This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

A Constrained Optimization Approach for Gaussian Splatting from Coarsely-posed Images and Noisy Lidar Point Clouds

Jizong Peng[1*], Tze Ho Elden Tse[2*], Kai Xu[2], Wenchao Gao[1], Angela Yao[2]
[1]dConstruct Robotics [2]National University of Singapore
{jizong.peng,wehchao.gao}@dconstruct.ai {eldentse,kxu,ayao}@comp.nus.edu.sg

Abstract

3D Gaussian Splatting (3DGS) is a powerful reconstruction technique; however, it requires initialization from accurate camera poses and high-fidelity point clouds. Typically, the initialization is taken from Structure-from-Motion (SfM) algorithms; however, SfM is time-consuming and restricts the application of 3DGS in real-world scenarios and large-scale scene reconstruction. We introduce a constrained optimization method for simultaneous camera pose estimation and 3D reconstruction that does not require SfM support. Core to our approach is decomposing a camera pose into a sequence of camera-to-(device-)center and (device-)center-to-world optimizations. To facilitate, we propose two optimization constraints conditioned on the sensitivity of each parameter group and restricts the search space of each parameter. In addition, as we learn the scene geometry directly from the noisy point clouds, we propose geometric constraints to improve the reconstruction quality. Experiments demonstrate that the proposed method significantly outperforms the existing (multi-modal) 3DGS baseline and methods supplemented by COLMAP on both our collected dataset and two public benchmarks. Project webpage: https://eldentse.github.io/contrained-optimization-3dgs.

1. Introduction

Simultaneous localization and mapping (SLAM) is critical for robotics and AR/VR applications. Traditional SLAM approaches [8, 13, 28] are reasonably accurate in localization but struggle to produce dense 3D maps with fine-grained detailing. Recently, 3D Gaussian Splatting (3DGS) [17] has shown great promise for fast and high-quality rendering. As a result, there is increasing interest in combining 3DGS with SLAM [10, 16, 23, 33, 38]. One way is to incorporate SLAM for 3DGS initialization as a faster alternative to Structure-from-Motion (S*f*M) algorithms.

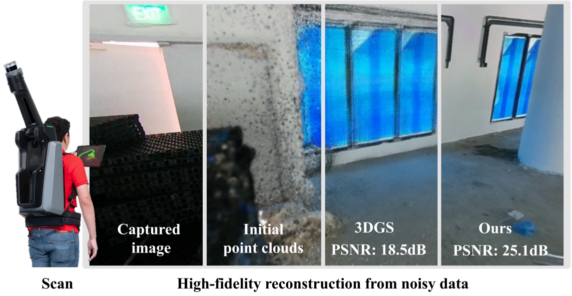

Figure 1. Given noisy point clouds and inaccurate camera poses, our constrained optimization approach reconstructs the 3D scene in Gaussian Splatting with high visual quality.

Yet standard SLAM systems produce only rough camera pose estimates and noisy point clouds. Additionally, less-than-perfect camera intrinsics and Lidar-to-camera extrinsic calibration introduce errors and uncertainty into the 3D reconstruction. Directly using such SLAM inputs results in blurry reconstructions and degraded geometry (see Fig. 1) for standard 3DGS methods. While the SLAM outputs can be enhanced by additional hardware [7, 14], this invariably increases hardware costs and acquisition time.

This paper addresses the challenge of training 3DGS under imprecise initialization conditions, *i.e.* inaccurate sensor calibration and approximate camera pose estimation. We consider inputs from a typical 3D scanning setup, comprising multiple RGB cameras, a Lidar, and an inertial motion unit (IMU) within a rigid body framework. In the absence of S*f*M support, we introduce a constrained optimization method for simultaneously estimating camera parameters and reconstructing 3D scenes. Specifically, our constrained optimization strategies are targeted at refining the extrinsics and intrinsics of the multi-camera setup, as well as 3DGS.

To achieve this, we first decouple multi-camera poses into a sequence of camera-to-(device-) center and (device-) center-to-world transformations. However, simply optimizing for camera parameters and scene reconstruction can result in sub-optimal solutions for two main reasons. First, there is inherent ambiguity in the perspective projection; the intrinsic parameters and camera poses describe relative

[*] Equal contribution

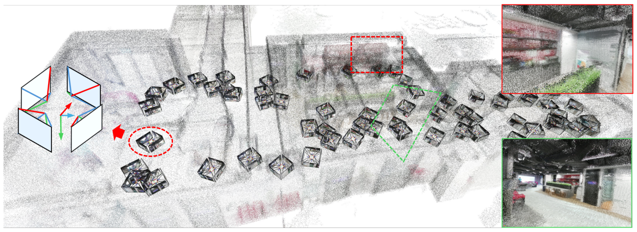

Figure 2. Qualitative example of camera poses and colored point clouds obtained from our multi-camera SLAM system.

and nonlinear relationships that can lead to multiple feasible solutions. Secondly, the ensemble camera poses are over-parameterized; adjusting one camera's orientation is equivalent to altering that of all device centers, creating unnecessary redundancy for optimization.

To address this problem, we precondition our optimization based on the sensitivity of each parameter group. We also employ a log-barrier method to ensure that critical parameters remain within a predefined feasibility region (*e.g.* focal length should not deviate by 2%). To further improve the quality of scene reconstructions, we propose two geometric constraints to serve as a strong regularization in the image space. Specifically, inspired by S*f*M algorithms, we introduce a soft epipolar constraint and a reprojection regularizer for robust training to mitigate noisy camera poses.

There are no existing benchmarks fitting to this problem setting, so we curate a new dataset featuring complex indoor and large-scale outdoor scenes. As illustrated in Fig. 2, our proposed dataset is captured with 4 RGB cameras, an IMU, and Lidar. We run an extensive ablation study as well as comparisons with state-of-the-art methods. Our experiments demonstrate that our constrained optimization approach is efficient and effective.

In summary, our contributions are:

- The first constrained optimization approach for training 3DGS that refines poor camera and point cloud initialization from a multi-camera SLAM system.
- We derive and enable refinement of camera intrinsics, extrinsics, and 3DGS scene representation using four of our proposed optimization constraints.
- A new dataset capturing complex indoor and large-scale outdoor scenes from hardware featuring multiple RGB cameras, IMU, and Lidar.
- Our approach achieves competitive performance against existing 3DGS methods that rely on COLMAP, but with significantly less pre-processing time.

2. Related Work

3D reconstruction. 3D reconstruction from multi-view images is a fundamental problem in computer vision. Traditional methods use complex multi-stage pipelines involving feature matching, depth estimation [24], point cloud fusion [5], and surface reconstruction [15]. In contrast, neural implicit methods such as NeRF [25] simplify this process by optimizing an implicit surface representation through volumetric rendering. Recent advancements include more expressive scene representations via advanced training strategies [4] and monocular priors [9]. However, these methods are often limited to foreground objects and are computationally intensive. More recently, 3DGS has been proposed as an efficient point-based representation for complex scenes. While all the aforementioned methods require accurate camera poses, 3DGS also requires a geometrically accurate sparse point cloud for initialization. This research addresses the challenges posed by inaccurate point clouds and camera poses to achieve a high-quality static reconstruction.

Camera pose optimization. Recently, there has been growing interest in reducing the need for accurate camera estimation, often derived from S*f*M. Initial efforts like i-NeRF [40] predict camera poses by matching keypoints using a pre-trained NeRF. Subsequently, NeRF$--$ [37] jointly optimizes the NeRF network and camera pose embeddings. BARF [21] and GARF [6] address the gradient inconsistency issue from high-frequency positional embeddings, with BARF using a coarse-to-fine positional encoding strategy for joint optimization. In the 3DGS field, iComMa [34] employs an iterative refinement process for camera pose estimation by inverting 3DGS, while GS-CPR [22] uses visual foundation models for pose optimization with accurate key-point matches. However, these methods assume a high-quality pre-trained 3DGS model and are computationally inefficient. In contrast, our method jointly optimizes camera poses and reconstruction through constrained optimization.

SLAM with 3DGS. The integration of 3DGS has garnered significant interest in the field of SLAM [10, 16, 23, 33, 38], serving as an efficient representation of 3D scenes. Methods in this domain offer several advantages, including continuous surface modeling, reduced memory usage, and improved gap filling and scene inpainting for partially observed or occluded data. In contrast, some work extends SLAM outputs to photometric reconstructions [7, 41, 42] by assuming accurate poses and point clouds due to complex hardware [7, 42] or multiple capture sequences [7]. In this paper, we consider coarsely estimated poses and noisy point clouds from a multi-camera SLAM system to achieve highly accurate 3D scene reconstruction.

Multimodal 3DGS. There has been an increasing interest in reconstruction using multimodal data [18, 20], particularly for autonomous driving. For instance, [39, 43] combine images with Lidar, though they rely on COLMAP for refining camera poses. Additionally, [39] optimizes camera poses independently without intrinsic parameter refinement. In contrast, we are the first to introduce a constrained optimization framework that refines intrinsic and extrinsic parameters of (multiple) cameras under various constraints.

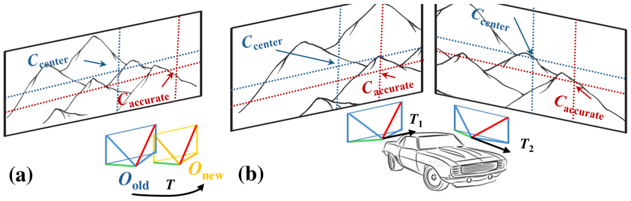

Figure 3. Illustration of camera intrinsic optimization. (a) In monocular settings, inaccurate intrinsic parameters could be corrected by adjusting the camera pose, *e.g*. shifting the camera origin right by T. (b) This approach is not feasible for multiple cameras under extrinsic constraints like self-driving cars or SLAM devices.

3. Methodology

In the following, we formulate our problem setting in Section 3.1 and detail how we enable intrinsic and extrinsic camera refinement in Section 3.2. We then present our proposed optimization and geometric constraints in Section 3.3 Section 3.4, respectively.

3.1. Multi-camera problem setting

Given a set of coarsely estimated camera poses [1], $\{\mathcal{P}_i\}|_{i=1}^{N} \in \mathbb{SE}(3)$, along with their respective RGB images $\{\mathcal{I}\}|_{i=1}^{N} \in \mathbb{R}^{H\times W\times 3}$, where H and W denote the height and width of the images, and i represents the image/pose index ($1 \leq i \leq N$) among N images. The poses are inaccurate due to two main reasons. Firstly, the orientation and position of the device $\hat{\mathcal{P}}_i$ derived from SLAM can be noisy due to sensor noise and drift in Lidar odometry estimation. Secondly, the RGB images are captured asynchronously to the device pose acquisition. Specifically, the image pose $\mathcal{P}_i$ is roughly estimated by combining the closest device pose $\hat{\mathcal{P}}_i$ and the camera-to-device extrinsic $\mathcal{E}$. This approach overlooks the inevitable time-frame offset (often up to 50 ms), further increasing the discrepancy between the estimated and true camera poses. In the following sections, we detail our approach to jointly correct the noisy set of camera poses and 3D point clouds within 3DGS scene representation.

3.2. Intrinsic and extrinsic refinement with 3DGS

Intrinsic refinement via analytical solution. Existing methods typically assume that camera intrinsics are provided [7, 41] and overlook the importance of refining these parameters. As illustrated in Fig. 3, the inaccuracies of camera intrinsics can be compensated via small extrinsic offsets for single-camera captures [23, 38]. However, this approach fails in multi-camera systems (*e.g*. SLAM or self-driving cars) where poses are constrained by the device $\hat{\mathcal{P}}_i$. In multi-camera setups, inaccurate intrinsic parameters can significantly degrade rendered details, leading to blurry reconstructions. To enable intrinsic refinement, we apply the chain rule of differentiation and obtain analytical solutions for computing the gradient of each intrinsic parameter. We detail the derivation procedures in Supplementary Sec. B and provide qualitative examples of this enhancement in Fig. 7, which improves image quality with clearer text.

Extrinsic refinement via camera decomposition. Refining the camera extrinsics in a multi-camera system is challenging due to the large number of parameters. For instance, a 4-camera rig with 10k images involves 60k degrees of freedom. To address this, we decompose each camera pose into two components: the camera-to-device pose and the device-to-world pose, expressed as:

$$\mathcal{P}^{(j,t)} = \hat{\mathcal{P}}^t \times \mathcal{E}^j, \tag{1}$$

where $\mathcal{P}^{(j,t)}$ is the camera-to-world pose for camera j at time t, $\hat{\mathcal{P}}^t$ is the device-to-world pose at t, and $\mathcal{E}^j$ is the camera-to-device extrinsic for camera j. This approach reduces the problem to modeling 4 shared extrinsics $\mathcal{E}^j$ and 2500 independent device poses $\hat{\mathcal{P}}^t$, totaling $6 \times 2500 + 6 \times 4 = 15024$ degrees of freedom. Shared parameters across cameras and time frames simplify optimization and enhance the stability of joint camera pose refinement and accurate 3D scene reconstruction. This is illustrated in a real SLAM acquisition and its decomposition in Fig. 4.

We can now refine the camera extrinsics by applying small offsets to Eq. 1:

$$\mathcal{P}^{(j,t)} = f(\hat{\mathcal{P}}^t, \vec{\phi}^t) \times g(\mathcal{E}^j, \vec{\rho}^j), \tag{2}$$

where $\vec{\phi}^t$ and $\vec{\rho}^j \in \mathbb{R}^6$ are learnable tensors, each consisting rotation $\vec{\phi}_{\text{rot}}, \vec{\rho}_{\text{rot}} \in \mathbb{R}^3$ and a translation $\vec{\phi}_{\text{trans}}, \vec{\rho}_{\text{trans}} \in \mathbb{R}^3$, to compensate for the device pose at time t and the j^{th} camera-to-device error, respectively. Functions $f(\cdot)$ and $g(\cdot)$ define how these small deltas refine the noisy poses.

There are two general approaches to refine these poses. The first approach is to left-multiply the original pose by the error matrix:

$$f(\hat{\mathcal{P}}^t, \vec{\phi}^t) = \underbrace{\Phi^t}_{\mathbb{SE}(3) \text{ representation of } \phi^t} \times \hat{\mathcal{P}}^t. \tag{3}$$

However, this leads to unstable optimization as it forces the camera location to rotate with respect to the world origin, which is often far from the initial camera value. To address this, we propose right-multiplying the error matrix with the original pose by defining the new device center as ${\mathcal{P}_{\text{d2w}}^{t}}^{*} = R_{\text{d2w}}\Delta t + t_{\text{d2w}}$, and thus:

$$f(\hat{\mathcal{P}}^t, \vec{\phi}^t) = \hat{\mathcal{P}}^t \times \underbrace{\Phi^t}_{\mathbb{SE}(3) \text{ representation of } \phi^t}. \tag{4}$$

We provide qualitative examples for these schemes in Supplementary and adopt the form in Eq. 4 for $f(\cdot)$ and $g(\cdot)$.

3.3. Optimization constraints

Directly optimizing the camera parameters as formulated in Section 3.2 leads to sub-optimal solutions for two main

[1] We refer to the camera pose as the *camera-to-world* pose, indicating the camera's position and orientation in world coordinates for simplicity.

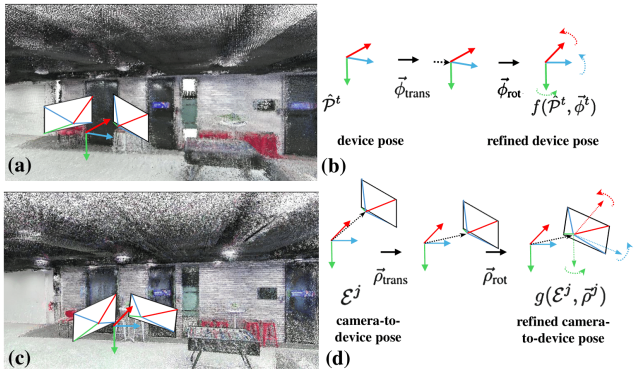
(a)
$\hat{\mathcal{P}}^t$
$\vec{\phi}_{\text{trans}}$
$\vec{\phi}_{\text{rot}}$
$f(\hat{\mathcal{P}}^t, \vec{\phi}^t)$
device pose
refined device pose
(b)
(c)
$\mathcal{E}^j$
$\vec{\rho}_{\text{trans}}$
$\vec{\rho}_{\text{rot}}$
$g(\mathcal{E}^j, \vec{\rho}^j)$
camera-to-device pose
refined camera-to-device pose
(d)

Figure 4. Illustration of our camera decomposition scheme. (a) Initial noisy point cloud from SLAM setup. (b) and (d) Optimization procedures of device-to-world and camera-to-device transformations. (c) Refined point cloud from our constrained optimization approach, showing improved visual quality.

reasons: 1) The inherent ambiguity in perspective projection, where intrinsic parameters and camera poses describe relative and nonlinear relationships, leading to multiple feasible solutions; and 2) The overparameterization of camera poses, where adjusting one camera's orientation affects all device centers, creating unnecessary redundancy for optimization. In this section, we propose a *sensitivity-based pre-conditioning* strategy to adjust the learning rate of each parameter and a *log-barrier* strategy to constrain optimization within the feasible region.

Sensitivity-based pre-conditioning. Inspired by the Levenberg-Marquardt algorithm, which is known to solve general nonlinear optimization problems, such as camera calibration [26], we propose an optimization approach that constrains parameter movements based on their *sensitivity* and initial coarse estimates of poses and intrinsics. This is strongly motivated as even a tiny refinement (1%) in these parameters can lead to significantly different behaviors.

Given a dense point cloud $\mathcal{G}$, we render into UV coordinates by camera-to-world $\mathcal{P}_{\text{c2w}}$ and intrinsic K matrices:

$$(u, v) = Proj(\phi_{\text{rot}}, \phi_{\text{trans}}, \rho_{\text{rot}}, \rho_{\text{trans}} | \mathcal{G}, \mathcal{P}_{\text{c2w}}, K), \qquad (5)$$

where $Proj(\cdot)$ is the projection function. We can then obtain the sensitivity matrix by solving the Jacobian of Eq. 5:

$$\mathcal{J}(\phi_{\text{rot}}, \phi_{\text{trans}}, \rho_{\text{rot}}, \rho_{\text{trans}} | \mathcal{G}, \mathcal{P}_{\text{c2w}}, K) = \begin{pmatrix} {}^{\partial u}/_{\partial \phi_{\text{rot}}} & {}^{\partial u}/_{\partial \phi_{\text{trans}}} & {}^{\partial u}/_{\partial \rho_{\text{rot}}} & {}^{\partial u}/_{\partial \rho_{\text{trans}}} \\ {}^{\partial v}/_{\partial \phi_{\text{rot}}} & {}^{\partial v}/_{\partial \phi_{\text{trans}}} & {}^{\partial v}/_{\partial \rho_{\text{rot}}} & {}^{\partial v}/_{\partial \rho_{\text{trans}}} \end{pmatrix}. \qquad (6)$$

The Jacobian matrix represents how small changes in each input component affect the output and can be efficiently computed. We take the average of individual $\mathcal{J}$ matrices for multi-view camera captures and adjust the learning rate based on the diagonal value ratio of $(\mathcal{J}^\top \mathcal{J})^{-1/2}$, which is the inverse square root of the first-order approximation of the Hessian matrix.

Log-barrier method to constrain the feasible region. In addition to refining each parameter set with its sensitivity-based learning rate, we further construct a log-barrier constraint to ensure crucial parameters remain within their feasible boundaries by empirically assessing the error margin of each parameter.

To achieve this, we define m inequality constraints $h_i(x) < 0$, $(1 \leq i \leq m)$ for parameter x. The log-barrier method expresses these constraints in the negative log form, as $\mathcal{L}_{\text{barrier}} = {}^1/_{\mathcal{T}} \sum_{i=1}^{m} log(-h_i(x))$, where $\mathcal{T}$ is a temperature term that increases from a small value to a very large one. This formulation offers several advantages for training by inspecting the gradient of the negative log form:

$$\frac{\partial {}^1/_\tau log(-h_i(x))}{\partial x} = -\frac{1}{\tau h_i(x)} \frac{\partial h_i(x)}{x}. \qquad (7)$$

As shown in Fig. 5, this creates a symmetric penalty function centered around the initial value. The penalty gradient increases significantly as the parameter approaches the predefined boundaries because the gradient term $-\frac{1}{\tau h_i(x)}$ becomes large. This prevents the parameter from entering infeasible regions. As optimization progresses, we increase the temperature $\mathcal{T}$ to reduce the penalty and allow the parameters to stabilize between the boundaries. This design is ideal for our problem scenario as we can empirically set two bounds and guide the optimization toward a plausible solution. We apply these constraints to both the camera intrinsics and the decomposed camera pose transformations.

3.4. Geometric constraints

In this section, we propose two geometric constraints to improve the robustness in mitigating noisy camera poses. We first use a state-of-the-art keypoint matching method [31] to output semi-dense (up to several hundreds) keypoint matches $\{\vec{x}_i, \vec{x}_{i+n}\}$ for adjacent image frames i and $i + n$. Here, $\vec{x}_i, \vec{x}_{i+n} \in \mathbb{R}^{M \times 2}$ represent M matches for the image pair, and n is a small integer $1 \leq n \leq 3$ to ensure high co-visibility between images. The following two geometric constraints can effectively provide a strong prior for the relative poses between cameras in a multi-camera system.

Soft epipolar constraint. This regularizes the learned relative camera poses to adhere the epipolar geometries. We implement this by first estimating the fundamental matrix $\mathbb{F}$, using the relative camera poses $\mathcal{P}_{i,j}$ and respective intrinsics K_i and K_j, *i.e.* $\mathbb{F}_{ij} = K_i^{-\top} [t]_\times R_{ij} K_j^{-1}$. We can then compute the Sampson distance [36] which takes the matched pixel pairs and $\mathbb{F}$ as inputs:

$$\mathcal{L}_{\text{epipolar}}(\vec{x}_i, \vec{x}_{i+n}, \mathbb{F}) = \sum_{j=0}^{M-1} \frac{\vec{x}^{j}_{i+n}{}^\top \mathbb{F} \vec{x}^j_i}{\left(\mathbb{F}\vec{x}^j_i\right)^2_1 + \left(\mathbb{F}\vec{x}^j_i\right)^2_2 + \left(\mathbb{F}^\top \vec{x}^j_{i+n}\right)^2_1 + \left(\mathbb{F}^\top \vec{x}^j_{i+n}\right)^2_2}.$$

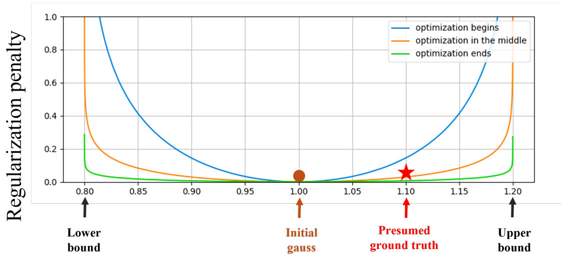

Figure 5. Illustration of the log-barrier method. Lower and upper bounds are predefined based on initial SLAM estimation. At the start of the optimization, the barrier imposes a strong penalty for significant deviations from the initial estimate. As the temperature increases, it transforms into a well-function, allowing the parameter to fully explore the feasible region.

With this constraint as regularizer, we can achieve robust optimization convergence by incorporating prior information about camera intrinsics and extrinsics. However, since the epipolar constraint does not consider depth information and has projective ambiguities, we propose an additional geometric constraint in the following.

Reprojection error regularization. We extend the Bundle Adjustment from traditional S*f*M algorithms into a geometric constraint that simultaneously optimizes both camera poses and 3DGS. This constraint can be expressed as:

$$\begin{aligned}\mathcal{L}_{\text{reproj}}(&\underbrace{\vec{x}_i, \vec{x}_{i+n}}_{\text{matched points}}, \underbrace{\vec{d}_i, \vec{d}_{i+n}}_{\text{depths}} | \underbrace{\mathcal{P}_i, \mathcal{P}_{i+n}}_{\text{camera poses}}, \underbrace{K_i, K_{i+n}}_{\text{intrinsics}}) \\ &= \sum_{j=0}^{M-1} (K_{i+n}\mathcal{P}_{i+n}\mathcal{P}_i^{-1}D_i^j K_i^{-1}\vec{x}_i^j - \vec{x}_{i+n}^j) \\ &\quad + \sum_{j=0}^{M-1} (K_i\mathcal{P}_i\mathcal{P}_{i+n}^{-1}D_{i+n}^j K_{i+n}^{-1}\vec{x}_{i+n}^j - \vec{x}_i), \quad (8)\end{aligned}$$

where $\vec{d}_i$ and $\vec{d}_{i+n} \in \mathbb{R}^{M\times 1}$ are the depths for the matched points in i^{th} and $i+n^{\text{th}}$ images. This regularization term minimizes errors by considering depth distances, thus constraining the geometry of the scene which is complementary to the previous soft epipolar constraint.

Note that many existing works compute alpha-blending along the z-axis component of Gaussians in camera space to approximate *rendered depth*. However, we found this approach unstable during optimization. Therefore, inspired by computer graphics, we instead compute line intersections to determine depths more accurately. We detail the mathematical derivation of this approach in the Supplementary Sec. E.

4. Experiments

Implementation details. We train 3DGS using the following loss objective, which is a weighted combination of our proposed constraints and can be written as:

$$\begin{aligned}\mathcal{L}_{\text{total}} = &\underbrace{\mathcal{L}_{\text{pixel}} + \lambda_{\text{ssim}} \cdot \mathcal{L}_{\text{ssmi}}}_{\text{original learning objective}} + \underbrace{\lambda_{\text{barrier}} \cdot \mathcal{L}_{\text{barrier}}}_{\text{log barrier constraint}} \\ &+ \underbrace{\lambda_{\text{epi}} \cdot \mathcal{L}_{\text{epipolar}} + \lambda_{\text{reproj}} \cdot \mathcal{L}_{\text{reproj}}}_{\text{geometry constraints}} . \quad (9)\end{aligned}$$

We empirically set $\lambda_{\text{ssim}} = 0.2$, $\lambda_{\text{barrier}} = 0.1$, $\lambda_{\text{epi}} = 1 \times 10^{-3}$ and $\lambda_{\text{reproj}} = 5 \times 10^{-4}$ for Eq. 9. The smaller values for λ_{epi} and λ_{reproj} prevent significant deviations in relative poses due to noisy key-point matches. We set the learning rate for intrinsic parameters to 8×10^{-4}. The base extrinsic learning rate is 5×10^{-3}, adjusted for each group of transformation parameters using the diagonal value ratios from $(\mathcal{J}^\top \mathcal{J})^{-1/2}$. For log-barrier constraint on intrinsic parameters, we impose a strict bound of $\pm 2\%$ deviation from the original value. We also apply adaptive constraints empirically for extrinsics: ±0.625°and ±2.5°for ϕ_{rot} and ρ_{rot}, and $\pm 0.125m$ and $\pm 0.5m$ for ϕ_{trans} and ρ_{trans}. For all experiments, we follow [11] and adopt a **test-time adaptation strategy** on the unseen images to refine their camera poses. During test-time adjustments, we apply a learning rate of 5×10^{-4} over 500 iterations while keeping the trained 3DGS parameters frozen. We apply this to the entire test set after training 48k iterations. As most images are captured in uncontrolled settings with varying lighting and exposure [30], we introduce an efficient **exposure compensation module**. We hypothesize that illumination variations are region-specific and affect image brightness gradually. Therefore, we correct this by a *learnable low-frequency* offset. We detail this approach in the Supplementary Sec. C.

Dataset. There is a lack of suitable public datasets of real-world multimodal SLAM sequences that well reflect the challenges faced in industrial applications, where scans are noisy and captured quickly. To address this, we collected data using our self-developed hardware across four scenes, including indoor and challenging outdoor settings. Our hardware, featuring four fisheye cameras, an IMU sensor, and a Lidar, scanned scenes such as a cafeteria, office room, laboratory (100-300m^2), and a residential district in East Asia (85×45m^2). Our captured dataset represents a unique problem setting and can be considered as a special case for autonomous driving. Specifically, as humans carry the capture device and walk around to capture the scene, it induces greater vertical movements than those typically found in autonomous driving datasets. Additionally, these scans feature stronger lighting variations and moving subjects. Due to the absence of advanced hardware synchronization and sophisticated sensor calibration in our rapid data acquisition process, the resulting camera poses and point clouds from SLAM are particularly noisy around object surfaces. We provide details on our devices, acquisition protocol, and data pre-processing in the Supplementary Sec. A, and have released the dataset. We also benchmark on public datasets, though they feature with less sensor noise: Waymo [32] for autonomous driving and GarageWorld [7] for indoor measurement and inspection.

Evaluation metrics. Obtaining ground truth camera poses from real-world settings is challenging so existing works [12, 27] often adopt COLMAP outputs as pseudo ground truth. However, Table 1 shows that COLMAP-generated poses are prone to failures, sometimes catastrophic, making them unreliable as ground truth. This aligns with existing research, where some approaches are more accurate than COLMAP on individual scenes [3], and evaluation rankings vary depending on the reference algorithm used for obtaining pseudo ground truths [2]. As such, we follow established methods [3, 11, 17] and assess pose quality in a self-supervised manner using novel view synthesis [35]. Specifically, we sample test images at N intervals, with N determined per scene to ensure it contains 60 testing images. We report Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) to evaluate rendering quality.

Comparison methods. We compare our constrained optimization approach with various reconstruction methods, both with and without COLMAP, as well as SLAM-based Gaussian Splatting methods. We categorize them as:

- **Direct reconstruction**: This baseline directly optimizes scene reconstruction using the outputs from SLA,M which include noise from various components. Therefore, this is considered the lower bound for our approach.
- **Pose optimization**: This baseline optimizes both the 3DGS parameters and the camera poses. It does not take into account the multi-camera configuration and does not refine camera intrinsic parameters. This comparison method is commonly seen in incremental SLAM papers [16, 19, 23] and can serve as a strong baseline as it aligns with the learning objectives of the *mapping* or *global bundle adjustment* process.
- **3DGS-COLMAP**: The following two methods leverage COLMAP to derive precise camera poses. Despite being time-consuming, COLMAP is widely adopted for training 3DGS, as the resulting poses can often be considered ground truth. We initially included this baseline as the upper bound for performance. In the first variation, **3DGS-COLMAP** uses roughly estimated camera intrinsics to guide the optimization of camera poses. The subsequent variant, **3DGS-COLMAP$^{\triangle}$**, integrates additional approximate camera poses and refines them through a rig-based bundle adjustment (BA). This rig-based BA maintains a learnable, yet shared, constant pose constraint across multiple cameras, making it the most relevant baseline for comparison.
- **Recent progress**: We compare with two SLAM-based 3DGS methods including CF-3DGS [12] and MonoGS [23]. We also compare with InstantSplat [11], which uses a foundation model to provide relative poses and refine reconstruction geometry.
- **Multimodal 3DGS**: We compare with LetsGo [7] and Street-GS [39], which take Lidar data as input for large-scale public benchmarks. We provide implementation details of these methods in the Supplementary Sec. F.
- **SfM-free NeRF**: We compare with CamP+ZipNeRF [1] and BARF [21]. They perform similarly to the baseline, which is a lower bound for our approach.

4.1. Experimental results - Tables 1 and 2

Direct baselines (Table 1 rows 1-2). We show that direct reconstruction using noisy SLAM outputs results in low rendering quality for all indoor/outdoor scenes. In contrast, the pose optimization method improves SSIM over the baseline by 8.3%, 7.89%, 6.97%, and 6.94% for each of the scenes. Both methods underperformed in the Town scene due to its complex geometry and noisy point clouds.

COLMAP-based methods (Table 1 rows 3-5). 3DGS-COLMAP is extensively applied to various 3D reconstruction tasks, yielding satisfactory results for three out of four datasets (SSIM: 0.88, 0.90, and 0.83) despite requiring up to 12 hours of computation time. However, it fails in the Cafeteria scene due to repetitive block patterns (see details in the Supplementary). In contrast, 3DGS-COLMAP$^{\triangle}$ has a reduced pose estimation time of 2-3 hours due to SLAM pose prior and Rig-BA. While it produces a more balanced rendering quality, it underperforms in the last two scenes compared to 3DGS-COLMAP, suggesting that rig optimization may lead to suboptimal outcomes. GLOMAP [29] is more efficient but generally underperforms the two baselines.

Recent progress (Table 1 rows 6-8). We show that both 3DGS for incremental SLAM methods, MonoGS and CF-3DGS, perform weakly across all evaluated datasets, with SSIM ranging from 0.40 to 0.75. This deficiency stems from their reliance on high-quality image sequences, where accurate relative pose estimation depends heavily on image covisibility. Specifically, our dataset imposes a stringent 85% covisibility threshold, which makes it more challenging to obtain relative camera poses across the global scene. Additionally, the dataset contains various recurring block patterns as well as plain surfaces, which can lead to degenerate solutions. Conversely, InstantSplat achieves better rendering quality by leveraging foundation models.

Multimodal 3DGS (Table 2). Our approach achieves the best score in 12 cases and the second-best in the remaining ones. Notably, Street-GS also includes pose optimization, similar to our 3DGS-COLMAP baseline. However, our method shows significant improvement due to the combination of camera decomposition, intrinsic optimization, and various constraints, all without relying on COLMAP. We present additional quantitative analysis and qualitative comparisons in the Supplementary Sec. G and H.

Table 1. Quantitative comparisons on our dataset. Red and blue highlights indicate the 1st and 2nd-best results, respectively, for each metric. $^\triangle$ performs additional rig-based bundle adjustment to refine initial camera estimations. Our proposed method matches or surpasses the performance of the widely-adopt 3DGS-COLMAP approach while requiring significantly less data pre-processing time (prep. time).

Methods	Prep. time	Cafeteria			Office			Laboratory			Town		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Direct reconst.	3 minutes	19.23	0.7887	0.2238	17.49	0.7577	0.2777	18.35	0.7975	0.2207	16.12	0.6151	0.3234
Pose optimize.	5 minutes	26.89	0.8716	0.1219	23.96	0.8366	0.1663	26.11	0.8673	0.1183	20.18	0.6845	0.2392
3DGS-COLMAP	4-12 hours	17.03	0.7681	0.2475	25.82	0.8832	0.1262	28.30	0.9080	0.0837	24.07	0.8304	0.1362
3DGS-COLMAP$^\triangle$	2-3 hours	26.51	0.8379	0.1281	23.91	0.8394	0.1797	23.76	0.8157	0.1277	23.51	0.8090	0.1534
3DGS-GLOMAP	2-6 hours	21.83	0.7889	0.1546	21.94	0.8609	0.1464	25.92	0.8805	0.1098	23.37	0.8254	0.1630
CF-3DGS [12]	1 minute	15.44	0.5412	0.5849	16.53	0.7555	0.4086	16.44	0.7557	0.3945	15.45	0.5412	0.5849
MonoGS [23]	1 minute	8.27	0.4684	0.6033	9.56	0.4957	0.6560	13.08	0.6011	0.5103	12.74	0.3085	0.5331
InstantSplat [11]	50 minutes	19.86	0.7743	0.2548	23.30	0.8718	0.1451	20.89	0.8624	0.1801	21.48	0.7378	0.2999
CamP+ZipNeRF	-	22.05	0.8544	0.3718	19.32	0.8253	0.2049	17.67	0.7527	0.2833	16.35	0.6797	0.5326
BARF	-	18.97	0.7340	0.2622	17.03	0.7001	0.3717	19.29	0.7529	0.2701	16.97	0.5249	0.5108
Ours	5 minutes	29.05	0.9168	0.0817	26.07	0.8850	0.1131	28.64	0.9104	0.0845	24.52	0.8259	0.1428

Table 2. Quantitative comparisons on GarageWorld (*left*) and Waymo (*right*) datasets with state-of-the-art multimodal methods.

Methods	GarageWorld [7]									Waymo [32]					
	Group 0			Group 3			Group 6			Scene 002			Scene 031		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
3DGS [17]	25.43	0.8215	0.2721	23.61	0.8162	0.2698	21.23	0.7002	0.4640	25.84	0.8700	0.1746	24.42	0.8328	0.1783
LetsGo [7]	25.29	0.8387	0.2978	25.31	0.8329	0.2804	21.72	0.7462	0.445	26.11	0.8429	0.2951	24.79	0.7851	0.3477
Street-GS [39]	24.20	0.8222	0.2993	24.19	0.8209	0.2849	20.52	0.7206	0.4763	27.96	0.8708	0.1664	25.04	0.8553	0.1697
Ours	26.06	0.8325	0.2605	25.07	0.8311	0.2523	23.76	0.7779	0.3537	29.75	0.883	0.161	28.48	0.868	0.1450

Table 3. Ablations on number of cameras. We show that the improvement consistently increases with number of cameras.

Methods	1 camera			2 cameras			4 cameras		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Cafeteria									
Pose optim.	27.51	0.881	0.079	27.52	0.885	0.093	26.43	0.859	0.119
Ours	29.81	0.917	0.067	29.76	0.921	0.072	29.50	0.922	0.077
Improv.	2.30	0.036	0.012	2.24	0.036	0.021	3.07	0.063	0.042
Office									
Pose optim.	24.36	0.845	0.121	24.00	0.832	0.141	23.38	0.827	0.169
Ours	26.51	0.885	0.103	26.20	0.881	0.110	26.12	0.891	0.109
Improv.	2.15	0.040	0.018	2.20	0.049	0.031	2.74	0.064	0.060

4.2. Ablations

Camera decompositon & pre-conditioning. Directly optimizing camera parameters in a multi-camera setup can be computationally inefficient without improving reconstruction quality. To address this, we propose a camera decomposition and sensitivity-based pre-conditioning optimization strategies. As shown in Table 4, this approach achieves optimal performance with fast training convergence.

Number of cameras. We evaluate the camera decomposition in Table 3 and show that our proposed method consistently improve the rendering quality. Our method is effective even in single-camera scenarios, as it links all camera poses with a shared camera-to-device matrix. This shared matrix provides a partial global constraint on the camera-to-device pose, simplifying the optimization process especially within limited training budgets.

Intrinsic optimization. Table 5 shows that intrinsic refinement improve rendering quality, with consistent gains across all metrics. In addition, we demonstrate that intrinsic refinement can deblur images by adjusting focal lengths and the principal point, as shown in Fig. 7.

Table 4. Ablations on camera decomposition and sensitivity-based pre-conditioning strategies. **C.P.** and **P.C.** denote camera decomposition and pre-conditioning, respectively. In addition to standard rendering metrics, we report convergence percentage (**CVG%**), indicating the training stage at which **SSIM** exceeds 95% of its peak. A smaller values refers more stable optimization.

Methods		Cafeteria				Laboratory			
C. D.	P. C.	PSNR ↑	SSIM ↑	LPIPS ↓	CVG%	PSNR ↑	SSIM ↑	LPIPS ↓	CVG%
✗	✗	26.91	0.8659	0.1129	34.38	27.00	0.8807	0.1045	31.25
✗	✓	26.45	0.8577	0.1072	22.92	26.07	0.8645	0.1096	18.76
✓	✗	28.87	0.9154	0.0850	43.10	28.52	0.9092	0.0894	39.58
✓	✓	29.05	0.9168	0.0817	15.65	28.64	0.9104	0.0845	16.67

Table 5. Ablations on intrinsic refinement.

Methods	Cafeteria			Laboratory		
Refinement	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
✗	27.40	0.8975	0.0976	26.79	0.8843	0.0932
✓	29.05	0.9168	0.0817	28.64	0.9104	0.0845

Log-barrier method. Using only the pre-conditioning optimization strategy is insufficient to prevent sensitive parameters from exceeding their feasible region. To address this, we use a log-barrier method to constrain the feasible region. We show that by simply constraining the feasible region within ±2% improves SSIM by 6.8% in Fig. 8.

Geometric constraints. We next assess the importance of the two proposed geometric constraints. In addition to standard metrics, we report the mean epipolar line error (Ep-e) and the reprojection error (RP-e) in Table 6. We observe consistent performance gains with both geometric

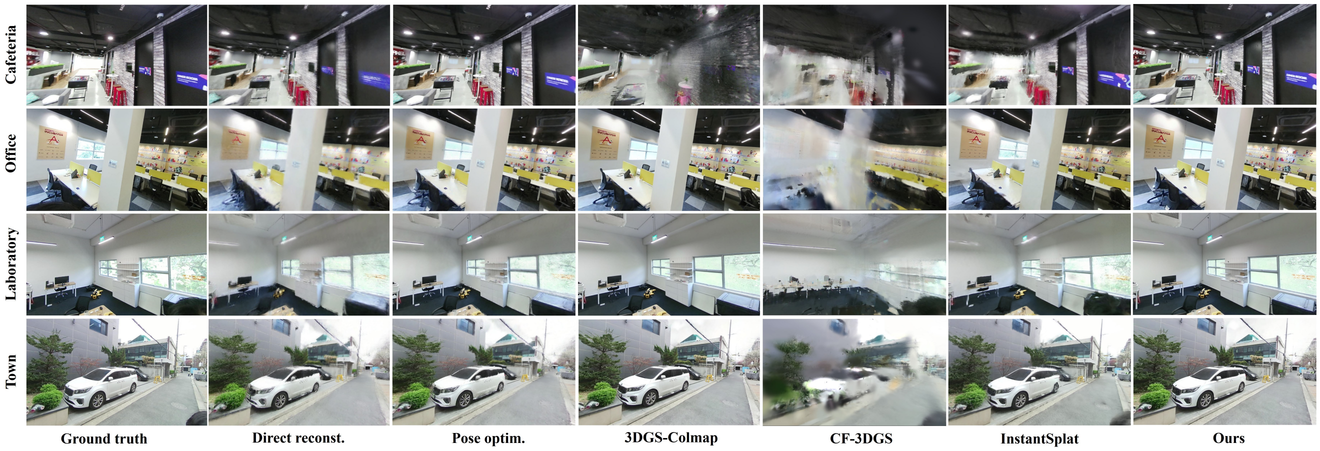

Figure 6. Qualitative comparisons with existing approaches. Our method achieves high rendering quality across a diverse range of scenes.

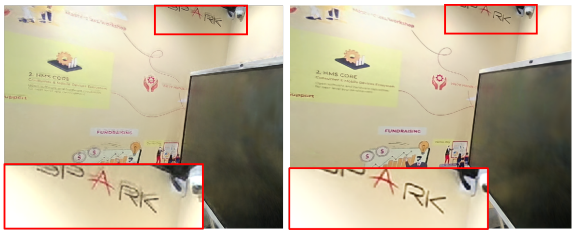

Figure 7. Qualitative examples for novel view synthesis with (*right*) and without (*left*) intrinsic refinement. We eliminate blurriness and enhance rendering quality by refining camera intrinsics during optimization.

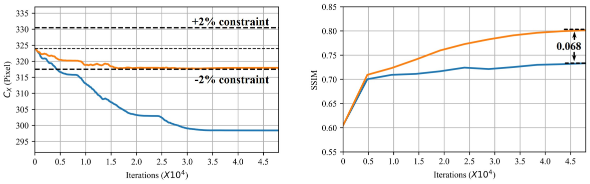

Figure 8. Ablations on log-barrier method. We show that training without log-barrier (blue plot) lead to significant principle point deviation (*left*) and sub-optimal solution (*right*). In contrast, using log-barrier method (orange plot) results in a higher SSIM (*right*).

constraints, even as random noise increases in both camera-to-device and device poses. We also provide qualitative examples of key-point matches and their corresponding epipolar lines in Fig. 9. We show that minor epipole displacements resulting from geometric constraints significantly reduce the epipolar line error from 2.70 to 0.75 pixels.

5. Conclusion

This paper presented a method for 3DGS with noisy camera and point cloud initializations from a multi-camera SLAM system. We proposed a constrained optimization framework that decomposes the camera pose into camera-to-device and device-to-world transformations. By optimizing these transformations individually under soft constraints, we can efficiently and accurately construct 3DGS. We also introduced a new multi-view 3D dataset captured under these noisy, albeit practical, settings, which we will release to the community to encourage further research development.

Table 6. Ablation study on geometric constraint. Ep-e stands for mean epipolar line error (Ep-e) and RP-e denotes mean reprojection error. Our proposed losses help to reduce both errors and increase the rendering quality.

Noise Level	Methods		Cafeteria				
	E.P.	R.P.	PSNR ↑	SSIM ↑	LPIPS ↓	Ep-e ↓	RP-e ↓
-	✗	✗	27.05	0.8945	0.1047	1.14	2.52
	✗	✓	27.24	0.9130	0.0906	1.11	2.04
	✓	✗	27.25	0.9141	0.0895	1.09	2.05
	✓	✓	27.31	0.9147	0.0891	1.08	1.88
0.2°	✗	✗	26.04	0.8901	0.1007	1.23	2.56
	✗	✓	26.16	0.8952	0.0989	1.17	2.19
	✓	✗	26.51	0.9007	0.0963	1.12	2.06
	✓	✓	26.84	0.9045	0.0958	1.11	2.00
0.5°	✗	✗	24.80	0.8584	0.1244	1.72	3.92
	✗	✓	24.87	0.8607	0.1196	1.42	2.99
	✓	✗	25.18	0.8665	0.1138	1.23	2.35
	✓	✓	25.20	0.8672	0.1120	1.21	2.32

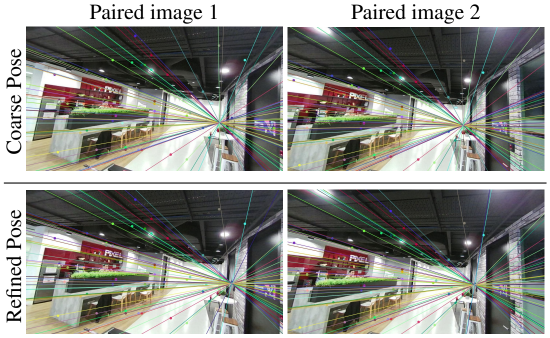

Figure 9. Qualitative examples on key-point matches and their corresponding epipolar lines. Vertical inspection shows that the geometric constraints cause minor epipole displacements towards lower epipolar error as well as better reconstruction quality.

References

[1] Jonathan T. Barron, Keunhong Park, Ben Mildenhall, John Flynn, Dor Verbin, Pratul Srinivasan, Peter Hedman, Philipp Henzler, and Ricardo Martin-Brualla. CamP Zip-NeRF: A Code Release for CamP and Zip-NeRF, 2024. 6

[2] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *ICCV*, 2021. 6

[3] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. In *ECCV*, 2024. 6

[4] Jiahao Chen, Yipeng Qin, Lingjie Liu, Jiangbo Lu, and Guanbin Li. Nerf-hugs: Improved neural radiance fields in non-static scenes using heuristics-guided segmentation. In *CVPR*, 2024. 2

[5] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. In *ICCV*, 2019. 2

[6] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In *ECCV*, 2022. 2

[7] Jiadi Cui, Junming Cao, Yuhui Zhong, Liao Wang, Fuqiang Zhao, Penghao Wang, Yifan Chen, Zhipeng He, Lan Xu, Yujiao Shi, et al. Letsgo: Large-scale garage modeling and rendering via lidar-assisted gaussian primitives. *arXiv preprint arXiv:2404.09748*, 2024. 1, 2, 3, 6, 7

[8] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *TPAMI*, 2007. 1

[9] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *CVPR*, 2022. 2

[10] Tianchen Deng, Yaohui Chen, Leyan Zhang, Jianfei Yang, Shenghai Yuan, Danwei Wang, and Weidong Chen. Compact 3d gaussian splatting for dense visual slam. *arXiv:2403.11247*, 2024. 1, 2

[11] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, Zhangyang Wang, and Yue Wang. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds, 2024. 5, 6, 7

[12] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. In *CVPR*, 2024. 6, 7

[13] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2010. 1

[14] Changjian Jiang, Ruilan Gao, Kele Shao, Yue Wang, Rong Xiong, and Yu Zhang. Li-gs: Gaussian splatting with lidar incorporated for accurate large-scale reconstruction. *arXiv preprint arXiv:2409.12899*, 2024. 1

[15] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM ToG*, 2013. 2

[16] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *CVPR*, 2024. 1, 2, 6

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM ToG*, 2023. 1, 6, 7

[18] Mustafa Khan, Hamidreza Fazlali, Dhruv Sharma, Tongtong Cao, Dongfeng Bai, Yuan Ren, and Bingbing Liu. Autosplat: Constrained gaussian splatting for autonomous driving scene reconstruction. *arXiv preprint arXiv:2407.02598*, 2024. 2

[19] Tian Lan, Qinwei Lin, and Haoqian Wang. Monocular gaussian slam with language extended loop closure. *arXiv preprint arXiv:2405.13748*, 2024. 6

[20] Hansol Lim, Hanbeom Chang, Jongseong Brad Choi, and Chul Min Yeum. Lidar-3dgs: Lidar reinforced 3d gaussian splatting for multimodal radiance field rendering. *arXiv preprint arXiv:2409.16296*, 2024. 2

[21] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021. 2, 6

[22] Changkun Liu, Shuai Chen, Yash Bhalgat, Siyan Hu, Zirui Wang, Ming Cheng, Victor Adrian Prisacariu, and Tristan Braud. GS-CPR: Efficient camera pose refinement via 3d gaussian splatting. In *ICLR*, 2025. 2

[23] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *CVPR*, 2024. 1, 2, 3, 6, 7

[24] Qingwei Mi and Tianhan Gao. 3d reconstruction based on the depth image: A review. In *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. Springer, 2022. 2

[25] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[26] Pradit Mittrapiyanuruk. A memo on how to use the levenberg-marquardt algorithm for refining camera calibration parameters. *Robot Vision Laboratory, Purdue University, West Lafayette, IN, USA*, 2006. 4

[27] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 2022. 6

[28] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 2015. 1

[29] Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes Lutz Schönberger. Global Structure-from-Motion Revisited. In *ECCV*, 2024. 6

[30] Christian Reiser, Richard Szeliski, Dor Verbin, Pratul P. Srinivasan, Ben Mildenhall, Andreas Geiger, Jonathan T. Barron, and Peter Hedman. Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. *SIGGRAPH*, 2023. 5

[31] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 4

[32] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou,

Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 5, 7

[33] Shuo Sun, Malcolm Mielle, Achim J Lilienthal, and Martin Magnusson. High-fidelity slam using gaussian splatting with rendering-guided densification and regularized optimization. *arXiv:2403.12535*, 2024. 1, 2

[34] Yuan Sun, Xuan Wang, Yunfan Zhang, Jie Zhang, Caigui Jiang, Yu Guo, and Fei Wang. icomma: Inverting 3d gaussians splatting for camera pose estimation via comparing and matching. *arXiv:2312.09031*, 2023. 2

[35] Michael Waechter, Mate Beljan, Simon Fuhrmann, Nils Moehrle, Johannes Kopf, and Michael Goesele. Virtual rephotography: Novel view prediction error for 3d reconstruction. *ACM TOG*, 2017. 6

[36] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *ICCV*, 2023. 4

[37] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2

[38] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *CVPR*, 2024. 1, 2, 3

[39] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street Gaussians: Modeling Dynamic Urban Scenes with Gaussian Splatting. In *ECCV*, 2024. 2, 6, 7

[40] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *IROS*, 2021. 2

[41] Cheng Zhao, Su Sun, Ruoyu Wang, Yuliang Guo, Jun-Jun Wan, Zhou Huang, Xinyu Huang, Yingjie Victor Chen, and Liu Ren. Tclc-gs: Tightly coupled lidar-camera gaussian splatting for surrounding autonomous driving scenes. *arXiv preprint arXiv:2404.02410*, 2024. 2, 3

[42] Chunran Zheng, Wei Xu, Zuhao Zou, Tong Hua, Chongjian Yuan, Dongjiao He, Bingyang Zhou, Zheng Liu, Jiarong Lin, Fangcheng Zhu, et al. Fast-livo2: Fast, direct lidar-inertial-visual odometry. *arXiv preprint arXiv:2408.14035*, 2024. 2

[43] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *CVPR*, 2024. 2