IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

FusionGS-SLAM: Multiple Sensors Fusion for Localization and Real-Time Photorealistic Mapping

Thanh-Danh Phan and Gon-Woo Kim, *Member, IEEE*

***Abstract*—This work presents a FusionGS-SLAM, a robust framework for simultaneous localization and real-time photorealistic mapping leveraging the power of sensor fusion techniques. To achieve this, the proposed method employs a tightly-coupled technique to effectively combine multiple factors from improved subsystems, thereby generating a robust odometry for the downstream tasks. Moreover, a dense 3D Gaussian map is constructed by leveraging geometric information across sensor modalities, with real-time mapping strategies designed to enhance robustness and rendering quality in large-scale and challenging environments. Experimental evaluation of various challenging scenes, including the public and self-collected datasets, showcases the superior performance compared to the current state-of-the-art 3DGS SLAM.**

***Index Terms*—Mapping, real-time 3DGS mapping, multiple sensor fusion, SLAM.**

I. Introduction

SLAM has been a cornerstone for numerous applications in robotics, providing precise localization and environmental mapping for autonomous navigation, augmented reality, and other perception-dependent tasks. Traditional SLAM systems [1], [2], [3] have focused primarily on geometric accuracy for robust navigation, often representing environments as sparse point clouds or occupancy grids.

Recent methods such as Neural Radiance Fields (NeRF) [4] and 3D Gaussian Splatting (3DGS) [5] have improved photorealistic scene representation, enabling detailed and photorealistic environmental maps. However, existing approaches face substantial limitations in real-world deployment. Camera-based systems [6], [7], [8] show promise but remain largely confined to controlled indoor environments. These methods struggle with fundamental outdoor challenges: scene scales, complex lighting conditions, and dynamic robotic platform motion. The absence of reliable depth information in purely visual systems creates scale ambiguity and mapping instabilities in unbounded environments.

Recent sensor fusion approaches for photorealistic mapping [9], [10], [11], [12] have made progress by integrating multimodal information. However, existing approaches [9], [12], [13] either function similarly to offline methods by incorporating extended optimization processes or achieve only sub-10 FPS performance in real-time operation, limiting their applicability for time-critical robotic applications.

Traditional multi-sensor SLAM approaches have focused on geometric accuracy but lack realistic representation, while recent neural rendering methods prioritize visual quality but struggle with robust localization under varying conditions. This fundamental gap highlights the need for a unified approach that maintains both robust tracking and high-quality mapping in challenging real-world scenarios. Motivated by these standpoints, we propose FusionGS-SLAM, leveraging visual-LiDAR-inertial fusion for robust localization and real-time photorealistic mapping capabilities. Our key contributions include:

- Proposing a visual-LiDAR-inertial framework in outdoor environments, demonstrating robust localization while simultaneously delivering real-time, realistic mapping capabilities.
- Developing effective photorealistic map construction strategies by using the strengths of our sensor fusion system, allowing the pipeline to work well in outdoor and challenging environments.
- Extensive experiments were conducted to prove our proposed frameworks' robustness and effectiveness. We perform our systems on both public datasets and our self-collected dataset to show the applicability of the system and the superiority compared to the existing state-of-the-art (SOTA) 3DGS SLAM pipelines.

The remainder is organized as follows: Section II presents the related works in the field. Meanwhile, the whole framework, including localization and photorealistic mapping techniques, is illustrated in Section III. Lastly, the experimental results and conclusion are manifested in Section IV and Section V respectively.

II. Related Works

A. LiDAR-Inertial-Visual Fusion in SLAM

Multi-sensor fusion has emerged as a key strategy for improving localization reliability in challenging environments. LVI-SAM [2] introduced a tightly coupled framework combining

Received 6 February 2025; accepted 8 June 2025. Date of publication 2 July 2025; date of current version 18 July 2025. This article was recommended for publication by Associate Editor K. Skinner and Editor J. Civera upon evaluation of the reviewers' comments. This work was supported in part by the National Research Foundation of Korea (NRF) funded by Korea Government (MSIT) under Grant RS-2025-00561031, 50 and in part by the Innovative Human Resource Development for Local Intellectualization Program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) funded by Korea Government (MSIT) under Grant IITP-2025-RS-2020-II201462, 50. *(Corresponding author: Gon-Woo Kim.)*

The authors are with the Intelligent Robotics Laboratory, Department of Intelligent Systems and Robotics, Chungbuk National University, Cheongju 28644, South Korea (e-mail: gwkim@cbnu.ac.kr).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3585388, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3585388

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

a visual-inertial system and LiDAR-inertial system, demonstrating improved accuracy over single-modality approaches. R3LIVE [14] efficiently integrates visual, LiDAR, and inertial data with novel photometric methods for real-time colored mapping. Meanwhile, Switch-SLAM [3] introduced a degeneration detection module and adaptively switches the systems depending on the environmental structure. FAST-LIVO2 [15] distinguishes itself through seamless integration of direct LiDAR, inertial, and visual measurements within a unified voxel framework, eliminating the need for computationally intensive feature extraction. These frameworks underscore the value of adaptive sensor fusion, yet primarily prioritize geometric accuracy over visual quality, leaving the integration of robust fusion with photorealistic rendering largely unexplored for real-time applications in challenging environments.

B. Neural Rendering Approaches for Photorealistic SLAM

Traditional SLAM systems have prioritized geometric accuracy over visual fidelity. Recent rendering techniques have transformed this landscape, with 3DGS [5] representing scenes as optimizable 3D Gaussian primitives, enabling significantly faster rendering while maintaining high visual quality. This advantage has spurred SLAM implementations like GSICPSLAM [8] and PhotoSLAM [6], which demonstrated promising results primarily in controlled indoor settings. The majority of high-quality mapping solutions in online SLAM frameworks rely solely on visual data. The methods [7], [16] optimize camera poses by minimizing photometric differences between synthesized and observed views. However, these approaches face challenges in tracking speed and stability during rapid motion.

Several works have explored integrating 3DGS with sensor fusion for more robust operation in unbounded environments. The method like [11] achieves impressive photorealistic reconstructions by combining LiDAR-visual-inertial inputs. While these methods produce visually compelling results, they currently operate in offline settings, which limits their utility for scenarios requiring on-the-fly processing. Recent efforts toward online operation [9], [13], [17] have shown promising advances in outdoor environment mapping. Nonetheless, these methods tend to extend their optimization process to guarantee the mapping fidelity, which makes them akin to offline methods. The letter [10] achieves photorealistic mapping through multisensor fusion; however, its methodology—optimizing 100 keyframes simultaneously—necessitates waiting for frames and their Gaussian primitives formation, making it suboptimal for real-time robotics applications. Despite these notable advancements, the fundamental challenge of real-time, realistic mapping remains largely unresolved, presenting significant opportunities for further research.

III. Localization and 3D Gaussian-Based Map

This section focuses on simultaneously estimating the pose of the system along with the geometry of the environment in the formation of a 3D geometric map and a 3DGS map. The sensor system comprises a 3D LiDAR, camera, and IMU, with the body frame assumed to coincide with the IMU frame. As illustrated in Fig. 1, our framework first presents a localization module that fuses multiple sensor data through factor graph optimization, followed by our approach to photorealistic mapping, which optimizes the hyper primitives of 3D Gaussians with time-critical mapping strategies for robust performance in outdoor environments.

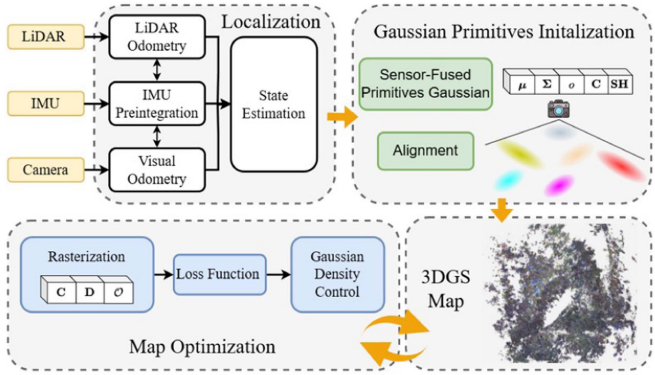


Fig. 1. System overview of FusionGS-SLAM's pipeline.

A. Localization Module

The localization module integrates data from three primary sources: visual odometry (VO), LiDAR odometry (LO), and IMU preintegration factor, combined through factor graph optimization. The system state is defined as:

$$\boldsymbol{\chi} = \begin{bmatrix} \mathbf{R}^T & \boldsymbol{p}^T & \boldsymbol{v}^T & \boldsymbol{b}^T \end{bmatrix}^T \tag{1}$$

where $\mathbf{R} \in SO(3)$ is the system rotation matrix, $\boldsymbol{p} \in \mathbb{R}^3$ is the system position, $\boldsymbol{v} \in \mathbb{R}^3$ is the system linear velocity, and $\boldsymbol{b} \in \mathbb{R}^3$ is the IMU bias.

The VO factor leverages the ALIKE model [18] to extract sub-pixel accurate keypoints that ensure robust tracking performance. Our system implements an adaptive confidence thresholding mechanism that dynamically adjusts feature selection criteria based on environmental conditions. Initially prioritizing high-confidence features (threshold: 0.8) for maximum reliability, the algorithm automatically reduces this threshold in textureless environments until sufficient features are secured. This balanced approach maintains tracking stability while ensuring adequate feature correspondence for effective Gaussian map initialization. Concurrently, the LO factor utilizes scan-to-map matching following [19]. Meanwhile, the IMU preintegration is augmented using [1], which enhances the robustness of the system in challenging environments. Upon incorporating the IMU preintegration factor, the IMU component is employed to refine the LO and VO.

These three factors are jointly optimized in the factor graph, with robust failure detection mechanisms that allow seamless switching between sensor modalities:

$$\delta\rho = \begin{cases} -(\mathbf{J}_v^\top \mathbf{J}_v)^{-1}\mathbf{J}_v^\top f_v(\rho_v), & \text{if LO fail} \\ -(\mathbf{J}_l^\top \mathbf{J}_l + \beta(\mathbf{J}_l^\top \mathbf{J}_l))^{-1}\mathbf{J}_l^\top f_l(\rho_l), & \text{else if VO fail} \\ \underset{\delta\rho}{\operatorname{argmin}} \left(\|\mathbf{e}_v(\delta\rho)\|^2 + \|\mathbf{e}_l(\delta\rho)\|^2 \right), & \text{otherwise.} \end{cases} \tag{2}$$

where $\mathbf{J}_v$ and $\mathbf{J}_l$ are the Jacobian matrices for visual and LiDAR measurements respectively; ρ_v and ρ_l are the poses of visual and LiDAR factor; $f_v(\rho_v)$ and $f_l(\rho_l)$ represent the residual functions for visual and LiDAR factors; β is the Levenberg-Marquardt damping factor; $\mathbf{e}_v$ and $\mathbf{e}_l$ are the error terms defined as $\mathbf{e}_v(\delta\rho) = \delta\rho + (\mathbf{J}_v^\top \mathbf{J}_v)^{-1}\mathbf{J}_v^\top f_v(\rho_v)$ and $\mathbf{e}_l(\delta\rho) = (\mathbf{J}_l^\top \mathbf{J}_l + \beta(\mathbf{J}_l^\top \mathbf{J}_l))^{-1}\mathbf{J}_l^\top f_l(\rho_l)$ respectively.

The framework implements failure detection mechanisms for both VO and LO to ensure system resilience in challenging environments. For VO, failures are detected when tracked feature counts drop suddenly or when IMU bias experiences significant spikes. Upon detection, VO is re-initialized while the system temporarily relies on LiDAR updates. The learned feature extraction approach reduces the frequency of these failures, improving overall system stability. For LO, we employ the method from [20] to detect LiDAR performance degradation, allowing the system to rely on VO during these instances adaptively. This failure handling strategy ensures continuous operation even when individual sensor modalities temporarily fail.

B. Gaussian Map Representation

The scene's underlying geometry is represented by 3D Gaussians, denoted as G_i :

$$G_i(\boldsymbol{\zeta}) = \exp\left(-\frac{1}{2}(\boldsymbol{\zeta}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{\zeta}-\boldsymbol{\mu}_i)\right) \tag{3}$$

where ζ is the variable of the Gaussian function. Each Gaussian G_i is parameterized by a set of parameters $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \boldsymbol{o}_i, \boldsymbol{c}_i, \boldsymbol{SH}\}$, where $\boldsymbol{\mu}_i \in \mathbb{R}^3$ is the mean vector; $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$ is the covariance matrix; $\boldsymbol{o}_i \in \mathbb{R}$ is the opacity scalar; $\boldsymbol{c}_i \in \mathbb{R}^3$ is the RGB color vector, $\boldsymbol{SH}$ is the spherical harmonics coefficient. Applying the principal component analysis, the 3×3 covariance matrix can be decomposed into $\mathbf{R}_G\mathbf{S}(\mathbf{R}_G\mathbf{S})^\top$, where $\mathbf{R}_G \in SO(3)$ is a rotation matrix of a Gaussian, $\mathbf{S}$ is a diagonal scaling matrix.

C. Gaussian Primitives Initialization

1) Gaussian Primitives Initialization Via LiDAR-Visual Fusion: Accurate initialization of Gaussian primitives' spatial $\boldsymbol{\mu}$ and chromatic $\boldsymbol{c}$ attributes is essential for efficient 3DGS convergence speed [6]. To achieve this precision, a robust methodology that strategically leverages the complementary characteristics of multi-modal sensor data is introduced. Fisrt, the accumulated LiDAR point cloud $\{(Y_j, c_j^l)\}_{j=1}^M$ is transformed to the camera coordinate frame via calibrated extrinsics T_{CL}, yielding $\Lambda \leftarrow \{(T_{CL}Y_j, c_j^l)\}_{j=1}^M$. These points are subsequently discretized into an angular range image to mitigate occlusions by preserving the nearest point within each angular bin. By projecting RGB values from camera images onto these points, we create a geometrically accurate point cloud enriched with chromatic information. Then, we achieve efficient initialization by fusing LiDAR-derived geometry with visual features through a depth-invariant spherical representation. The spherical projection maps both LiDAR points and visual feature rays onto a unit sphere centered at the camera's optical center, creating $\Lambda_{sphere} \leftarrow \{(\frac{p}{\|p\|}, c^l)|(p, c^l) \in \Lambda\}$ and $y_{sphere} \leftarrow \frac{\bar{y}}{\|\bar{y}\|}$ respectively. This projection converts 3D Cartesian coordinates to unit vectors by dividing by their Euclidean norm. This mathematical operation preserves directional information while discarding distance, creating a common representation where points in the same direction map to the same location regardless of depth. A KD-tree spatial index enables rapid identification of the three closest LiDAR neighbors for each visual feature on the sphere. To ensure only reliable geometry is incorporated, depth estimates undergo further validation: results are discarded if local depth variation exceeds a set threshold or if the intersection parameter is unstable.

Algorithm 1: Multi-Modal Hyper Primitives Initial Aggregation

Require: 2D visual features $F = \{(y_i, c_i^v)\}_{i=1}^N$, LiDAR points with colors $\{(Y_j, c_j^l)\}_{j=1}^M$ stacked from multiple frames, Camera intrinsics K, Lidar-Camera Calibration T_{CL}
Ensure: Combined colored point cloud P_{fuse}
$P_{fuse} \leftarrow \emptyset$
$\Lambda \leftarrow \{(T_{CL}Y_j, c_j^l)\}_{j=1}^M$ ▷ Transform LiDAR points to camera frame
$\Lambda_{sphere} \leftarrow \{(\frac{p}{\|p\|}, c^l)|(p, c^l) \in \Lambda\}$ ▷ Project to unit sphere
$\Gamma \leftarrow \text{BuildKDTree}(\Lambda_{sphere})$
$P_{fuse} \leftarrow P_{fuse} \cup \{(p, c^l, S^l(p)), \forall (p, c^l) \in \Lambda_{sphere}\}$ ▷ Add LiDAR points with scales
for each feature $(y, c^v) \in F$ **do**
  $\bar{y} \leftarrow K^{-1}[y^\top, 1]^\top$ ▷ Unprojection
  $y_{sphere} \leftarrow \frac{\bar{y}}{\|\bar{y}\|}$ ▷ Project to unit sphere
  $(p_1, p_2, p_3) \leftarrow \Gamma.\text{kNearestNeighbors}(y_{sphere}, k = 3)$
  $(z_1, z_2, z_3) \leftarrow (\|\mathcal{C}(p_1)\|, \|\mathcal{C}(p_2)\|, \|\mathcal{C}(p_3)\|)$ ▷ $\mathcal{C}$ maps to Cartesian space
  $\boldsymbol{n} \leftarrow (p_1 z_1 - p_2 z_2) \times (p_2 z_2 - p_3 z_3)$ ▷ Plane normal
  $s \leftarrow \frac{\boldsymbol{n}\cdot p_1 z_1}{\boldsymbol{n}\cdot y_{sphere}}$ ▷ Intersection parameter
  if $|\max(z_1, z_2, z_3) - \min(z_1, z_2, z_3)| \le d_0$ and $s > s_0$ **then**
    $S^v \leftarrow \text{computeLocalScale}((p_1, p_2, p_3), (z_1, z_2, z_3))$
    $P_{fuse} \leftarrow P_{fuse} \cup \{(y_{sphere} \cdot s, c^v, S^v)\}$ ▷ Add point with color and scale
return P_{fuse}

2) Gaussian Primitives Correction: Hyper primitives need to be accurately initialized for effective scene representation. However, high-speed motion and rough terrain can significantly impact their performance in estimating 3D structures. These conditions can cause motion blur and inconsistent feature tracking, making initialization unstable in challenging environments. To enhance the stability of these primitives, this letter introduces a correction method using an optimization algorithm that minimizes residual errors in 3D feature mismatches presented in Fig. 2. The approach relies on a well-estimated system pose to align the mean of hyper primitives, ensuring accurate projection in the 2D-rendered image. Initially, constraints are established between the newly updated mean of hyper primitives $\boldsymbol{\mu}_t$ and its

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

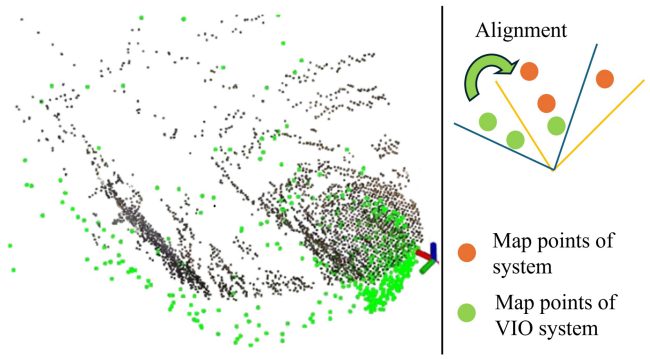


Fig. 2. Demonstration of system and VIO map points alignment.

corresponding global geometric map M:

$$r_F = \sum \left\| M - \left(\mathbf{R}_t \boldsymbol{\mu}_{c,t} + \boldsymbol{p}_t \right) \right\|^2 \tag{4}$$

where $\mathbf{R}_t$ and $\boldsymbol{p}_t$ are the rotation and position of the system at timestamp t, which is defined at equation (1).

Conversely, the inertial data from the IMU provides prior information on the trajectory, including rotational velocity and translational acceleration. The constraints for the IMU measurements are expressed as follows:

$$r_\alpha = \sum \left\| \hat{\boldsymbol{a}}_t - \mathbf{R}_{c,t}^T \left(\frac{d^2}{dt^2} \boldsymbol{p}_{c,t} \right) \right\|^2 \tag{5}$$

$$r_\omega = \sum \left\| \hat{\boldsymbol{\omega}}_t - \hat{\boldsymbol{\omega}}_{c,t} \right\|^2 \tag{6}$$

where $\hat{\boldsymbol{a}}_t$ and $\hat{\boldsymbol{\omega}}_t$ are acceleration and angular velocity from IMU preintegration, respectively; $\mathbf{R}_{c,t}^T$ and $\boldsymbol{p}_{c,t}$ are the rotation matrix and translation matrix of VO at timestamp t, respectively; and $\hat{\boldsymbol{\omega}}_{c,t}$ is the rotational velocity interpolated from the output pose of the VO. Ultimately, the set of corrected hyper primitives is defined as:

$$\hat{\aleph} = \underset{\aleph}{\operatorname{argmin}} \left(r_F + r_\alpha + r_\omega \right) \tag{7}$$

By solving the optimization equation above we can obtain the robustly aligned hyper primitives with a more precise mean.

3) Hyper Primitives Formation: In this step, the entire hyper primitive is assembled from four elements, as shown in Section III-B. The mean $\boldsymbol{\mu}$ and color $\boldsymbol{c}$ are determined according to Sections III-C1 and III-C2. For the scale computation S, we employ a strategy as described in Algorithm 1. Particularly, for LiDAR points, the scale matrix S^l is calculated from the local spatial distribution of neighboring points. For visual features, the scale S^v is derived using LiDAR depth priors from points sharing similar viewing directions rather than physical proximity. By transferring scale information along similar view rays, we maintain geometric consistency while enabling effective initialization even in LiDAR-sparse regions. This method eliminates the need for additional KD-tree construction for visual features, instead efficiently utilizing the existing LiDAR spatial index for efficient computation. Meanwhile, the rotation component of Gaussians is initialized as an identity matrix, and opacity parameters are set via inverse sigmoid initialization at 0.3.

D. Keyframing

Optimizing every frame is computationally prohibitive, necessitating strategic keyframe selection to balance information gain and efficiency. In this work, we adopt a keyframe-based approach [21] for both visual odometry and 3DGS mapping. Keyframe selection relies on three complementary criteria: (1) when mean feature displacement between the current and latest keyframe exceeds threshold θ_d, providing sufficient parallax for geometric estimation; (2) when tracked feature count diminishes below threshold θ_n; and (3) when a substantial proportion of features demonstrate high confidence scores above θ_q, indicating good feature quality. The satisfaction of any single criterion from this triad is sufficient to trigger keyframe designation.

E. Map Optimization

Following the keyframing process, each received new keyframe contains information about the unobserved scene. We implement a sliding window approach for learning local viewpoints to avoid local minima and maintain high mapping quality in real-time operations. This strategy involves the immediate optimization of newly received keyframes, along with the random selection of local neighboring frames. To address the global map forgetting problem, a randomly selected distant historical keyframe is concatenated for one-time windowing.

The mapping process focuses on refining the Gaussian features visible within the currently visible set of keyframes. Firstly, the opacity mask is rasterized, similar to [16] to identify regions requiring additional optimization from the current viewpoint:

$$\mathcal{O}(u,v) = \sum_{i=1}^{N} \alpha_i \prod_{j=0}^{i-1} (1-\alpha_j), \mathcal{D}(u,v) = \sum_{i=1}^{N} d_i \alpha_i \prod_{j=0}^{i-1} (1-\alpha_j) \tag{8}$$

where N is the total number of Gaussians affecting the pixel, α_i is the alpha value of the i-th Gaussian at (u,v), d_i is the depth of the i-th Gaussian. The 3D Gaussians visible in this frame are then optimized according to a comprehensive loss function combining photometric consistency, structural geometry, and geometric constraints $\mathcal{L}_{\text{rgb}} + \lambda_d \mathcal{L}_{\text{geo}} + \lambda_\delta \mathcal{L}_{\text{delta}}$, where the RGB loss component combines L1 and SSIM losses:

$$\mathcal{L}_{\text{rgb}} = (1-\lambda_s) \| I_r - I_{gt} \|_1 + \lambda_s (1 - \text{SSIM}(I_r, I_{gt})) \tag{9}$$

For geometry supervision, we compute normalized depth loss when ground truth is available:

$$\mathcal{L}_{\text{geo}} = \frac{1}{|\mathcal{M}|} \sum_{(u,v)\in\mathcal{M}} |\mathcal{D}(u,v) - \mathcal{D}_{gt}(u,v)| \tag{10}$$

where $\mathcal{M}$ represents the valid depth mask. Additionally, we enforce temporal consistency through delta depth loss between consecutive frames:

$$\mathcal{L}_{\text{delta}} = \| \mu_{t+1} - \mathcal{D}_{t+1}(\pi(\mu_{t+1})) \|_2 \tag{11}$$

where $\mu_{t+1} = T_{t+1,t}\pi^{-1}(u, v, D_t)$ denotes the transformed 3D point from frame t to $t+1$, and $\pi(\mu_{t+1})$ projects this 3D point onto the 2D image coordinates (u, v) using the camera intrinsics. This loss function jointly optimizes for visual appearance, geometric accuracy, and temporal coherence.

F. Gaussian Density Control

In this part, we implement adaptive densification to optimize primitive density across spatial segments. Our densification algorithm enhances scene fidelity using a temporal modulation factor $\varepsilon(\iota) = 1 + \exp(-\kappa(\iota - \iota_0))$, where ι_0 initiates adaptive control and κ represents the decay rate. This factor calibrates both the uv gradient threshold $\Theta_{uv}(\iota) = \Phi(||\nabla_{uv}||, 1 - (1 - \beta_{uv}) \cdot \varepsilon(\iota))$ and scale threshold $\Theta_{\Xi}(\iota) = \Phi(\Psi_{max}, 1 - (1 - \beta_{\Psi}) \cdot \varepsilon(\iota))$, with $\Phi(\cdot, \cdot)$ as the quantile function and β_{uv}, β_{Ψ} as baseline percentiles. Gaussians are cloned when $||\nabla_{uv}|| > \Theta_{uv}(\iota)$ and split when $\Psi_{max} > \Theta_{\Psi}(\iota)$. This dynamic thresholding transitions from aggressive early refinement to conservative adjustments later in optimization. We periodically remove Gaussians with negligible opacity or excessive spatial extent to maintain computational efficiency.

IV. Experimental Results

A. Evaluation Setup

1) Hardware and Parameter Settings: All experiments are conducted on a computer equipped with an Intel Core i9-10980XE CPU and an NVIDIA RTX 3090 GPU. The implementation combines CUDA, C/C++, and Python within a unified framework running on the Robot Operating System. For the SLAM optimization process, we leverage the GTSAM library to solve the pose graph optimization problem. Our sensor setup consists of a 16-channel Velodyne 3D LiDAR, a ZED 2 stereo camera, and a MicroStrain 3DM-GX5-25 IMU. We perform calibrations using open-source tools: LiDAR-IMU calibration with [22] and camera-IMU calibration with [23]. Through this LiDAR-IMU-camera calibration chain, we establish the extrinsic transformation matrix between the LiDAR and camera sensor. For parameter settings, we set θ_d, θ_n, and θ_q to 12, 30, and 0.8, respectively, while the keyframing windowing size for map expansion is set to 5. The map optimization loss weights λ_d and λ_θ are both set to 0.1. With a base learning rate of 0.01, the learning rates for Gaussian attributes μ_i, Σ_i, o_i, and c_i are multiplied by factors of 0.1, 0.25, 0.3, 2, and 0.5, respectively. For Gaussian density control every, we establish baseline percentiles of $\beta_{uv} = 0.96$ and $\beta_{\Psi} = 0.99$.

2) Experimental Datasets: The performance of the proposed framework is evaluated across diverse outdoor environments using both established public datasets and our own collected data. For an effective comparison with state-of-the-art methods, we follow the benchmarking approach referenced in [10] and [13]. Our evaluation includes sequences from the Botanic Garden dataset (1018-00 and 1018-13) [24], the KITTI dataset [25], and the MCD dataset (sequences tuhh-02 and tuhh-04) [26]. Additionally, we validate our method's practical applicability using data collected from our custom-designed sensor system in two challenging outdoor settings: a university campus playground and an unstructured agricultural crop field [1].

TABLE I
Quantitative Rendering Performance on Different Datasets With Image Resolution Being (480 × 270) and (480 × 300) for Self-Collected and Botanic Garden Sequences, Respectively

Method	Metric	Playground	Field	101800	101813	Avg
PhotoSLAM	PSNR↑	12.39	11.53	11.05	13.26	12.06
	SSIM↑	0.441	0.346	0.338	0.342	0.370
	LPIPS↓	0.722	0.761	0.703	0.704	0.720
GSICPSLAM	PSNR↑	6.920	8.910	6.550	11.86	8.560
	SSIM↑	0.118	0.270	0.024	0.351	0.190
	LPIPS↓	0.865	1.050	0.716	0.949	0.895
MonoGaussian	PSNR↑	16.18	11.80	12.72	15.13	14.21
	SSIM↑	0.479	0.269	0.346	0.343	0.451
	LPIPS↓	0.650	0.851	0.679	0.632	0.608
Our	PSNR↑	**21.08**	**18.46**	**19.24**	**17.41**	**19.05**
	SSIM↑	**0.610**	**0.594**	**0.621**	**0.534**	**0.590**
	LPIPS↓	**0.330**	**0.460**	**0.413**	**0.463**	**0.420**

3) Baselines and Metrics: To rigorously evaluate our method's efficacy, we conducted comprehensive experiments comparing its performance against both proprietary and leading open-source approaches. For proprietary methods [10], [13], we adhered to their established benchmarks to ensure fair comparison. Our evaluation against open-source alternatives specifically targeted state-of-the-art real-time 3DGS SLAM systems with outdoor environment capabilities, as most existing implementations focus primarily on indoor settings. The comparative analysis included PhotoSLAM [6], GSICPSLAM [8], and MonoGS [7]. Additionally, juxtaposing our odometry results with the sensor fusion pipeline proposed in [2], [14], [15] is further demonstrated.

We evaluate our Gaussian splatting map reconstruction with the standard image quality metrics: Peak Signal to Noise Ratio (PSNR), SSIM, and Learned Perceptual Image Patch Similarity (LPIPS). Additionally, the accuracy of the tracking trajectory is compared with the ground truth using the root mean square error (RMSE), rotational error (RE), and translational error (TE).

B. Comparisons

1) Rendering Results: The evaluation spans several distinct environmental scenarios, each presenting a different aspect for demonstrating our systems' performance. We compare against existing real-time 3DGS SLAM methods using their public code, with results shown in Fig. 3 and Table I. Although [7] produces higher-quality rendered images through extensive optimization iterations, it still exhibits map artifacts. Our method achieves superior performance across all metrics due to robust sensor-fused localization and optimized map construction strategies. Fig. 4 shows the constructed Gaussian map with optimized geometry and primitives. Our 3DGS map maintains well-structured properties while providing denser, more informative primitives that enhance the rendering process.

We further extended our comparative analysis to include state-of-the-art sensor fusion approaches [10], [13]. As demonstrated in Table II, our method achieves performance comparable to

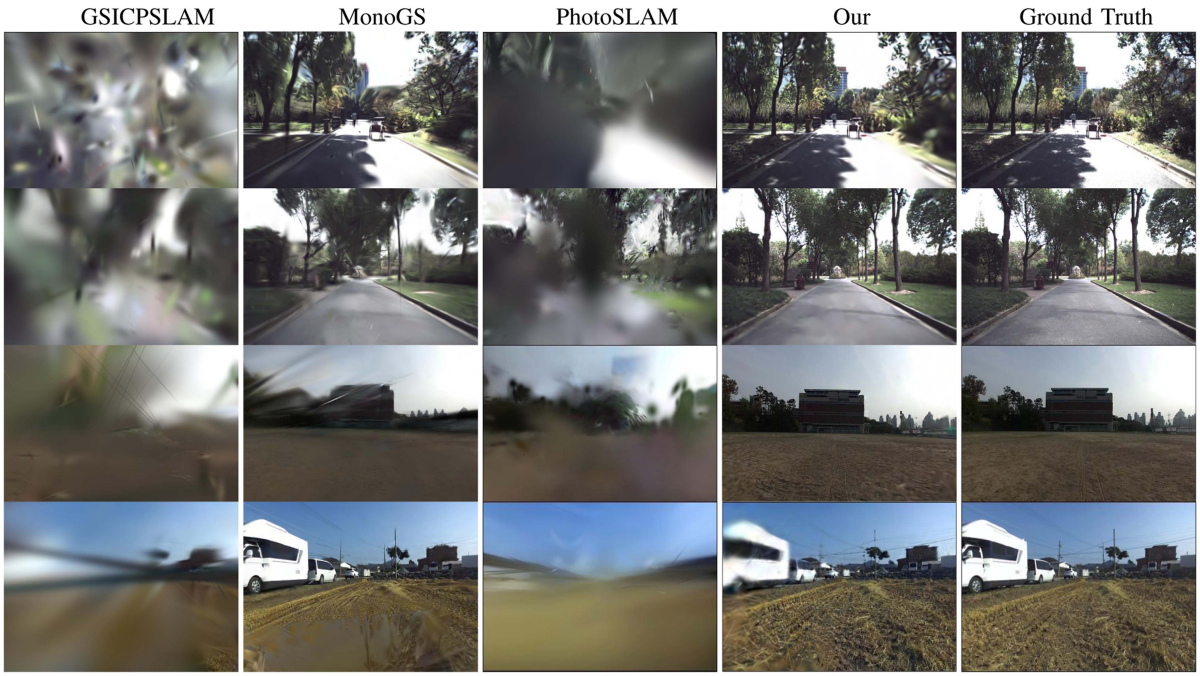


Fig. 3. Viewpoint demonstration of rendering performance throughout three sequences. The rows present rendered images across distinct sequences: Botanic Garden 1018-00 and 1018-13, Playground, and Field.

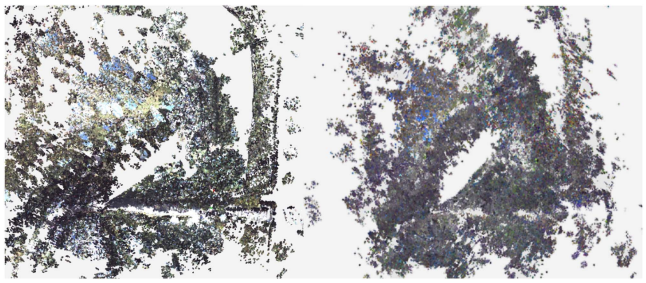
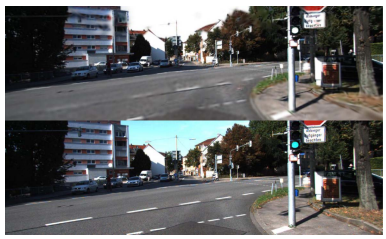


Fig. 4. Visualization comparison between initial geometric point cloud (left) and optimized 3D Gaussian map (right).

TABLE II
COMPARISON WITH AVAILABLE RESULTS FROM [10] ON GARDEN 1018-00 SEQUENCE

	101800		tuhh-02		tuhh-04	
Method	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
NERF-SLAM	15.17	0.352	18.70	0.632	15.40	0.428
SplaTAM	16.18	0.487	13.68	0.488	10.01	0.327
Gaussian-LIC	18.65	0.596	**20.19**	**0.639**	**19.57**	0.531
Our	**19.24**	**0.621**	19.97	0.612	19.39	**0.607**

Gaussian-LIC despite operating on sparse spinning LiDAR data rather than dense depth information. For a comprehensive evaluation, we also assessed the mapping efficiency of competing methods benchmarked in [10] using Gaussian-LIC's odometry instead of their original estimated trajectories. While several competing approaches produce similar qualitative results, they typically require significantly longer optimization times, which compromises true real-time performance. Even after eliminating the constraints in odometry and expanding processing time in their pipelines, our baseline still achieves superior performance in rendering quality. Our comparison with HGS-Mapping [13] on the KITTI dataset, presented in Fig. 5, reveals methodological distinctions. Despite HGS-Mapping's claims of online capability, its requirements—specifically 100 iterations per keyframe—are inconsistent with true online processing and instead reflect resources typically reserved for offline approaches. Additionally, HGS-Mapping inherits pre-estimated poses from other state-of-the-art methods and re-optimizes from these initial conditions, which might reduce computational burden by reallocating resources from odometry to mapping modules. In contrast, our approach maintains comparable visual fidelity while strictly adhering to real-time constraints throughout the entire operation.



	KITTI		
	PSNR	SSIM	FPS
[6]	13.26	0.478	**278**
[16]	14.78	0.52	0.38
[13]	**22.52**	**0.800**	0.41
Our	19.12	0.651	16.72

Fig. 5. Rendered image (top-left), groundtruth image (bottom-left), and performance comparison of methods on KITTI dataset (right) at 1242 × 375 resolution.

2) Tracking Results: Our framework's localization performance was benchmarked against three SOTA methods: LVI-SAM [2], R3LIVE [14], and FASTLIVO2 [15]. Because the R3LIVE source is not well-designed for sparse-spinning LiDAR, both FAST-LIVO2 and LVISAM are evaluated using spinning LiDAR in the Playground sequence. The evaluation spanned four distinct environmental scenarios, as illustrated in

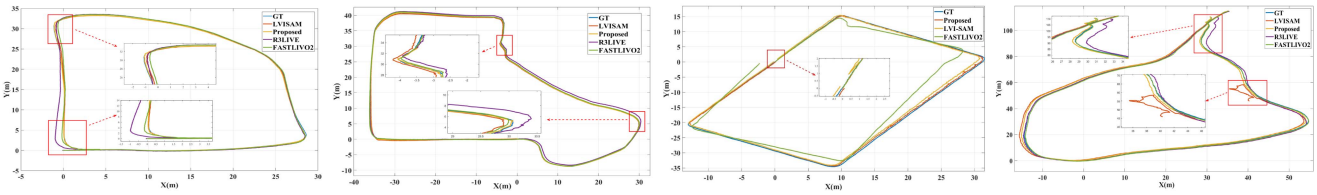


Fig. 6. Comparative odometry in four cases: Botanic Garden 1018-00, 1018-13, Playground, and Tuhh04 sequences in left to right order.

TABLE III
COMPARATIVE LOCALIZATION PERFORMANCE

Method	Metric	1018-00	1018-13	PG	Tuhh04
LVISAM	RMSE (m)↓	0.214	0.103	0.471	Fail
	TE (m)↓	0.259	0.204	1.404	Fail
	RE (deg)↓	1.45	11.96	2.530	Fail
R3LIVE	RMSE (m)↓	1.124	0.358	-	0.589
	TE (m)↓	1.286	0.252	-	2.022
	RE (deg)↓	2.74	13.71	-	12.27
FASTLIVO2	RMSE (m)↓	0.484	0.149	2.489	**0.229**
	TE (m)↓	0.122	0.385	2.366	**0.062**
	RE (deg)↓	1.57	12.89	10.60	**2.15**
Our	RMSE (m)↓	**0.143**	**0.049**	**0.181**	0.469
	TE (m)↓	**0.083**	**0.153**	**0.038**	0.337
	RE (deg)↓	**0.62**	**9.64**	**2.490**	1.31
	Distance (m)	115.2	166.9	218.0	297

TABLE IV
COMPARISON OF RUNTIME ON BOTANIC 1018-00 GARDEN SEQUENCE

	Tracking (s)↓	Mapping (s)↓	Completion (s) ↓
PhotoSLAM	0.0544	0.00273	98
GSICPSLAM	0.0334	0.0289	98
MonoGS	1.4115	0.7061	2714.325
Gaussian-LIC	0.1	-	98
Our (480×270)	0.0668	0.0405	98
Our (960×540)	0.0668	0.0544	98

Fig. 6. Quantitative results presented in Table III demonstrate that our proposed pipeline achieves performance competitive with FASTLIVO2 while outperforming both R3LIVE and LVI-SAM across the tested sequences. In the playground sequence, FASTLIVO2 performs well during the initial phase but begins to drift noticeably after the first sharp turn.

C. Runtime Analysis

In this part, we recorded both component-wise execution time and overall system completion time. The runtime comparison presented in Table IV reveals that our approach achieves comparable execution speeds to SOTA methods like PhotoSLAM and GSICPSLAM, while maintaining substantially faster performance than MonoGS. Fig. 7 provides deeper insights into the computational dynamics of our system through a temporal breakdown of seven key components: densification, backward propagation, forward propagation, rasterization, number of Gaussians, GPU storage, and overall iteration time. The performance profile exhibits two distinct phases: an initial convergence period followed by sustained steady-state operation. During initialization, the system requires higher computational resources to establish the foundational map structure through predefined iterations. Despite this increased initial load, the total processing time remains acceptable for online operation at 120 ms. Once stabilized, execution times maintain consistent performance across all Gaussian submap optimization components.

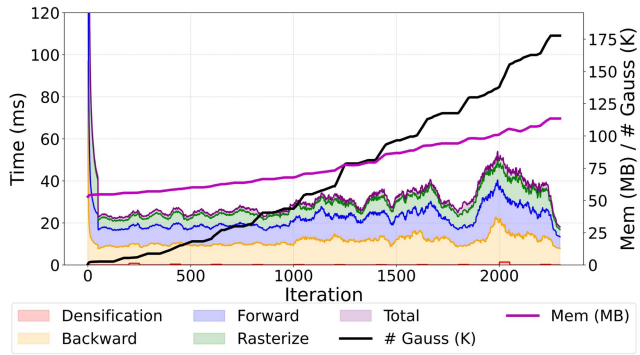


Fig. 7. Time analysis breakdown of computational components over iterations.

TABLE V
QUANTITATIVE ILLUSTRATION OF ABLATION STUDY ON BOTANIC 1018-00 SEQUENCE. THIS SHOWS THE IMPACT OF EACH MODULE

Method	PSNR ↑	SSIM ↑	Time(s) ↓
G-Primitive Source			
G_{vis} (Visual)	16.81	0.490	0.0341
G_{vis} w/o $\hat{\aleph}$	10.57	0.800	0.0286
G_L (LiDAR)	18.54	0.580	0.0361
Σ-Initialization			
Σ_{prev} [11]	18.32	0.540	0.0412
Density Optimization			
Method [5]	18.79	0.594	0.0387
w/o $\mathcal{L}_{geo} + \mathcal{L}_{delta}$	18.91	0.580	0.0334

D. Ablation Study

We conduct an ablation study to evaluate the effectiveness of different components in our framework presented in Table V. Initially, we assess the impact of various primitive source configurations. The analysis of hyper primitives initialization reveals how LiDAR points G_L and learned features G_{vis} achieve comprehensive scene coverage through their complementary properties. Fusing these modalities enables our framework to maintain dense and complete scene representation, as demonstrated in Fig. 8. These effects are also illustrated on the 'Primitive Source' of Table V. This integrated approach ensures robust system performance across diverse environments, even under sensor degradation cases. Second, evaluation of scale initialization

Fig. 8. Visualization comparison of multi-modal feature representation. Four views demonstrate a 3DGS map from: LiDAR source (top-left), learned visual features (top-right), fused cooperative representation (bottom-left), and ground truth reference (bottom-right).

reveals the effects of using the initialization methods from [11], which yield lower rendering quality compared to our strategy (19.24 in the 1018-00 sequence) as shown in Table I. This proves that the initialization time and structural information are highly important in the optimization process. Evaluation of post-densification against the original 3DGS method [5] yields a marginal trade-off between reconstruction quality and computational cost. Finally, experiments without geometry supervision validate the importance of maintaining geometric constraints for robust mapping performance.

V. Conclusion

We present FusionGS-SLAM, a multi-sensor SLAM system that effectively addresses the dual challenges of robust localization and real-time high-fidelity mapping. Our framework incorporates sensor fusion methodologies specifically designed to construct well-structured, high-quality maps across diverse environments. Extensive experimental validation demonstrates competitive performance compared to SOTA approaches. While the system performs well in most scenarios, challenging conditions with rugged surfaces or sharp turns require additional keyframes, reducing optimization iterations on keyframes. Future work should explore parallel and batch computing to allow more optimization iterations.

References

[1] Q. H. Hoang and G. -W. Kim, "IMU augment tightly coupled Lidar-visual-inertial odometry for agricultural environments," *IEEE Robot. Automat. Lett.*, vol. 9, no. 10, pp. 8483–8490, Oct. 2024.

[2] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: Tightly-coupled LiDAR-visual-inertial odometry via smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 5692–5698.

[3] J. Lee et al., "Switch-SLAM: Switching-based LiDAR-inertial-visual SLAM for degenerate environments," *IEEE Robot. Automat. Lett.*, vol. 9, no. 8, pp. 7270–7277, Aug. 2024.

[4] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 405–421.

[5] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph.*, vol. 42, no. 4, Jul. 2023, Art. no. 139. [Online]. Available: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[6] H. Huang, L. Li, C. Hui, and S. -K. Yeung, "Photo-SLAM: Real-time simultaneous localization and photorealistic mapping for monocular, stereo, and RGB-D cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 21584–21593.

[7] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, "Gaussian splatting SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 18039–18048.

[8] S. Ha, J. Yeon, and H. Yu, "RGBD GS-ICP SLAM," in *Proc. Eur. Conf. Comput. Vis.*, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds., Cham, Germany, 2025, pp. 180–197.

[9] R. Xiao, W. Liu, Y. Chen, and L. Hu, "LIV-GS: LiDAR-vision integration for 3D Gaussian splatting SLAM in outdoor environments," *IEEE Robot. Autom. Lett.*, 2024.

[10] X. Lang et al., "Gaussian-LIC: Real-time photo-realistic SLAM with Gaussian splatting and LiDAR-inertial-camera fusion," 2024, *arXiv:2404.06926*.

[11] S. Hong, J. He, X. Zheng, and C. Zheng, "LIV-GaussMap: LiDAR-inertial-visual fusion for real-time 3D radiance field map rendering," *IEEE Robot. Automat. Lett.*, vol. 9, no. 11, pp. 9765–9772, Nov. 2024.

[12] C. Wu, Y. Duan, X. Zhang, Y. Sheng, J. Ji, and Y. Zhang, "MM-Gaussian: 3D Gaussian-based multi-modal fusion for localization and reconstruction in unbounded scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*. 2024, pp. 12287–12293.

[13] K. Wu et al., "HGS-mapping: Online dense mapping using hybrid Gaussian representation in urban scenes," *IEEE Robot. Automat. Lett.*, vol. 9, no. 11, pp. 9573–9580, Nov. 2024.

[14] J. Lin and F. Zhang, "R^3 LIVE: A robust, real-time, RGB-colored, LiDAR-inertial-visual tightly-coupled state estimation and mapping package," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 10672–10678.

[15] C. Zheng et al., "FAST-LIVO2: Fast, direct LiDAR-inertial-visual odometry," *IEEE Trans. Robot.*, vol. 41, pp. 326–346, 2025.

[16] N. Keetha et al., "SplaTAM: Splat, track & map 3D Gaussians for dense RGB-D SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 21357–21366.

[17] C. Wu, Y. Duan, X. Zhang, Y. Sheng, J. Ji, and Y. Zhang, "MM-Gaussian: 3D Gaussian-based multi-modal fusion for localization and reconstruction in unbounded scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 12287–12293. [Online]. Available: https://api.semanticscholar.org/CorpusID:268987792

[18] X. Zhao, X. Wu, J. Miao, W. Chen, P. C. Y. Chen, and Z. Li, "ALIKE: Accurate and lightweight keypoint detection and descriptor extraction," *IEEE Trans. Multimedia*, vol. 25, pp. 3101–3112, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID244908616

[19] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot. Sci. Syst.*, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID18612391

[20] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 809–816.

[21] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[22] S. Mishra, G. Pandey, and S. Saripalli, "Target-free extrinsic calibration of a 3D-LiDAR and an IMU," *Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst.*, 2021, pp. 1–7. [Online]. Available: https://api.semanticscholar.org/CorpusID238215859

[23] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 4304–4311.

[24] Y. Liu et al., "BotanicGarden: A high-quality dataset for robot navigation in unstructured natural environments," *IEEE Robot. Automat. Lett.*, vol. 9, no. 3, pp. 2798–2805, Mar. 2024.

[25] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[26] T. -M. Nguyen et al., "MCD: Diverse large-scale multi-campus dataset for robot perception," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2024, pp. 22304–22313. [Online]. Available: https://mcdviral.github.io/