as:

$$\hat{\mathbf{T}} = \underset{\mathbf{T} \in \mathrm{SE}(3)}{\arg\min} \|\mathbf{p} - \boldsymbol{\pi}(\mathbf{P}, \mathbf{K}, \mathbf{T})\|_2, \tag{12}$$

where $\boldsymbol{\pi}(\mathbf{P}, \mathbf{K}, \mathbf{T})$ represents the projection of the 3D point $\mathbf{P}$ onto the image plane.

The pseudocode of the pose estimation is presented in Algorithm 1:

Algorithm 1 Pseudocode of image pose estimation

Input: Camera intrinsics $K := \{f_x, f_y, c_x, c_y\}$, depth D_r, reference image I_t in the world coordinate system
Output: $\hat{\mathbf{T}}$ (image pose in the world coordinate system)

for each pair of 2D matched points $i = 1, 2, \ldots, N$ **do**
  $q_u, q_v \leftarrow$ keypoint $\mathbf{q}_i$ in the current image I_r
  $p_u, p_v \leftarrow$ keypoint $\mathbf{p}_i$ in the reference image I_t
  Depth $P_z \leftarrow D[v][u]$
  $P_x \leftarrow (q_u - c_x) * P_z / f_x$
  $P_y \leftarrow (q_v - c_y) * P_z / f_y$
  if matching confidence $\omega > 0$ **then**
    3D point $\leftarrow [P_x, P_y, P_z]$
    2D pixels $\leftarrow [p_u, p_v]$
  end if
end for
if total number of 2D pixels ≥ 30 **then**
  $\hat{\mathbf{T}}$, success status $\leftarrow$ PnP(3D points, 2D pixels)
  if successful **then**
    return $\hat{\mathbf{T}}$
  end if
end if

4. Experiments

4.1 Dataset and Evaluation Metrics

The 00 and 09 sequences of the KITTI-360 (Liao et al., 2023) dataset are selected to evaluate the performance of the proposed method. The left view of the stereo image is selected as the input image, with a resolution of 376×1408. The sequences are divided into a training set and a test set. For all sequence images, one out of every eight images is selected and added to the test set. The images and point clouds from the training set are used to construct the 3DGS representation model, while the test set is used to evaluate the rendering quality from novel image synthesis and assess pose estimation accuracy.

The peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and learned perceptual image patch similarity (LPIPS) are selected to evaluate the rendering performance of the 3DGS. This paper uses accuracy (success rate), average rotation error (ARE), and average translation error (ATE) to evaluate the performance of pose estimation. The pose estimation result is considered accurate when the rotational error between the estimated pose and the ground truth pose is less than δ_1, and the translational error is less than δ_2. The rotation error is computed using the following formula:

$$Error_{\mathbf{R}} = \arccos\left(\frac{\mathrm{tr}(\Delta\mathbf{R}) - 1}{2}\right), \tag{13}$$

where $\Delta\mathbf{R}$ denotes the relative rotation between the estimated and the ground truth pose, and tr denotes the trace of the matrix.

The translation error is computed as follows:

$$Error_{\mathbf{t}} = ||\Delta\mathbf{t}||_2, \tag{14}$$

where $\Delta\mathbf{t}$ represents the translation difference between the estimated and the ground truth pose, and $||\cdot||_2$ denotes the Euclidean norm.

4.2 Implementation

The 3DGS are optimized using an NVIDIA RTX 4090 24GB GPU. The dense LiDAR point clouds are used for Gaussian position initialization, with the color parameters initialized to 0. The learning rates for position, opacity, scale, rotation, and SH

coefficients are set to 0.000016, 0.05, 0.001, 0.001, and 0.0025, respectively. Similar to the settings in 3DGS, the parameters at the 30,000th iteration are chosen as the final model.

In this paper, image poses with initial errors are constructed by randomly rotating around the x, y, and z axes within the range of $[-20, 20]$ degrees, and by adding random perturbations within $[-1, 1]$ meters along the x, y, and z axes. These erroneous poses are used as initial image poses, which are then subjected to the pose estimation method.

Any image matching methods could establish the 2D correspondences, where two of them are selected in this paper: LightGlue (Lindenberger et al., 2023), which is based on sparse keypoints, and LoFTR (Sun et al., 2021), which is based on dense keypoints. Both methods implement 2D correspondence matching using self-attention (Vaswani et al., 2017) and cross-attention mechanisms. LightGlue uses sparse feature point matching, following the "keypoint extraction, feature descriptor extraction, feature matching" paradigm and is highly efficient. LoFTR, on the other hand, uses a keypoint-free paradigm and implements pixel-level dense matching with a coarse-to-fine matching strategy, which is more robust in texture-poor scenarios. It is noteworthy that the proposed OneStep-GSPE is compatible with different types of image matching methods.

In this paper, LightGlue (Lindenberger et al., 2023) is employed to establish accurate 2D correspondences between the rendered and the reference image. The number of keypoints extracted is set to 2048, and if the number of matching keypoints between two images is fewer than 30, the match is considered a failure.

4.3 Baseline

The pose-gradient optimization-based method iComMa (Sun et al., 2024) is selected for comparative experiments. This method treats pose estimation as an inverse operation for 3DGS reconstruction. Given a 3DGS model of an indoor scene, the 3DGS parameters are fixed, and only the image pose is optimized. iComMa first establishes 2D correspondences between the rendered image and the reference image, while jointly minimizing the photometric error between the rendered and the reference image. It iteratively estimates the optimal pose by optimizing the relative pose increment between the current pose and the optimal pose. This updated pose is then used to render new images, proceeding to the next iteration until the maximum number of iterations is reached. In this paper, the number of iterations is set to 300. The iComMa does not account for the influence of lighting factors on pose optimization, resulting in relatively poor estimation robustness in outdoor scenes.

4.4 Quantitative Results

The quantitative results in Table 1 indicate that integrating dense LiDAR point cloud prior achieves better rendering results. For both sequences, SSIM and PSNR show significant improvements. In terms of the LPIPS, the images rendered by the 3DGS are nearly indistinguishable from the ground truth images, with the feature encoder almost unable to distinguish between them.

Table 2 reports the pose estimation accuracy, ARE, and ATE on the test dataset. The same evaluation criteria are applied across all methods, where cases with fewer than 30 matched points are treated as failed matches. OneStep-GSPE achieves the best performance, with rotation and translation errors below 0.1 degrees and 2 cm, respectively. As shown in the table, OneStep-GSPE

Table 1. The novel image synthesis results of KITTI-360 dataset.

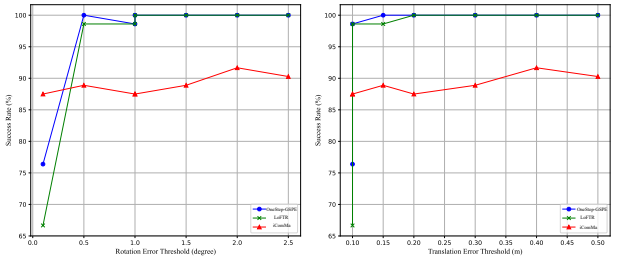
Method	00 Sequence			09 Sequence		
	SSIM ↑	**PSNR ↑**	**LPIPS ↓**	**SSIM ↑**	**PSNR ↑**	**LPIPS ↓**
3DGS (SfM)	0.798	23.99	0.259	0.835	24.89	0.225
3DGS (LiDAR)	0.805	23.55	0.228	0.842	24.90	0.196
OneStep-GSPE (SfM+LiDAR)	**0.811**	**24.22**	**0.220**	**0.847**	**25.17**	**0.189**

Table 2. The quantitative results of the evaluation dataset. The accurate estimation is defined when the rotation and translation error is smaller than $\delta_1 = 1$ degree and $\delta_2 = 0.1$ m separately.

Method	accuracy (% ↑)	ARE (deg)(↓)	ATE (m)(↓)	time per image (s)(↓)
iComMa	87.50	**0.042**	**0.009**	59.6
LoFTR	**98.61**	0.090	0.017	3.20
OneStep-GSPE	**98.61**	0.089	0.016	**1.81**

Table 3. Average time cost of pose estimation for single image.

Module	Rendering	Matching	Lifting	PnP&RANSAC
Time cost (ms)	42.2	137.6	0.7	0.6



(a) Different rotation thresholds (b) Different translation thresholds

Figure 4. The success rate curves at different rotation and translation thresholds.

is compatible with different image matching algorithms, and both variants yield nearly identical quantitative results.

The table also presents a comparison of runtime. Pose estimation for a single image requires two rendering and one matching step, with rendering accelerated by GPU implementation, resulting in high overall efficiency. Both LofTR and LightGlue exhibit high matching efficiency, significantly outperforming iComMa, which relies on iterative optimization. Consequently, this work adopts the more efficient LightGlue, achieving an average runtime of approximately 1.81 seconds per image, representing over 90% improvement in computational efficiency compared to the iComMa baseline.

To further evaluate the effectiveness of OneStep-GSPE, different rotation and translation thresholds are used to assess its performance. The corresponding success rate curves on the KITTI-360 dataset are shown in Figures 4. Under various thresholds, OneStep-GSPE consistently outperforms the iComMa baseline in most settings.

Finally, Table 3 provides a detailed breakdown of the time cost for each component of OneStep-GSPE, averaged across the test set. It is evident that the time required for pose computation is nearly negligible.

4.5 Qualitative Results

The distribution of 3D Gaussians is illustrated in Figure 5. As expected, the 3DGS representation constructed by the proposed

Figure 5. 3D Gaussian ellipsoids.

(a) Reference images

(b) Rendered RGB images from initial pose

(c) Rendered depths from initial pose

Figure 6. The rendered images and depths from the initial pose.

(a) Vanilla 3DGS

(b) 3DGS initialized by LiDAR point clouds

(c) OneStep-GSPE

(d) Groud truth reference image

Figure 7. Visualization of the novel synthesis. The red box highlights the rendering quality differences between the images.

method aligns well with the geometric structure of the scene. In regions lacking geometric texture, the Gaussians tend to be larger, whereas in areas rich in geometric detail, the Gaussians are smaller and more densely distributed, effectively capturing scene intricacies.

The sequence images in the KITTI-360 dataset are predominantly front-facing, and the rendering quality is primarily evaluated under similar viewing conditions, with limited viewpoint variation. Nevertheless, as highlighted in the red boxes of Figure 7, the rendered images produced using the proposed LiDAR point cloud-integrated 3DGS initialization are visually much

Figure 8. The overlay comparison of the rendered images from both the initial and estimated pose with the reference image. Left is from the initial pose, while the right froms the estimated pose.

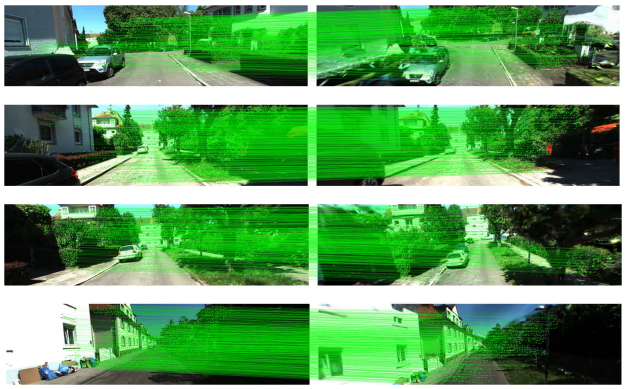


Figure 9. The 2D feature correspondences between rendered and reference images.

closer to the reference images, compared to those initialized using SfM-based 3DGS.

Figure 6 presents the RGB images and corresponding depth maps rendered from the initial pose. The rendered depths align well with the actual spatial structure of the scene, further validating the effectiveness of incorporating LiDAR point cloud priors into 3DGS initialization. These accurate depth maps provide a reliable basis for subsequent pose estimation. To further verify pose estimation accuracy, this paper overlays the images rendered from both the initial and estimated poses with the reference image, as shown in Figure 8. The absence of ghosting artifacts in the overlay confirms the high precision of the estimated poses.

Figure 9 shows the matched keypoints between the rendered and reference images using LightGlue, with green lines indicating the correspondences. The high image quality of novel views rendered from the reconstructed 3DGS allows the system to tolerate significant initial pose errors. Even under large initial errors, the rendered images remain sufficient for reliable image matching, enabling the extraction of stable keypoint correspondences and thus enhancing the robustness of the pose estimation process.

4.6 Compared to the Baseline

As shown in Table 2, iComMa demonstrates lower accuracy in image pose estimation for outdoor scenes. This performance degradation is primarily attributed to its strong reliance on the

(a) Initial

(b) iComMa-300th

(c) OneStep-GSPE

Figure 10. The overlay comparison of the baseline iComMa and proposed OneStep-GSPE.

quality of 3DGS scene reconstruction. In addition, the 2D image matching in iComMa provides insufficient geometric constraints, often resulting in failure to converge to the globally optimal pose during iterative optimization. Moreover, iComMa does not explicitly account for illumination changes and viewpoint variations common in outdoor environments. As illustrated in Figure 10b, even after 300 iterations, the rendered and reference images remain misaligned, indicating a failure in pose optimization.

In contrast, the proposed OneStep-GSPE eliminates the need for iterative optimization and can efficiently and accurately estimate the correct pose. Notably, iComMa also consumes a significant amount of GPU memory during optimization, whereas the OneStep-GSPE enables fast computation, making it well-suited for real-time pose estimation on mobile devices.

5. Conclusion

This paper presents OneStep-GSPE, an efficient image pose estimation framework based on 3D Gaussian Splatting as the scene representation. To improve the quality of novel view synthesis and provide a more reliable foundation for pose estimation, dense LiDAR point clouds are integrated for initializing Gaussian positions. Furthermore, a lightweight depth-lifting strategy is introduced to establish accurate 2D–3D feature correspondences, enabling fast and precise pose estimation. The proposed method is category-agnostic and avoids costly iterative optimization, making it both efficient and broadly applicable. Experiments on the KITTI-360 dataset demonstrate the effectiveness, accuracy, and robustness of OneStep-GSPE in real-world urban scenarios. In future work, we plan to incorporate a cross-modal coarse image pose estimation module to build a closed-loop localization system, further improving its practicality for autonomous driving applications. Moreover, as an emerging paradigm for scene representation, 3DGS shows great potential for integration with traditional modeling approaches, contributing to ubiquitous perception, large-scale 3D reconstruction, and digital twin-enabled smart city development.

References

Botashev, K., Pyatov, V., Ferrer, G., Lefkimmiatis, S., 2024. GSLoc: Visual Localization with 3D Gaussian Splatting.

Chen, R., Cong, Y., Ren, Y., 2024. Marrying NeRF with Feature Matching for One-step Pose Estimation. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 7302–7309.

Deng, K., Yang, J., Wang, S., Xie, J., 2025. GigaSLAM: Large-scale monocular SLAM with hierachical gaussian splats.

Ha, S., Yeon, J., Yu, H., 2025. RGBD GS-ICP SLAM. A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, G. Varol (eds), *Computer Vision – ECCV 2024*, Springer Nature Switzerland, Cham, 180–197.

Homeyer, C., Begiristain, L., Schnörr, C., 2024. DROID-Splat: Combining end-to-end SLAM with 3D Gaussian Splatting.

Hu, J., Chen, X., Feng, B., Li, G., Yang, L., Bao, H., Zhang, G., Cui, Z., 2025. CG-SLAM: Efficient Dense RGB-D SLAM in a Consistent Uncertainty-Aware 3D Gaussian Field. A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, G. Varol (eds), *Computer Vision – ECCV 2024*, Springer Nature Switzerland, Cham, 93–112.

Jiang, P., Pandey, G., Saripalli, S., 2024. 3DGS-ReLoc: 3D Gaussian Splatting for Map Representation and Visual ReLocalization.

Kang, S., Liao, Y., Li, J., Liang, F., Li, Y., Zou, X., Li, F., Chen, X., Dong, Z., Yang, B., 2024. CoFiI2P: Coarse-to-Fine Correspondences-Based Image to Point Cloud Registration. *IEEE Robotics and Automation Letters*, 9(11), 10264–10271.

Kerbl, B., Kopanas, G., Leimkuehler, T., Drettakis, G., 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4), 139:1–139:14.

Lambert, J., Hays, J., 2021. Trust, but verify: Cross-modality fusion for HD map change detection. *Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Lepetit, V., Moreno-Noguer, F., Fua, P., 2009. EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision*, 81(2), 155–166.

Li, J., Hee Lee, G., 2021. DeepI2P: Image-to-Point Cloud Registration via Deep Classification. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15955–15964.

Li, Y., Li, J., Dong, Z., Wang, Y., Yang, B., 2025. SaliencyI2PLoc: Saliency-guided Image–Point Cloud Localization Using Contrastive Learning. *Information Fusion*, 118, 103015.

Li, Y., Zou, X., Li, T., Sun, S., Wang, Y., Liang, F., Li, J., Yang, B., Dong, Z., 2023. MuCoGraph: A Multi-Scale Constraint Enhanced Pose-Graph Framework for MLS Point Cloud Inconsistency Correction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 204, 421–441.

Liao, Y., Xie, J., Geiger, A., 2023. KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3), 3292–3310.

Lindenberger, P., Sarlin, P.-E., Pollefeys, M., 2023. LightGlue: Local Feature Matching at Light Speed. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 17581–17592.

Liu, C., Chen, S., Bhalgat, Y. S., Hu, S., Cheng, M., Wang, Z., Prisacariu, V. A., Braud, T., 2024. GS-CPR: Efficient camera pose refinement via 3D gaussian splatting. *The Thirteenth International Conference on Learning Representations*.

Matsuki, H., Murai, R., Kelly, P. H. J., Davison, A. J., 2024. Gaussian Splatting SLAM. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., Ng, R., 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (eds), *Computer Vision – ECCV 2020*, 12346, Springer International Publishing, Cham, 405–421.

Peng, Z., Shao, T., Liu, Y., Zhou, J., Yang, Y., Wang, J., Zhou, K., 2024. RTG-SLAM: Real-time 3D Reconstruction at Scale using Gaussian Splatting. *ACM SIGGRAPH 2024 Conference Papers*, SIGGRAPH '24, Association for Computing Machinery, New York, NY, USA, 1–11.

Sarlin, P.-E., Cadena, C., Siegwart, R., Dymczyk, M., 2019. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12708–12717.

Sarlin, P.-E., Trulls, E., Pollefeys, M., Hosang, J., Lynen, S., 2024. SNAP: Self-supervised neural maps for visual positioning and semantic understanding. *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Curran Associates Inc., Red Hook, NY, USA, 7697–7729.

Segal, A., Haehnel, D., Thrun, S., 2009. Generalized-ICP. *Robotics: Science and Systems V*, 25.

Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X., 2021. LoFTR: Detector-Free Local Feature Matching with Transformers. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8918–8927.

Sun, Y., Wang, X., Zhang, Y., Zhang, J., Jiang, C., Guo, Y., Wang, F., 2024. iComMa: Inverting 3D Gaussian Splatting for Camera Pose Estimation via Comparing and Matching.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., 2017. Attention is All you Need. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (eds), *Advances in Neural Information Processing Systems*, 30, Curran Associates, Inc.

Wang, Y., Li, Y., Chen, Y., Peng, M., Li, H., Yang, B., Chen, C., Dong, Z., 2022. Automatic Registration of Point Cloud and Panoramic Images in Urban Scenes Based on Pole Matching. *International Journal of Applied Earth Observation and Geoinformation*, 115, 103083.

Yan, C., Qu, D., Xu, D., Zhao, B., Wang, Z., Wang, D., Li, X., 2024. GS-SLAM: Dense Visual SLAM with 3D Gaussian Splatting. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19595–19604.

Yen-Chen, L., Florence, P., Barron, J. T., Rodriguez, A., Isola, P., Lin, T.-Y., 2021. iNeRF: Inverting Neural Radiance Fields for Pose Estimation. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1323–1330.

Zhanabatyrova, A., Souza Leite, C. F., Xiao, Y., 2023. Automatic Map Update Using Dashcam Videos. *IEEE Internet of Things Journal*, 10(13), 11825–11843.

Zhang, L., Tao, Y., Lin, J., Zhang, F., Fallon, M., 2024. Visual Localization in 3D Maps: Comparing Point Cloud, Mesh, and NeRF Representations.

Zwicker, M., Pfister, H., van Baar, J., Gross, M., 2001. Surface splatting. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, Association for Computing Machinery, New York, NY, USA, 371–378.