2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

MCGS-SLAM: A Multi-Camera SLAM Framework Using Gaussian Splatting for High-Fidelity Mapping

Zhihao Cao[1], Hanyu Wu[2], Li Wa Tang[2], Zizhou Luo[3],
Wei Zhang[4], Marc Pollefeys[5,6], Zihan Zhu[5,*], and Martin R. Oswald[7]

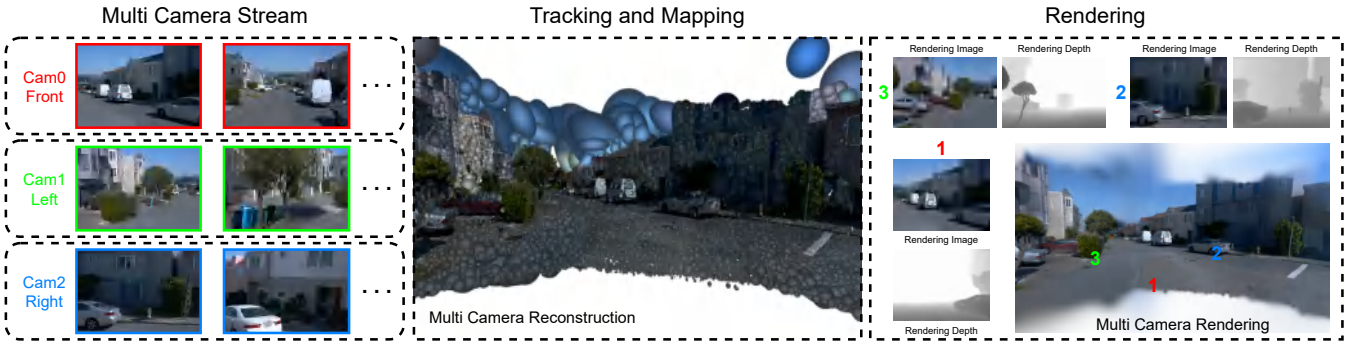

Fig. 1: MCGS-SLAM synchronizes RGB inputs from the front, left, and right cameras of the multi-camera rig in the Waymo dataset and fuses them into a unified 3D Gaussian Splatting map. The system performs real-time tracking and mapping, enabling high-fidelity reconstruction of both color and depth views from each individual camera. Through joint multi-camera optimization, MCGS-SLAM ensures accurate pose and geometry alignment, while supporting comprehensive multi-view rendering for photorealistic visualization. `https://zhihao-ethz.github.io/mcgs-slam/`

***Abstract*— Recent progress in dense SLAM has primarily targeted monocular setups, often at the expense of robustness and geometric coverage. We present MCGS-SLAM, the first purely RGB-based multi-camera SLAM system built on 3D Gaussian Splatting (3DGS). Unlike prior methods relying on sparse maps or inertial data, MCGS-SLAM fuses dense RGB inputs from multiple viewpoints into a unified, continuously optimized Gaussian map. A multi-camera bundle adjustment (MCBA) jointly refines poses and depths via dense photometric and geometric residuals, while a scale consistency module enforces metric alignment across views using low-rank priors. The system supports RGB input and maintains real-time performance at large scale. Experiments on synthetic and real-world datasets show that MCGS-SLAM consistently yields accurate trajectories and photorealistic reconstructions, usually outperforming monocular baselines. Notably, the wide field of view from multi-camera input enables reconstruction of side-view regions that monocular setups miss, critical for safe autonomous operation. These results highlight the promise of multi-camera Gaussian Splatting SLAM for high-fidelity mapping in robotics and autonomous driving.**

*Zihan Zhu is the Project Lead of this work.
[1]Zhihao Cao is with the Department of Mathematics, ETH Zurich, Switzerland. (e-mail: zhicao@student.ethz.ch)
[2]Hanyu Wu and Li Wa Tang are with the Department of Mechanical and Process Engineering, ETH Zurich, Switzerland. (e-mail: hanywu@student.ethz.ch; litang1@student.ethz.ch)
[3]Zizhou Luo is with the Department of Informatics, University of Zurich, Switzerland. (e-mail: zizhou.luo@uzh.ch)
[4]Wei Zhang is with the Institute for Photogrammetry, University of Stuttgart, Germany (e-mail: wei.zhang@ifp.uni-stuttgart.de)
[5]Marc Pollefeys and Zihan Zhu are with Computer Vision and Geometry Group, ETH Zurich, 8092 Zurich, Switzerland. (e-mail: zihan.zhu@inf.ethz.ch; marc.pollefeys@inf.ethz.ch)
[6]Marc Pollefeys is also with Microsoft Spatial AI Lab, 8038 Zurich, Switzerland (e-mail: mapoll@microsoft.com)
[7]Martin R. Oswald is with Computer Vision Research Group, University of Amsterdam, Netherlands (e-mail: m.r.oswald@uva.nl)

I. Introduction

Simultaneous Localization and Mapping (SLAM) remains a foundational component in robotic navigation and 3D scene reconstruction. Early monocular SLAM systems, such as ORB-SLAM3 [1], LSD-SLAM [2], and DSO [3], achieve real-time camera tracking by minimizing sparse geometric or photometric residuals. However, their reliance on a single narrow field-of-view (FoV) camera renders them susceptible to scale drift, motion blur, and occlusions. Learning-augmented approaches, including DROID-SLAM [4] and MAC-VO [5], alleviate some of these issues, yet the core limitation of monocular viewpoint remains a fundamental bottleneck for scene completeness and depth accuracy. A natural solution is to employ overlapping multi-camera systems. Early visual-inertial odometry pipelines improved robustness through fisheye clusters [6]. More recently, Kuo et al. [7] proposed a generalization of visual-inertial bundle adjustment (BA) to wide-baseline multi-camera systems. BAMF-SLAM [8] introduced a scalable BA formulation for general camera networks, achieving state-of-the-art odometry accuracy. Nevertheless, these systems typically yield only sparse landmarks, and some systems heavily rely on inertial sensors, relegating high-fidelity geometry and photorealistic rendering to costly offline post-processing.

In parallel, dense scene representations have made remarkable strides, though predominantly in monocular settings, thus underutilizing the potential of multi-camera

platforms. Traditional map structures such as surfels and TSDF volumes [9], [10] have evolved towards neural implicit fields. NeRF-based methods [11], [12] enable impressive photorealism, while SLAM variants like NICER-SLAM [13] and GLORIE-SLAM [14] integrate neural fields into SLAM pipelines for high-quality novel view synthesis. However, these methods remain computationally expensive and lack explicit geometric control. In contrast, 3D Gaussian Splatting (3DGS) [15] offers an efficient alternative that combines explicit geometry, differentiable rasterization, and fast optimization. Recent extensions, including MonoGS [16] for dense tracking, Loop-Splat [17] for loop closure, Splat-SLAM [18] for global joint optimization, and HI-SLAM2 [19] for monocular refinement, demonstrate strong results. Still, they inherit the limitations of monocular input: limited FoV, scale ambiguity, and degraded performance in low-texture or occluded regions. These drawbacks highlight the unmet potential of fusing multi-view observations with the efficiency of Gaussian splatting. While multi-agent extensions [20] also support multiple cameras, they cannot benefit from calibrated rigs.

Leveraging a calibrated multi-camera rig with k spatially overlapping views offers rich observational redundancy but presents challenges in fusing dense RGB streams into a unified Gaussian representation, specifically, maintaining inter-camera scale consistency, achieving drift-free tracking, and enabling efficient online mapping with Gaussians. We propose MCGS-SLAM, to the best of our knowledge, the first fully vision-based multi-camera SLAM system built upon 3D Gaussian Splatting with purely RGB input. MCGS-SLAM jointly estimates accurate camera trajectories and high-fidelity 3D reconstructions by fusing raw RGB inputs into a globally consistent Gaussian map. Our framework also supports RGB-D inputs, but this paper focuses on the RGB-only setting. Central to our framework is a Multi-Camera Bundle Adjustment (MCBA) module that jointly optimizes pose and dense depth across views via photometric and geometric consistency. To ensure metric-scale alignment, we introduce a complementary module that leverages low-rank geometric priors from a learned network. These components enable scalable Gaussian optimization, yielding reconstructions with sharp geometry and photorealistic textures under wide baselines. Fig. 1 presents the unified multi-camera Gaussian Splatting reconstruction and rendering results of our method. Our contributions are as follows.

- An efficient multi-camera Gaussian SLAM system supporting RGB inputs, with joint optimization over camera poses and 3DGS maps.
- A unified multi-camera framework that combines Multi-Camera Bundle Adjustment (MCBA) and Joint Depth-Scale Alignment (JDSA), jointly optimizing photometric consistency, geometric priors, and global scale alignment across views.
- A practical and scalable implementation that generalizes across real-world and synthetic benchmarks, demonstrating strong performance in both geometry and appearance.

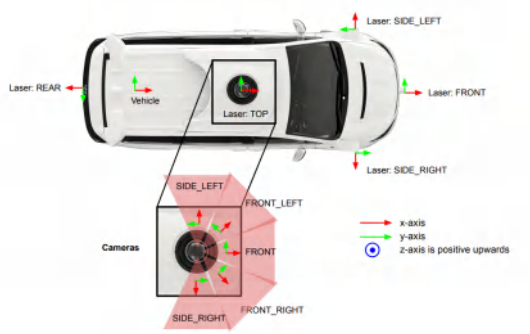

Fig. 2: The sensor suite integrates multiple wide-angle RGB cameras centrally mounted on the vehicle's roof in Waymo Open Dataset [21], whose fan-shaped fields of view collectively provide full 240° coverage. This configuration enables high-density observations for multi-camera SLAM and autonomous driving algorithms.

Through these innovations, MCGS-SLAM bridges the gap between wide-baseline multi-camera tracking and dense 3D Gaussian mapping, laying the groundwork for next-generation robotic perception, digital twin construction, and autonomous systems at scale.

II. Preliminaries

This section introduces the core concepts underpinning our multi-camera Gaussian Splatting SLAM framework. We first review the dense SLAM formulation and the multi-camera setting, followed by an overview of Recurrent Field Transforms and learning-based SLAM front-ends such as DROID-SLAM and BAMF-SLAM. Finally, we present the 3D Gaussian Splatting representation, which serves as the foundational structure of our mapping system.

A. Problem Setting: Dense Multi-Camera SLAM

1) Dense SLAM: Given a temporally ordered stream of color (or color-depth) images I_t captured at time t by a calibrated rig with k camera views, dense SLAM jointly estimates the metric camera trajectory $\mathbf{T} = \{\mathbf{T}_t\}_{t=0}^{L}$ with $\mathbf{T}_t \in \mathrm{SE}(3)$ and a continuous scene map $\boldsymbol{\mathcal{M}}$ by minimizing photometric and geometric residuals across all pixels. To ensure both temporal and spatial consistency, we define a set of frame pairs $(t, t') \in \mathcal{E}$, where t' denotes either a temporally adjacent frame or one selected via keyframe heuristics. The overall objective is formulated as

$$\underset{\mathbf{T},\boldsymbol{\mathcal{M}}}{\arg\min} \sum_{(t,t')\in\mathcal{E}} \left\| I_t - I_{t'} \circ \Pi\big(\mathbf{T}_{tt'}\, \Pi^{-1}(\mathbf{p}_t, d_t)\big) \right\|_{\rho}, \quad (1)$$

where Π and Π^{-1} denote the projection and back-projection functions, d_t is the depth at pixel $\mathbf{p}_t$, and $\rho(\cdot)$ is the robust ℓ_2 penalty function. The transformation $\mathbf{T}_{tt'}$ denotes the relative camera pose from frame t to t'. The optimization in Equation (1) is performed at full image resolution, enabling recovery of dense scene geometry.

2) Multi-Camera Setting: The calibrated multi-camera system is defined by fixed extrinsic transformations $\mathbf{T}_C^B \in \mathrm{SE}(3)$, which map points from each individual camera frame C to a shared body frame B. As illustrated in Fig. 2, modern

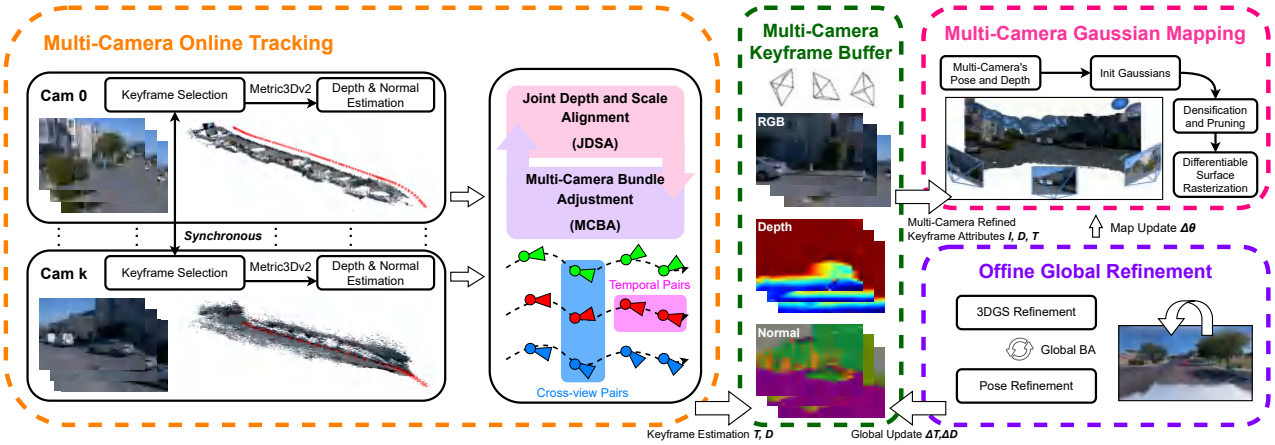

Fig. 3: Our method performs real-time SLAM by fusing synchronized inputs from a multi-camera rig into a unified 3D Gaussian map. It first selects keyframes and estimates depth and normal maps for each camera, then jointly optimizes poses and depths via multi-camera bundle adjustment and scale-consistent depth alignment. Refined keyframes are fused into a dense Gaussian map using differentiable rasterization, interleaved with densification and pruning. An optional offline stage further refines camera trajectories and map quality. The system supports RGB inputs, enabling accurate tracking and photorealistic reconstruction.

automotive datasets such as the Waymo Open Dataset provide time-synchronized, wide-baseline camera clusters composed of multiple global-shutter RGB sensors with accurate intrinsic and extrinsic calibrations. These offer a compelling testbed for SLAM systems, as they introduce strong parallax, large fields of view, and a complex environment.

B. Recurrent Field Transforms and Learning-based SLAM

Recurrent Field Transforms (RFT) extend the RAFT family of recurrent optical flow networks to iteratively refine dense correspondences between two views. Given a current reprojection $\hat{\mathbf{p}}_{ij}$, RFT predicts a flow increment $\boldsymbol{\delta}_{ij}$ and an associated per-pixel confidence weight w_{ij}. The refined target location is defined as $\tilde{\mathbf{p}}_{ij} = \hat{\mathbf{p}}_{ij} + \boldsymbol{\delta}_{ij}$ and is used to minimize the reprojection error. During optimization, the resulting weighted residual is inserted into the normal equations of bundle adjustment (BA) [4] as

$$r_{ij} = \left\| \tilde{\mathbf{p}}_{ij} - \Pi\big(\mathbf{T}_{ij}\Pi^{-1}(\mathbf{p}_i, d_i)\big) \right\|^2_{w_{ij}} \tag{2}$$

where Π and Π^{-1} denote projection and back-projection, respectively. Equation (2) forms the foundation of the dense, differentiable front-end in DROID-SLAM. To tightly couple correspondence estimation and geometric optimization, DROID-SLAM augments classical photometric BA with RFT, treating optical flow as a latent variable updated via a gated recurrent unit (GRU). This formulation enables joint, real-time optimization of camera poses, per-frame depths, and inter-frame flow, achieving state-of-the-art accuracy in monocular visual odometry. BAMF-SLAM builds upon DROID-SLAM by generalizing it to multi-fisheye camera systems, with optional visual-inertial integration.

C. 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) represents the scene as a set $\{(\boldsymbol{\mu}_i, \Sigma_i, \alpha_i, \mathbf{c}_i)\}_{i=1}^{M}$ of M anisotropic Gaussians, where each Gaussian is defined by its mean $\boldsymbol{\mu}_i \in \mathbb{R}^3$, covariance $\Sigma_i \in \mathbb{R}^{3\times3}$, opacity $\alpha_i \in \mathbb{R}$, and RGB color $\mathbf{c}_i \in \mathbb{R}^3$. Under a camera pose $\mathbf{T}_i \in \mathrm{SE}(3)$, a 3D Gaussian is projected into an elliptical footprint on the image plane as

$$\boldsymbol{\mu}_i' = \pi(\mathbf{T}_i\boldsymbol{\mu}_i), \qquad \Sigma_i' = J\,R\,\Sigma_i R^\top J^\top, \tag{3}$$

where R is the rotational component of $\mathbf{T}$ and J denotes the Jacobian of the perspective projection [15]. The resulting splats are composited in a front-to-back order using α-blending to produce color and depth images as

$$\begin{aligned} \hat{C}(\mathbf{p}) &= \sum_{i\in\mathcal{N}_\mathbf{p}} \mathbf{c}_i\alpha_i \prod_{j<i}(1-\alpha_j), \\ \hat{D}(\mathbf{p}) &= \sum_{i\in\mathcal{N}_\mathbf{p}} d_i\alpha_i \prod_{j<i}(1-\alpha_j). \end{aligned} \tag{4}$$

where $\mathcal{N}_\mathbf{p}$ denotes the set of Gaussians intersecting the ray. Here, c_i and d_i are the color and depth of the i-th Gaussian, respectively, and α_i represents its contribution to pixel translucency, obtained from the Gaussian's opacity at the ray Gaussian intersection. The projection and blending operations in Equations (3) and (4) are fully differentiable, allowing gradients to be backpropagated with respect to both the camera pose $\mathbf{T}$ and all Gaussian parameters.

III. Method

This section presents MCGS-SLAM, a dense multi-camera SLAM pipeline that integrates learning-based tracking with a differentiable 3D Gaussian map representation. An overview of the system architecture is shown in Fig. 3. The system follows a two stage design, an online pipeline, and an optional offline refinement. Specifically, the online stage consists of (i) multi-camera tracking, (ii) a shared keyframe buffer, and (iii) incremental Gaussian mapping. During online tracking, the system addresses the scale ambiguity of monocular priors and performs multi-camera bundle adjustment (MCBA) to jointly optimize per-view depths and rig poses under both temporal and cross-view constraints. In the optional offline stage, all rig poses and Gaussian parameters are jointly

optimized to enforce global consistency and further improve the geometric and photometric fidelity of the reconstruction.

A. Online Multi-Camera Tracking

1) KeyFrame Selection, Depth and Normal Estimation: For each synchronized RGB frame, we compute the average Recurrent Field Transform (RFT) flow relative to the current reference keyframe. If the flow magnitude exceeds a threshold, the frame is promoted to a multi-camera keyframe, $K_t := \{I_t, d_t^+, \mathbf{n}_t^+\}$, where d_t^+ and $\mathbf{n}_t^+$ denote the per-pixel depth and surface normal maps. These are obtained from Metric3Dv2 [22], which also allows our system to support RGB-D input. Keyframes are stored in a shared buffer accessible to both tracking and mapping threads. Although the depths from Metric3Dv2 are metric, they often suffer from noise and inconsistent scaling across viewpoints, leading to misaligned poses and depths. Our proposed MCBA module corrects this by jointly refining poses and depths, enforcing geometric consistency and scale alignment across the rig.

2) Joint Depth and Scale Alignment (JDSA): Depth maps predicted for each RGB camera are only defined up to an unknown, spatially varying scale. To compensate for this ambiguity, [23] introduce a learnable $m \times n$ scale grid $\mathbf{s}_t$ for each keyframe. This grid is bilinearly interpolated to yield a per-pixel scale factor $B_t(\mathbf{p}, \mathbf{s}_t)$, which relates the predicted and optimized depths as $\tilde{d}_t(\mathbf{p}) = d_t^+(\mathbf{p}) \cdot B_t(\mathbf{p}, \mathbf{s}_t)$, where d_t^+ denotes the monocular depth map and $\tilde{d}_t$ the rescaled depth used during optimization. However, directly coupling the scale factors with bundle adjustment, by jointly optimizing camera poses, depths, and scale coefficients, has been shown to cause unstable convergence and scale drift [19]. To mitigate this, we adopt the Joint Depth and Scale Alignment (JDSA) formulation proposed in [19], which introduces a dedicated loss function as

$$\begin{aligned}\arg\min_{\mathbf{s},\mathbf{d}} \quad & \sum_{(i,j)\in\mathcal{E}} \left\| \tilde{\mathbf{p}}_{ij} - \Pi\left(\mathbf{T}_{ij}\Pi^{-1}(\mathbf{p}_i, \mathbf{d}_i)\right)\right\|^2_{\omega_{ij}} + \\ & \sum_{i\in\mathcal{V}} \left\| \tilde{\mathbf{d}}_i \cdot B_i(\mathbf{p}_i, \mathbf{s}_i) - \mathbf{d}_i \right\|^2, \end{aligned} \tag{5}$$

where the node set $\mathcal{V}$ consists of keyframes, each associated with a pose $T \in SE(3)$ and an estimated depth map d. The edge set $\mathcal{E}$ connects keyframes that exhibit sufficient overlap, as determined by their optical flow correspondences. The first term enforces multi-view photometric and geometric consistency, and the second term aligns scaled depths to the optimized depths. By interleaving JDSA with local multi-camera bundle adjustment, our system achieves stable scale calibration and improved depth initialization.

3) Multi-Camera Bundle Adjustment (MCBA): To jointly optimize camera poses and dense depth maps, we minimize a weighted photometric reprojection loss over both temporal and cross-view image pairs. Specifically, for each valid correspondence between a source view (i, C_i) and a target view (j, C_j), we define the following objective:

$$\arg\min_{\mathbf{T},\mathbf{d}} \sum_{(i,j)\in\mathcal{E}} \left\| \tilde{\mathbf{p}}_{ij} - \Pi_{C_j}\left(\hat{\mathbf{T}}_{ij} \cdot \Pi_{C_i}^{-1}(\mathbf{p}_i, d_i)\right)\right\|^2_{w_{ij}}, \tag{6}$$

where $\mathbf{T} \in \mathrm{SE}(3)$ denotes the body pose, and d_i is the estimated inverse depth parametrization in view (i, C_i). The function $\Pi_{C_i}^{-1}(\cdot)$ back-projects the pixel using the intrinsics of camera C_i, while $\Pi_{C_j}(\cdot)$ reprojects it into the target view. The norm $\|\cdot\|_{w_{ij}}$ incorporates a confidence w_{ij} per pixel predicted by the RFT module in multi-camera settings. The transformation $\hat{\mathbf{T}}_{ij}$ maps 3D points from the source to the target camera frame, and is defined differently based on the type of correspondence as

- **Temporal pairs** (i.e., same camera across time):

$$\hat{\mathbf{T}}_{ij} = \mathbf{T}_C^B \, \mathbf{T}_j^{-1} \, \mathbf{T}_i \, \mathbf{T}_C^{B^{-1}}, \tag{7}$$

  where $\mathbf{T}_C^B$ is the known extrinsic between the camera frame C and the body frame B.
- **Cross-view pairs** (i.e., different cameras at the same timestamp):

$$\hat{\mathbf{T}}_{ij} = \mathbf{T}_{C_i C_j}, \tag{8}$$

  which is the pre-calibrated extrinsic between camera C_i and camera C_j.

This unified formulation allows for simultaneous optimization over both time-varying motion and multi-camera geometry in a single bundle adjustment framework. The resulting non-linear least-squares problem is solved via a damped Gauss-Newton method, yielding a block-structured linear system of the form as

$$\begin{bmatrix} \mathbf{B} & \mathbf{E} \\ \mathbf{E}^\top & \mathbf{C} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\xi} \\ \Delta\mathbf{d} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}, \tag{9}$$

where $\Delta\boldsymbol{\xi} \in \mathbb{R}^6$ represents the pose update in the Lie algebra of SE(3), applied via $\Delta\mathbf{T} = \exp(\Delta\boldsymbol{\xi})$ as in [4]. Matrices $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{E}$ correspond to the Hessian blocks with respect to pose, depth, and their coupling terms, while $\mathbf{v}$ and $\mathbf{w}$ are the respective residual gradients. Since the pose block $\mathbf{B}$ is typically much smaller than the depth block $\mathbf{C}$, we solve the system efficiently using the Schur complement. The pose update is obtained via

$$\begin{aligned} \Delta\boldsymbol{\xi} &= \left[\mathbf{B} - \mathbf{E}\mathbf{C}^{-1}\mathbf{E}^\top\right]^{-1}\left(\mathbf{v} - \mathbf{E}\mathbf{C}^{-1}\mathbf{w}\right), \\ \Delta\mathbf{d} &= \mathbf{C}^{-1}\left(\mathbf{w} - \mathbf{E}^\top\Delta\boldsymbol{\xi}\right). \end{aligned} \tag{10}$$

In the implementation, the depth Hessian $\mathbf{C}$ is diagonal and thus admits a cheap closed-form inverse $\mathbf{C}^{-1} = 1/\mathbf{C}$.

B. Multi-Camera Gaussian Mapping

1) Gaussian Initialization and Maintenance: After each MCBA and JDSA update, we back-project the depth map of the latest keyframe K_t into 3D space to initialize new Gaussian primitives. For each valid pixel $\mathbf{p}$, a Gaussian g_i is created with mean $\boldsymbol{\mu}_i \in \mathbb{R}^3$ corresponding to the back-projected 3D point, and covariance $\Sigma_i \in \mathbb{R}^{3\times3}$ estimated from the average distance to its three nearest neighbors. To keep the map compact yet expressive, the system alternates every few iterations between two complementary operations: **(1) densification**, which adds Gaussians at previously unobserved pixels to grow underrepresented regions; and **(2) pruning**, which removes nearly transparent Gaussians to reduce redundancy and computational overhead.

2) Differentiable Rasterization and Losses: We follow [19] and avoid depth bias by analytically intersecting each viewing ray with the ellipsoidal surface defined by the anisotropic Gaussian, yielding a more accurate intersection depth. Each Gaussian is jointly optimized through a multi-term loss function per keyframe K_t as

$$\mathcal{L} = \lambda_c \left\| \hat{C}_t - I_t \right\|_2 + \lambda_d \left\| \hat{D}_t - d_t \right\|_2 + \lambda_n \left\| 1 - \left\langle \hat{\mathbf{n}}_t, \mathbf{n}_t^{\text{pri}} \right\rangle \right\|_2 + \lambda_s \left\| \mathbf{s}_t - \bar{\mathbf{s}} \right\|_2, \quad (11)$$

where $\hat{C}_t$ and $\hat{D}_t$ denote the rendered color and depth from the viewpoint of the MCBA-refined camera pose, d_t is the depth refined via MCBA and JDSA, $\hat{\mathbf{n}}_t$ and $\mathbf{n}_t^+$ represent the rendered and estimated surface normals by Metric3Dv2, and $\mathbf{s}_t$ and $\bar{\mathbf{s}}$ denote the current and average scale of the corresponding Gaussian ellipsoids, respectively. Optimization is performed using the optimizer for a fixed number of iterations per keyframe.

3) Pose-Consistent Gaussian Updates: When the pose of a keyframe K_t is updated via MCBA or loop closure by a relative transform $\Delta\mathbf{T}_t \in \mathrm{SE}(3)$, we propagate the update to all Gaussians g_i anchored in that frame as

$$\boldsymbol{\mu}_i \leftarrow \Delta\mathbf{T}_t \cdot \boldsymbol{\mu}_i, \quad \Sigma_i \leftarrow \mathbf{R}(\Delta\mathbf{T}_t) \cdot \Sigma_i \cdot \mathbf{R}^\top(\Delta\mathbf{T}_t), \quad (12)$$

where $\mathbf{R}(\cdot)$ extracts the rotational component of $\Delta\mathbf{T}_t$. If scale changes are introduced via scale updates, we additionally rescale the ellipsoids as

$$\mathbf{s}_i \leftarrow s_t \cdot \mathbf{s}_i. \quad (13)$$

This deformation ensures consistency of the 3D map without requiring re-initialization or re-rendering, enabling efficient and flexible map maintenance.

C. Offline Global Refinement

after the real-time pipeline finishes, we apply two global refinement stages to enhance the consistency and overall quality of the reconstruction.

1) Global Bundle Adjustment: All keyframes that includes synthetically inserted views, are jointly optimized via global bundle adjustment. The optimization minimizes both photometric and geometric residuals across all overlapping image pairs, refining the camera poses and improving the consistency of the reconstructed scene geometry.

2) Joint Pose and 3DGS Map Refinement: In the final stage, we jointly optimize all 3D Gaussian parameters $\boldsymbol{\Theta} := \{\boldsymbol{\mu}, \Sigma, \alpha, \mathbf{c}\}$, along with per-frame exposure matrices $\mathbf{A}_t$ and camera poses $\mathbf{T}_t$. Gradients are backpropagated through the differentiable rasterization pipeline to minimize a weighted combination of photometric, depth, normal, and scale regularization losses. This optimization stage effectively reduces global drift and improves geometric accuracy.

IV. Results

A. Datasets, Metrics, and Protocol

We evaluated MCGS-SLAM on both real-world and synthetic datasets. For real-world experiments, we employ the **Waymo Open Dataset** [21], which provides urban driving sequences with five synchronized wide-angle roof cameras. We select three of them, as this already ensures a sufficiently wide front-facing field of view while keeping GPU memory usage manageable. The proposed formulation directly generalizes to additional cameras, since all views share the same rig pose and are incorporated as additional cross-view edges in MCBA and mapping. We further use the **Oxford Spires Dataset** [24], which contains large-scale Oxford landmarks recorded by three fisheye cameras with LiDAR/IMU ground truth. For synthetic evaluation, we adopt the **AirSim** [25] simulator with three photorealistic UE5 environments, captured using a four-camera aircraft rig in the simulation setting. Reconstruction quality is quantified using standard image-based metrics: PSNR (↑), SSIM (↑) and LPIPS (↓) - computed over all keyframes after mapping. The trajectory accuracy is measured by the absolute trajectory error (ATE, meters; ↓) after Sim(3)-alignment with the ground truth. We compared against baselines using their publicly released implementations and default recommended settings. For better readability, the result tables highlight the top three results with **first**, second, and third.

Method	Metric	100613	106762	134763	152706
NICER-SLAM [13]	PSNR ↑	12.91	13.09	8.79	11.32
	SSIM ↑	0.498	0.587	0.330	0.611
	LPIPS ↓	0.695	0.626	0.791	0.754
GLORIE-SLAM [14]	PSNR ↑	25.78	27.35	25.71	24.90
	SSIM ↑	**0.916**	**0.918**	**0.883**	**0.878**
	LPIPS ↓	0.282	0.272	0.365	0.338
MonoGS [16]	PSNR ↑	20.58	22.31	21.41	22.34
	SSIM ↑	0.674	0.741	0.620	0.784
	LPIPS ↓	0.607	0.503	0.625	0.641
DROID-Splat [26]	PSNR ↑	26.77	27.21	26.20	25.92
	SSIM ↑	0.829	0.864	0.792	0.782
	LPIPS ↓	0.273	0.281	0.376	0.512
Photo-SLAM [27]	PSNR ↑	19.03	20.26	21.28	20.84
	SSIM ↑	0.640	0.712	0.624	0.759
	LPIPS ↓	0.527	0.440	0.471	0.466
MCGS-SLAM (Ours)	PSNR ↑	**27.09**	**27.70**	**27.20**	**28.45**
	SSIM ↑	0.830	0.819	0.813	0.797
	LPIPS ↓	**0.223**	**0.262**	**0.233**	**0.330**

TABLE I: Appearance reconstruction comparison of different methods on four scenes of the Waymo dataset [21] (**Real-World Dataset**). MCGS-SLAM achieves the best PSNR and LPIPS results, highlighted as **first**, second and third.

B. Rendering Results Study

Table I and Table II present quantitative appearance metrics, while Fig. 4 and Fig. 5 show reconstruction results in different Waymo and AirSim environments. On the four held-out urban sequences from the Waymo dataset, MCGS-SLAM consistently ranks among the top two performers, demonstrating strong photometric fidelity and perceptual quality. In contrast, competing methods report inferior LPIPS values and fail to reconstruct critical side-view structures, such as alley facades, that are clearly recovered by MCGS-SLAM (see Fig. 4). This advantage stems from the wide field of view (FoV) provided by the multi-camera rig (Fig. 2), which

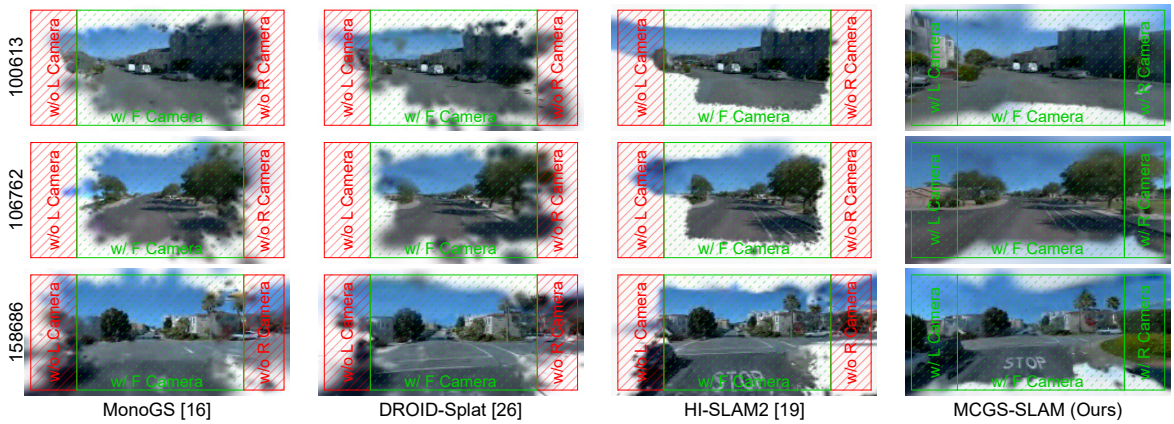

Fig. 4: Qualitative results on the Waymo dataset [21] (**Real-World Dataset**). MCGS-SLAM reconstructs urban scenes with higher fidelity and completeness, preserving structural details and textures that are often missed by monocular methods.

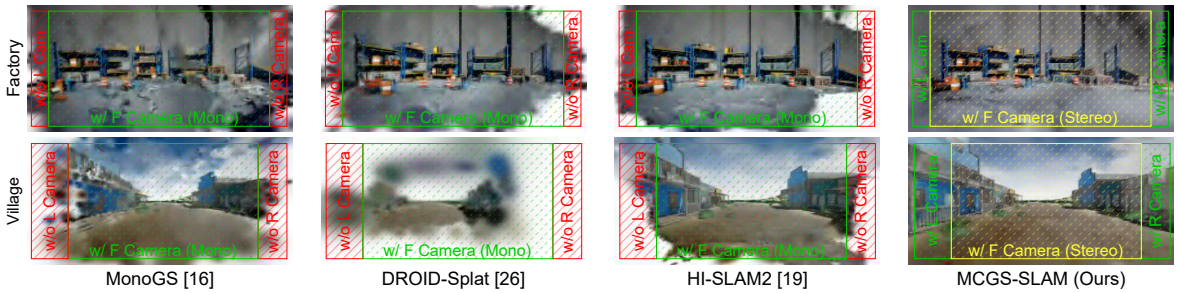

Fig. 5: MCGS-SLAM produces complete reconstructions on AirSim [25] (**Synthetic Dataset**), demonstrated in the Factory and Village environments, where it preserves structural details and textures that monocular methods often fail to capture.

enables MCGS-SLAM to resolve occluded elements such as building corners. Although GLORIE-SLAM and DROID-Splat occasionally reconstruct sharper specular surfaces, their limited spatial coverage leads to incomplete scene geometry. Overall, MCGS-SLAM achieves a better balance between reconstruction fidelity and spatial completeness, making it particularly well suited for complex urban environments.

Similar trends are observed in the AirSim benchmark (Table II and Fig. 5), where MCGS-SLAM consistently ranks among the top two methods across all environments. In the Garden scene, it surpasses all single-camera baselines by approximately 4 dB PSNR, highlighting the benefit of leveraging complementary viewpoints from a wide-baseline rig. In the Factory scene, although Photo-SLAM achieves the highest PSNR and SSIM, MCGS-SLAM yields cleaner and more geometrically consistent reconstructions thanks to dense cross-view constraints. The Village scene, characterized by abrupt turns and large FoV discontinuities, remains challenging for single-camera baselines. By exploiting multi-view priors and robust depth-scale alignment, MCGS-SLAM reconstructs sharper structures and more complete geometry even under wide-baseline motion.

Method	Metric	Garden	Factory	Village
NICER-SLAM [13]	PSNR ↑	12.30	9.84	11.18
	SSIM ↑	0.450	0.332	0.504
	LPIPS ↓	0.801	0.690	0.653
GLORIE-SLAM [14]	PSNR ↑	24.50	23.39	17.56
	SSIM ↑	0.849	0.888	0.494
	LPIPS ↓	0.351	0.346	0.712
MonoGS [16]	PSNR ↑	25.59	21.45	21.39
	SSIM ↑	0.766	0.760	0.689
	LPIPS ↓	0.258	0.175	0.444
DROID-Splat [26]	PSNR ↑	24.12	26.50	17.25
	SSIM ↑	0.822	0.898	0.669
	LPIPS ↓	0.242	0.107	0.652
Photo-SLAM [27]	PSNR ↑	25.47	**28.38**	26.77
	SSIM ↑	0.775	0.923	0.805
	LPIPS ↓	0.156	**0.041**	**0.205**
MCGS-SLAM (Ours)	PSNR ↑	**29.36**	28.37	**28.10**
	SSIM ↑	**0.879**	**0.924**	**0.853**
	LPIPS ↓	**0.126**	0.083	0.219

TABLE II: Quantitative comparison of appearance reconstructions of different methods on 3 scenes of the AirSim dataset [25] (**Synthetic Dataset**). Best results are highlighted as **first**, second and third.

C. Tracking Results Study

Table III and Table IV summarize the quantitative tracking accuracy on diverse real-world datasets. These Waymo sequences use original images without distortion correction, providing a more challenging and realistic setting to evaluate tracking robustness in autonomous driving conditions. For the Oxford Spires dataset [24], the original fisheye images were undistorted to fit the pinhole camera model, and sequences with severe distortion were excluded. MCGS-SLAM achieves the lowest average ATE and ranks first in five of eight Waymo sequences, demonstrating strong robustness to wide baselines and complex environments. It maintains low drift even in difficult cases such as 100613 and 106762, where methods like MonoGS show large trajectory errors. To account for the scene-dependent behavior of the JDSA

Method	Metric	100613	158686	132384	134763	152706	153495	106762	163453	Avg.
NICER-SLAM [13]	ATE [m] ↓	2.351	2.362	56.363	2.642	19.409	19.782	1.634	14.708	14.906
MonoGS [16]	ATE [m] ↓	10.727	10.101	12.033	3.394	9.073	1.628	19.532	9.189	9.459
Splat-SLAM [18]	ATE [m] ↓	0.802	2.575	1.133	1.625	1.092	2.572	**1.973**	3.115	1.861
HI-SLAM2 [19]	ATE [m] ↓	0.790	1.782	**0.888**	1.281	**0.964**	1.389	2.132	2.558	1.473
MCGS-SLAM (Ours)	ATE [m] ↓	**0.398**	**0.612**	1.242	**1.107**	2.554	**1.180**	2.366	**0.927**	**1.298**

TABLE III: Quantitative comparison of tracking accuracy (ATE RMSE) across different methods and scenes on the Waymo dataset [21]. MCGS-SLAM yields the best average results. Best results are highlighted as **first**, second and third.

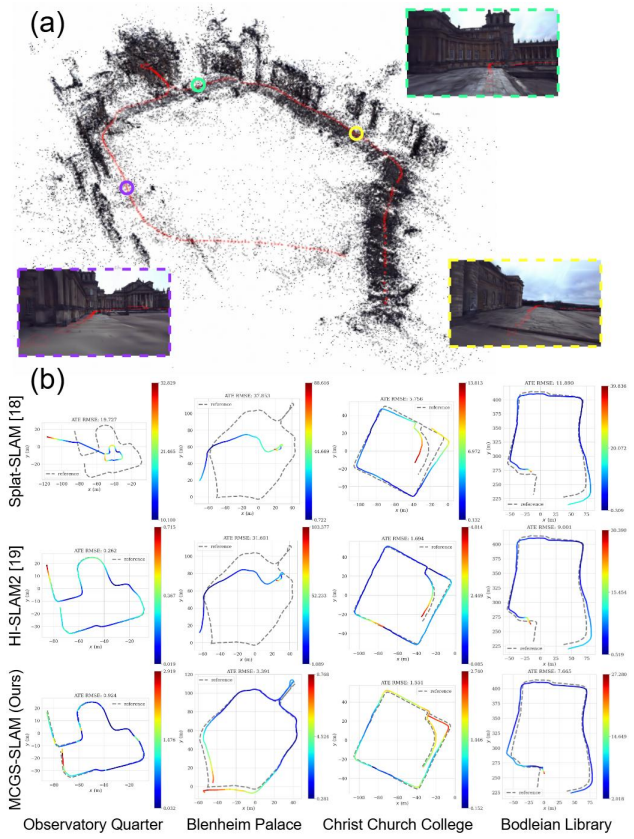

Fig. 6: Tracking performance on the different sequences from the Oxford Spires Dataset [24]. (a) Top-down view of the reconstructed scene and MCGS-SLAM trajectory on the Blenheim Palace sequence. (b) Trajectory comparison on four representative sequences against ground truth, Splat-SLAM [18], HI-SLAM2 [19], and our MCGS-SLAM.

module, which improves metric-scale consistency, but can occasionally increase drift, we evaluated both configurations and reported the better result. The performance gains mainly stem from the joint optimization of inter-camera depth and pose in the MCBA module, supported by effective scale alignment via JDSA. Although HI-SLAM2 and Splat-SLAM perform competitively in terms of ATE, their monocular design leads to greater drift in long or wide-baseline sequences.

Similar trends appear in the Oxford Spires dataset shown in Fig. 6 and Table IV, which features complex large-scale outdoor scenes with frequent occlusions and strong parallax. MCGS-SLAM again delivers superior performance, achieving the lowest average ATE and outperforming all baselines by a significant margin. In contrast, MonoGS fails on several sequences, leading to heavily degraded ATE values, while HI-SLAM2 and Splat-SLAM suffer from scale ambiguity

Method	Library	Palace	College	Observatory
NICER-SLAM [13]	77.593	41.593	24.580	23.621
MonoGS [16]	FAILED	29.451	30.794	11.814
Splat-SLAM [18]	11.890	37.853	5.756	19.727
HI-SLAM2 [19]	9.001	31.601	1.694	**0.262**
MCGS-SLAM (Ours)	**7.665**	**3.391**	**1.551**	0.924

TABLE IV: Quantitative comparison of tracking accuracy (ATE RMSE) across different methods and scenes on the Oxford Spires Dataset [24] (Bodleian Library, Blenheim Palace, Christ Church College, and Observatory Quarter). Best results are highlighted as **first**, second and third.

and tracking discontinuities. The ability of MCGS-SLAM to exploit multi-view redundancy and recover occluded structures is essential in challenging large-scale environments.

D. Ablation Study

Table V analyzes the contributions of the Joint Depth-Scale Alignment (JDSA) module and the monocular depth maps predicted by Metric3Dv2 [22]. Removing both components leads to a notable degradation in performance, with PSNR dropping, SSIM decreasing, and LPIPS increasing substantially. Introducing only the previous depth improves PSNR, but results in subtle double-edge artifacts due to inconsistent estimates of the per-camera scale. In contrast, the full configuration of MCGS-SLAM, with both JDSA and estimated monocular depth, achieves the best scores across all three metrics, justifying its superior photometric accuracy. Although reconstruction quality generally improves, ATE slightly degrades in a few challenging cases due to depth errors, indicating the need for stronger depth predictors. While depth estimates provide better initialization for optimization in multi-camera setting, the absence of JDSA still leads to inter-camera scale inconsistencies and visible artifacts. By enforcing per-camera scale alignment, JDSA compensates for missing or noisy depth estimates and, together with scale-consistent optimization, improves depth reliability.

E. Summary and Discussion

MCGS-SLAM demonstrates its most significant advantages on challenging wide-FoV multi-camera sequences with strong parallax, frequent occlusions, and abrupt viewpoint changes, where monocular baselines often suffer from drift and incomplete reconstruction. However, the current framework assumes a predominantly static environment and does not explicitly model dynamic objects, which may lead to degraded performance in highly dynamic scenes. Overall, MCGS-SLAM is particularly suitable for calibrated camera rigs in autonomous driving and robotic navigation, where wide coverage and metric consistency are critical.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
w/o Depth* + w/o JDSA	25.01	0.751	0.404
w/ Depth* + w/o JDSA	27.02	0.809	0.271
MCGS-SLAM (full)	**27.17**	**0.816**	**0.262**

*From Metric3Dv2 [22]

TABLE V: Ablation of Joint Depth-Scale Alignment (JDSA) on the Waymo dataset [21], averaged over 6 sequences (134763, 106762, 132384, 152706, 153495, and 163453). Best results are highlighted as **first**, second and third.

V. Conclusion and Future Work

In this work, we introduced MCGS-SLAM, a fully vision-based SLAM framework that constructs unified 3D Gaussian maps from synchronized multi-camera RGB inputs. By jointly optimizing camera poses and dense depths through Multi-Camera Bundle Adjustment (MCBA) and enforcing inter-camera scale consistency via our proposed Joint Depth-Scale Alignment (JDSA) module, the system achieves real-time, photorealistic, and geometrically consistent reconstructions. MCGS-SLAM performs well in both synthetic and real-world scenarios. Our analysis highlights the critical role of wide-baseline, overlapping views in enhancing scene completeness and robustness, particularly under occlusion and viewpoint discontinuities where monocular systems often fail. Looking ahead, promising directions include integrating inertial or event-based sensing for improved performance in dynamic or low-texture environments, extending the system to support asynchronous rigs, and further incorporating semantic understanding for object-aware mapping.

References

[1] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021.

[2] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[3] Rui Wang, Martin Schworer, and Daniel Cremers. Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In *Proceedings of the IEEE international conference on computer vision*, pages 3903–3911, 2017.

[4] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

[5] Yuheng Qiu, Yutian Chen, Zihao Zhang, Wenshan Wang, and Sebastian Scherer. Mac-vo: Metrics-aware covariance for learning-based stereo visual odometry mac-vo. github. io. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3803–3814. IEEE, 2025.

[6] Steffen Urban and Stefan Hinz. Multicol-slam-a modular real-time multi-camera slam system. *arXiv preprint arXiv:1610.07336*, 2016.

[7] Juichung Kuo, Manasi Muglikar, Zichao Zhang, and Davide Scaramuzza. Redesigning slam for arbitrary multi-camera systems. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2116–2122. IEEE, 2020.

[8] Wei Zhang, Sen Wang, Xingliang Dong, Rongwei Guo, and Norbert Haala. Bamf-slam: Bundle adjusted multi-fisheye visual-inertial slam using recurrent field transforms. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6232–6238. IEEE, 2023.

[9] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011.

[10] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual slam for rgb-d cameras. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pages 2100–2106. IEEE, 2013.

[11] 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. *Communications of the ACM*, 65(1):99–106, 2021.

[12] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

[13] Zihan Zhu, Songyou Peng, Viktor Larsson, Zhaopeng Cui, Martin R Oswald, Andreas Geiger, and Marc Pollefeys. Nicer-slam: Neural implicit scene encoding for rgb slam. In *2024 International Conference on 3D Vision (3DV)*, pages 42–52. IEEE, 2024.

[14] Ganlin Zhang, Erik Sandström, Youmin Zhang, Manthan Patel, Luc Van Gool, and Martin R Oswald. Glorie-slam: Globally optimized rgb-only implicit encoding point cloud slam. *arXiv:2403.19549*, 2024.

[15] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[16] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18039–18048, 2024.

[17] Liyuan Zhu, Yue Li, Erik Sandström, Shengyu Huang, Konrad Schindler, and Iro Armeni. Loopsplat: Loop closure by registering 3d gaussian splats. In *2025 International Conference on 3D Vision (3DV)*, pages 156–167. IEEE, 2025.

[18] Erik Sandström, Ganlin Zhang, Keisuke Tateno, Michael Oechsle, Michael Niemeyer, Youmin Zhang, Manthan Patel, Luc Van Gool, Martin Oswald, and Federico Tombari. Splat-slam: Globally optimized rgb-only slam with 3d gaussians. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1680–1691, 2025.

[19] Wei Zhang, Qing Cheng, David Skuddis, Niclas Zeller, Daniel Cremers, and Norbert Haala. Hi-slam2: Geometry-aware gaussian slam for fast monocular scene reconstruction. *IEEE Transactions on Robotics*, 41:6478–6493, 2025.

[20] Vladimir Yugay, Theo Gevers, and Martin R Oswald. Magic-slam: Multi-agent gaussian globally consistent slam. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6741–6750, 2025.

[21] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[22] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[23] Wei Zhang, Tiecheng Sun, Sen Wang, Qing Cheng, and Norbert Haala. Hi-slam: Monocular real-time dense mapping with hybrid implicit fields. *IEEE Robotics and Automation Letters*, 9(2):1548–1555, 2023.

[24] Yifu Tao, Miguel Ángel Muñoz-Bañón, Lintong Zhang, Jiahao Wang, Lanke Frank Tarimo Fu, and Maurice Fallon. The oxford spires dataset: Benchmarking large-scale lidar-visual localisation, reconstruction and radiance field methods. *The International Journal of Robotics Research*, page 02783649251369905, 2024.

[25] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics: Results of the 11th international conference*, pages 621–635. Springer, 2017.

[26] Christian Homeyer, Leon Begiristain, and Christoph Schnörr. Droid-splat combining end-to-end slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2767–2777, 2025.

[27] Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular stereo and rgb-d cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21584–21593, 2024.