

Robust and Efficient Monocular 3D Gaussian SLAM for Kilometer-Scale Outdoor Scenes

Sicheng Yu^{*}, Dongxu Shen^{*}, Beizhen Zhao, Guanzhi Ding, and Hao Wang[†]

The Hong Kong University of Science and Technology (Guangzhou), China

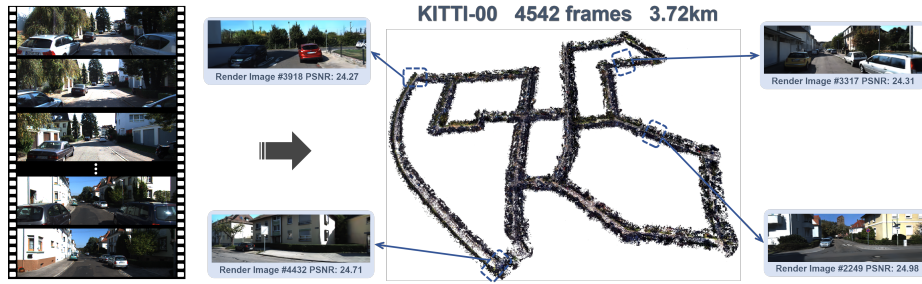


Fig. 1: KiloGS-SLAM achieves robust pose tracking and precise scene reconstruction in kilometer-scale outdoor environments using only monocular RGB input, enabling high-fidelity novel view synthesis.

Abstract. Scaling monocular 3D Gaussian Splatting (3DGS) SLAM to kilometer-level outdoor environments poses two tightly coupled challenges: fragile long-term pose tracking and excessive memory overhead during large-scale mapping. In this paper, we propose KiloGS-SLAM, a highly efficient and robust monocular 3DGS-SLAM system that jointly addresses both bottlenecks. Since high-fidelity scene reconstruction fundamentally relies on drift-free camera poses, we first introduce a motion-adaptive hybrid tracking module. This module features a condition-triggered three-tier solving pipeline. It dynamically switches between Essential matrix and PnP models to handle geometric degeneracies. An on-demand foundation model can also be activated to rescue the trajectory from catastrophic drift. To ensure the system can sustain these long trajectories without memory exhaustion, we subsequently design a lifecycle-managed Gaussian mapping strategy. By integrating probabilistic initialization with chunk-based multi-view densification and pruning, this full-pipeline optimization effectively reduces primitive redundancy while preserving high-frequency details. Together, the robust tracking guarantees the geometric foundation required for accurate mapping, while the memory-efficient lifecycle-managed mapping enables large-scale operation. Extensive experiments across three challenging outdoor datasets demonstrate that our approach achieves state-of-the-art tracking accuracy and rendering quality, successfully scaling to sequences of over 10,000 frames on a single GPU. Project Page: <https://3dagentworld.github.io/KiloGS-SLAM/>

^{*}Equal contribution, [†]Corresponding author: haowang@hkust-gz.edu.cn

Keywords: Monocular SLAM · 3D Gaussian Splatting · Large-Scale Reconstruction · Outdoor Environments

1 Introduction

While 3D Gaussian Splatting (3DGS) [19, 24, 49] has emerged as a powerful representation for SLAM [18, 41, 45], offering real-time and photorealistic rendering, scaling it to kilometer-level outdoor scenes remains challenging. Although multi-sensor setups (e.g., LiDAR [9, 14, 43] or IMU [44]) can provide reliable geometric initialization, monocular RGB-only approaches remain highly desirable for their minimal hardware dependency. However, current monocular 3DGS-SLAM systems are fundamentally bottlenecked by two tightly coupled limitations: **fragile pose estimators** in complex environments and **excessive memory consumption** over long trajectories.

First, robust long-term tracking in outdoor environments is difficult. Traditional multi-view geometric solvers [1, 37, 38] frequently collapse when confronting planar degeneracies (e.g., flat roads), pure rotations, or dynamic occlusions. Decoupled 3DGS-SLAM systems [8, 35] that inherit these geometric frontends naturally suffer from the same vulnerability, often resulting in irreversible cumulative drift. While some recent approaches [2, 7, 25, 29] attempt to replace geometric solvers with 3D foundation models, these inference-based methods are computationally heavy and lack rigorous geometric verification, limiting their continuous tracking precision.

Second, the large scale and extreme depth variations of outdoor scenes make Gaussian mapping highly memory-intensive. Naive initialization and densification strategies [26] inevitably cause an unconstrained growing of Gaussian primitives. Existing systems either introduce heavy hierarchical structures that fail to address the root cause of floater accumulation [8], or employ local heuristic pruning [44] that ignores multi-view consistency, often erroneously deleting valid scene structures.

To address these issues, we propose **KiloGS-SLAM**, a RGB-only monocular 3DGS-SLAM system designed specifically for kilometer-scale outdoor environments (Fig. 1). We argue that successfully scaling 3DGS-SLAM requires a systematic design: the pose estimator must provide a robust geometric foundation under varying conditions, while the mapping process must effectively minimize redundancy to sustain continuous operation.

To ensure stable tracking, we introduce a motion-adaptive hybrid tracking module. Instead of relying on a single solver, we design a condition-triggered, three-tier solving pipeline. Under normal conditions, it utilizes an efficient Essential matrix model. When standard geometry degrades (e.g., flat-ground scenes), it dynamically switches to a Perspective-n-Point (PnP) model. Crucially, to prevent severe drift, the system continuously verifies tracking consistency against motion priors. Any detected anomaly triggers an on-demand foundation model [21] activation. By using foundation models as a supplementary model rather

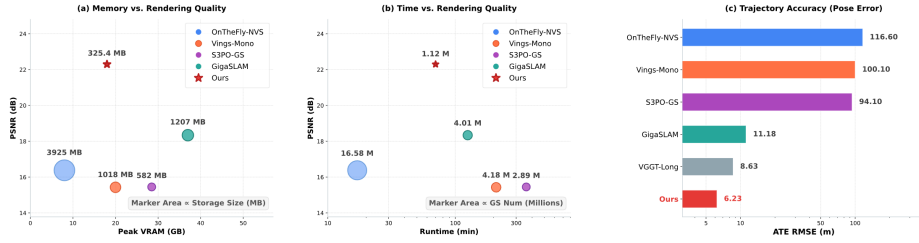


Fig. 2: Performance comparison on KITTI-00. Our method strikes the optimal balance between rendering quality, runtime, and memory overhead, while achieving the lowest camera tracking error.

than a primary pose estimator, we achieve the speed and precision of geometric solvers, backed by the robustness of deep priors.

Given these reliable poses, we tackle the memory bottleneck by proposing a lifecycle-managed Gaussian mapping strategy. Rather than allowing Gaussians to grow randomly, this full-pipeline approach utilizes depth gradients and the Difference of Gaussians (DoG) for probabilistic initialization. As the map evolves, we continuously evaluate multi-view photometric errors within local chunks to guide densification and pruning. This lifecycle-managed optimization effectively suppresses redundant floaters while preserving high-frequency textural details, ensuring an optimal and lightweight primitive distribution.

We evaluate our system on the Waymo [36], KITTI [12], and KITTI-360 [22] datasets, encompassing nearly 30 sequences ranging from 200 to over 13,000 frames. Experimental results demonstrate that our approach achieves SOTA performance in both tracking and rendering. By achieving an optimal balance between visual fidelity and computational constraints (Fig. 2), our system scales seamlessly to ultra-long scenes. Our main contributions are:

1. We propose a motion-adaptive hybrid tracking module featuring a condition-triggered three-tier pipeline. By dynamically switching between geometric solvers and an on-demand foundation model activation, it ensures robust tracking in kilometer-level outdoor sequences.
2. We design a lifecycle-managed Gaussian mapping strategy. By integrating probabilistic initialization and chunk-based multi-view pruning and densification into a continuous optimization pipeline, it minimizes redundancy and memory overhead while maintaining high reconstruction quality.
3. We achieve SOTA tracking and rendering performance across three demanding datasets, successfully scaling to continuous sequences exceeding 10,000 frames on a single GPU.

2 Related Work

2.1 Visual Odometry & SLAM in Large-Scale Environments

Monocular SLAM in large-scale outdoor environments remains a challenging task due to environmental complexities and long-term drift. Early methods like

LSD-SLAM [10] and ORB-SLAM series [1,28] pioneered large-scale tracking but remain fragile in low-texture or dynamic outdoor scenarios. With the advent of deep learning, DF-VO [48] and RoMeO [6] leveraged depth priors to enhance tracking stability, while DROID-SLAM [37] and DPVO [38] significantly enhanced robustness via recurrent optical flow and dense bundle adjustment. DPV-SLAM [23] further mitigated long-term drift by incorporating appearance-geometric loop closure. Despite these advances, such methods rely heavily on multi-view geometric constraints, which frequently fail in complex outdoor scenarios characterized by planar degeneracies (e.g., flat roads), pure rotational motions, and dynamic occlusions, rendering these systems prone to tracking instability.

Recent 3D foundation models, like DUST3R [42], MAST3R [21], and VGGT [40], attempt to transcend these limitations by directly regressing 3D geometry and pose from large-scale pre-training. To adapt these models to long sequences, recent approaches employ a “chunk-and-align” strategy [3,4,7,25,29]. However, they are limited by the local consistency of the base model, often suffering from temporal discontinuities and geometric misalignments at sub-map boundaries.

This motivates our design to integrate foundation models not as pose solvers, but as an on-demand activation mechanism for unreliable geometric estimation.

2.2 Gaussian Splatting SLAM

The emergence of 3D Gaussian Splatting (3DGS) [19,24,50,51] has revolutionized SLAM with its real-time and high-fidelity rendering. Early 3DGS-based systems integrated explicit Gaussian representations for joint tracking and mapping [15,18,26,45,47] or loosely coupled them with Visual Odometry [13,17,35,52]. While effective in indoor environments, these methods struggle in outdoor sequences, bottlenecked by fragile frontend robustness and excessive memory overhead.

To address these scalability issues, sensor-fused approaches [9,14,43] leverage LiDAR for geometric initialization, ensuring structural accuracy but restricting deployment due to hardware dependency. VINGS-Mono [44] introduces a visual-inertial framework with score-based pruning and concurrent CPU-GPU scheduling to mitigate memory pressure; however, its stability relies on IMU.

In the more challenging monocular setting, OpenGS-SLAM [46] and S3PO-GS [5] incorporate geometric priors and scale correction. While effective for street-level scenes, they lack efficient data structures for kilometer-scale sequences, causing memory exhaustion and drift. OnTheFly-NVS [27] achieves rapid reconstruction but suffers from significant accuracy degradation in long sequences. Most related to our work is GigaSLAM [8], the first pure visual kilometer-scale 3DGS-SLAM system using a Hierarchical Sparse Voxel Map. Yet, it suffers from Gaussian redundancy and relies on heavy post-optimization.

To bridge this gap, we propose a lifecycle-managed Gaussian mapping strategy, which integrates probabilistic Gaussian initialization and adaptive management based on multi-view consistency, ensuring efficient, high-fidelity mapping.

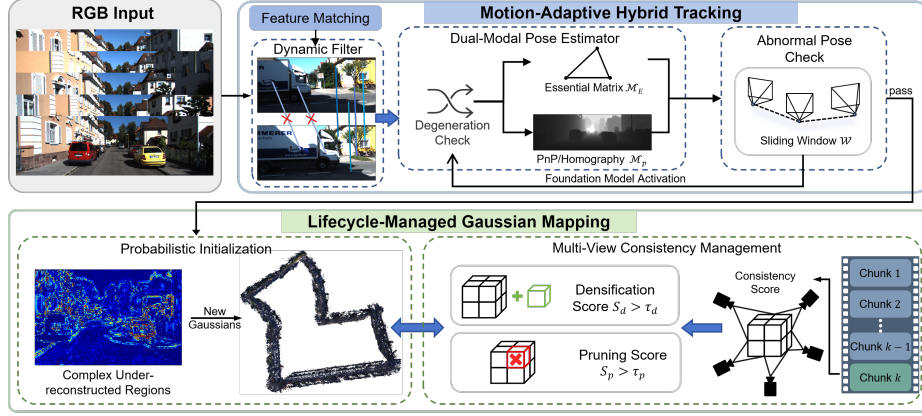


Fig. 3: Framework of KiloGS-SLAM. Input RGB frames first undergo sparse matching and dynamic filtering once entering the tracking module. A dual-modal pose estimator dynamically switches between Essential and PnP matrices to handle degeneracies. The estimation is verified against a sliding-window motion prior, triggering an on-demand foundation model dense matching upon failure. Valid poses proceed to the Mapping module. New Gaussians are probabilistically initialized strictly in complex, under-reconstructed regions. During map optimization, chunk-based multi-view consistency scores are evaluated to guide continuous densification and pruning.

3 Method

As illustrated in Fig. 3, KiloGS-SLAM sequentially processes incoming RGB frames, estimates camera poses with Motion-adaptive Hybrid Tracking, and performs Lifecycle-managed Gaussian Mapping over keyframes.

3.1 Motion-Adaptive Hybrid Tracking

The core of long-sequence pose tracking lies in consistent robustness. In kilometer-scale trajectories, tracking errors in merely a few frames can lead to irreversible cumulative drift. Fundamentally, the robustness of monocular visual odometry depends on two factors: the numerical stability of the pose solver and the reliability of feature correspondences. Complex outdoor environments severely challenge both. Large depth variations and planar degeneracies often render a single geometric solver ill-conditioned. Simultaneously, dramatic illumination changes, textureless regions, and high-frequency structural clutter (*e.g.*, foliage) frequently induce massive spatial aliasing, leading to erroneous matches.

To tackle this dual challenge, we propose an adaptive hybrid tracking module. First, we utilize sparse optical flow statistics to filter out dynamic interference. The purified input is then fed into a condition-triggered, three-tier solving pipeline to combat varying degrees of geometric degeneracies. Crucially, to prevent severe mismatches from inducing physically implausible trajectory drift,

we introduce a sliding-window motion consistency check. If an anomaly is detected, the system triggers a foundation model-based dense matching activation, rescuing the trajectory from severe visual degradation.

Dual-Modal Pose Estimator Relying on a single pose solver is fragile in complex outdoor scenarios due to common geometric degeneracies. Therefore, we introduce a dual-modal pose estimator as the primary component of our tracking module. Specifically, after establishing frame-to-frame correspondences via lightweight feature matching [39], we adaptively switch between an epipolar-based Essential matrix model ($\mathcal{M}_E$) [30] and a Perspective-n-Point/Homography model ($\mathcal{M}_P$) [20] for pose estimation. This switching is inspired by the Geometric Robust Information Criterion (GRIC) [48] score, effectively handling geometric degeneracies like pure rotations or flat-ground scenes.

Motion Consistency Verification To detect tracking anomalies caused by massive mismatches, we validate the estimated pose against a local motion prior. We establish a Constant-Velocity (CV) motion prior using a short-term sliding window $\mathcal{W}$ of recent keyframes. Let $\Delta T_k = \{\Delta R_k, \Delta \mathbf{t}_k\} \in SE(3)$ denote the relative transformation at step k . The motion prior for the current frame t is approximated by the moving average over the local window:

$$\Delta \tilde{\mathbf{t}}_t = \frac{1}{|\mathcal{W}| - 1} \sum_{i=t-|\mathcal{W}|+1}^{t-1} \Delta \mathbf{t}_i, \quad \Delta \tilde{R}_t \approx \Delta R_{t-1}. \quad (1)$$

Upon receiving the raw visual estimation $\Delta T_t = \{\Delta R_t, \Delta \mathbf{t}_t\}$, we evaluate its physical plausibility against two critical criteria:

Rotational Consistency. Sudden severe rotations are physically impossible for large vehicles. We compute the geodesic distance d_R between the estimated rotation and the motion prior in $SO(3)$ space:

$$d_R(\Delta R_t, \Delta \tilde{R}_t) = \arccos \left(\frac{\text{Tr}(\Delta \tilde{R}_t^\top \Delta R_t) - 1}{2} \right) > \tau_R, \quad (2)$$

where τ_R is the maximum allowable angular deviation. A violation typically indicates planar degeneracy (*e.g.*, a "ghost rotation" induced by ground shadows).

Non-Holonomic Constraint. According to Ackermann steering geometry [32], a vehicle's primary translation must align with its forward axis $\mathbf{u}_{\text{forward}}$ (typically the Z-axis in camera coordinates). To prevent physically impossible lateral side-slips or vertical jumps, we check the cosine similarity:

$$\frac{\Delta \mathbf{t}_t \cdot \mathbf{u}_{\text{forward}}}{\|\Delta \mathbf{t}_t\|_2} < \cos(\tau_\theta). \quad (3)$$

If the current geometric estimation violates either the rotational consistency in Eq. (2) or the non-holonomic bound in Eq. (3), the frame is flagged as a motion anomaly, triggering a 3D foundation model-based dense matching activation.

Foundation Model Activation When the motion verification detects an anomaly, it typically indicates a catastrophic failure of standard sparse feature matching, often caused by severe shadows or textureless regions. To recover the camera pose, we introduce a dense matching activation driven by a 3D foundation model [21]. Since continuous inference is impractical for real-time SLAM, this mechanism is strictly on-demand. It is activated only when the initial sparse matching yields insufficient inliers or when the motion verification fails. Upon activation, we sparsify the dense correspondences using a confidence-based Non-Maximum Suppression (NMS). These refined matches subsequently replace the erroneous sparse points in the Dual-Modal Pose Estimator, successfully rescuing the trajectory without sacrificing overall efficiency.

Dynamic Filtering via Sparse Optical Flow To prevent salient moving objects from interfering with camera tracking, we introduce a lightweight pruning step without relying on external semantic segmentation modules [16, 33]. We treat the established correspondences $\mathcal{C} = \{(\mathbf{u}_{t-1}^i, \mathbf{u}_t^i)\}$ as a sparse optical flow field.

Exploiting the tight clustering of static background flows, we identify dynamic agents as statistical outliers. Let $f_i = \|\mathbf{u}_t^i - \mathbf{u}_{t-1}^i\|_2$ denote the L_2 norm of each flow vector, and $\mathcal{F}$ denote the set of all magnitudes. Using robust statistics, we filter out features satisfying:

$$f_i > \text{Median}(\mathcal{F}) + 3 \cdot \text{MAD}(\mathcal{F}), \quad (4)$$

where MAD is the Median Absolute Deviation. By statistically eliminating high-velocity dynamic objects and large-displacement mismatches, this operation significantly reduces the overall outlier ratio, enhancing the effectiveness of the subsequent geometric RANSAC.

3.2 Map Representation and Optimization

To achieve memory-efficient and accurate mapping in large-scale outdoor environments, our backend builds the map upon the voxelized representation of Scaffold-GS [24] and the hierarchical structure from [8], continuously optimizing the scene within a local sliding window.

Voxelized Gaussian Representation We divide the 3D scene into a grid of sparse voxels, parameterized by a set of anchor points $\mathcal{A} = \{\mathbf{p}_i \in \mathbb{R}^3\}_{i=1}^N$ centered in each voxel. Each anchor $\mathbf{p}_i$ stores a neural context feature $\mathbf{f}_i \in \mathbb{R}^{32}$, a local scaling factor $\mathbf{I}_i \in \mathbb{R}^3$, and k positional offsets $\mathbf{O}_i \in \mathbb{R}^{k \times 3}$. During rendering, MLPs dynamically decode these attributes into k explicit 3D Gaussian primitives, conditioned on the camera-to-anchor distance and viewing direction. By encoding the scene into sparse voxels and neural features, this representation effectively models view-dependent effects and improves memory efficiency.

Hierarchical Anchor Management Outdoor driving scenes exhibit vast depth ranges. Representing distant backgrounds (*e.g.*, sky, far buildings) with dense primitives yields diminishing returns and incurs severe computational bottlenecks. Inspired by [8], we incorporate a hierarchical Level-of-Detail (LoD) architecture to manage the sparse anchor points.

The scene is discretized into m resolution levels with corresponding voxel sizes $\{\epsilon_1, \dots, \epsilon_m\}$. When new observations are integrated, the initialized 3D position $\mathbf{x}$ is assigned to a specific level l based on its distance to the camera. The position of the new anchor $\mathbf{p}_i$ is then quantized to the corresponding voxel grid: $\mathbf{p}_i = \lfloor \mathbf{x}/\epsilon_l \rfloor \cdot \epsilon_l$. During rasterization, a distance-aware mask activates fine-grained anchors for nearby regions and coarse anchors for distant areas. This hierarchical management preserves visual quality while substantially reducing memory overhead.

Sliding-Window Map Optimization To maintain real-time efficiency, we jointly optimize anchor attributes and MLP parameters within a local sliding window $\mathcal{W}$. The total mapping loss combines a photometric loss $\mathcal{L}_{\text{photo}}$ (using L_1 and SSIM), a geometric loss $\mathcal{L}_{\text{geo}}$, and an isotropic regularization $\mathcal{L}_{\text{iso}}$ [26] to penalize highly elongated primitives:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{photo}} + \lambda_g \mathcal{L}_{\text{geo}} + \lambda_{\text{iso}} \mathcal{L}_{\text{iso}}, \quad (5)$$

where λ_g and λ_{iso} are weighting coefficients.

While these operations ensure scalability, the primary challenge for large-scale 3DGS-SLAM remains strategically managing anchor points to maximize accuracy and minimize redundancy.

3.3 Lifecycle-Managed Gaussian Mapping

To build a robust geometric baseline, we manage the Gaussian lifecycle in two stages: probabilistic insertion of new anchors at incoming keyframes, followed by a scalable, chunk-based refinement mechanism that leverages multi-view photometric errors for densification and pruning.

Probabilistic Gaussian Initialization To balance rendering quality and memory footprint, new Gaussian anchors are instantiated only in under-reconstructed regions with high textural or geometric complexity, rather than relying on uniform random sampling [5, 8, 46].

Inspired by the LoG-based sampling in [27], we employ the Difference of Gaussians (DoG) for robust multi-scale texture edge detection:

$$\text{DoG}(u, v) = \min(|(G_{\sigma_1} - G_{\sigma_2}) * I(u, v)|, 1), \quad (6)$$

where G_σ denotes a 2D Gaussian blur kernel, and $*$ is the convolution operation.

Table 1: Camera tracking results (ATE RMSE [m]) on the KITTI dataset. **LC** denotes loop closure, “-” indicates failure. The best results are highlighted in **bold**, and the second-best are underlined. Our method achieves the best overall performance, remaining highly competitive even without loop closure.

Methods	LC	Avg.	00	02	03	04	05	06	07	08	09	10
<i>Seq. frames</i>	-	2169	4542	4661	801	271	2761	1101	1101	4077	1591	1201
<i>Seq. length (m)</i>	-	2012	3724	5067	560	393	2205	1232	649	3222	1705	919
ORB-SLAM2 [28]	✓	10.8	7.62	18.6	1.52	1.34	<u>4.57</u>	18.4	1.83	38.8	8.39	6.80
DROID-SLAM [37]	✓	96.5	121	239	13.2	2.7	85.9	66.4	43.4	138	157	63.3
DPVO [38]	✗	57.3	113	120	2.55	<u>1.12</u>	60.8	52.9	18.2	113	77.2	13.1
DPV-SLAM++ [23]	✓	22.3	<u>8.30</u>	39.4	2.50	0.81	5.74	11.6	1.58	110	76.9	13.6
MASt3R-SLAM [29]	✓	89.5	-	-	18.1	88.9	159	92.3	-	-	-	-
VGGT-SLAM [25]	✓	152	190	296	169	13.1	137	148	61.2	207	179	126
VGGT-Long [7]	✓	22.4	8.63	52.9	8.74	4.21	9.84	4.67	2.66	73.1	31.8	27.7
DA3-Streaming [7]	✓	21.9	22.9	68.3	5.15	1.84	18.4	5.48	4.89	24.9	54.4	13.5
Splat-SLAM [35]	✓	45.4	83.1	-	3.40	1.72	33.0	130	14.3	52.0	27.4	63.5
OnTheFly-NVS [27]	✗	77.5	116	169	18.1	9.08	78.4	82.2	39.3	-	114	46.4
S3PO-GS [5]	✗	77.1	94.1	264	22.9	14.5	56.5	49.4	20.9	60.9	91.9	96.8
Vings-Mono [44]	✓	74.3	100	206	9.70	1.67	73.0	93.8	31.9	124	76.7	25.2
GigaSLAM [8]	✓	5.46	11.2	<u>11.3</u>	<u>1.41</u>	1.58	4.92	<u>2.11</u>	3.17	<u>13.5</u>	<u>3.24</u>	2.15
Ours (w/o LC)	✗	<u>4.26</u>	8.94	<u>11.3</u>	1.18	1.53	4.86	2.66	<u>1.80</u>	4.90	3.06	2.38
Ours (w/ LC)	✓	3.46	6.23	8.70	1.18	1.53	4.32	1.26	1.06	4.90	3.06	<u>2.38</u>

However, purely textural metrics often fail in textureless yet geometrically complex areas (*e.g.*, structural corners). Thus, we introduce the normalized spatial depth gradient $\|\nabla D(u, v)\|$ as a geometric complement. For the initial frame, the sampling probability $P(u, v)$ combines textural and geometric saliency:

$$P(u, v) = \min \left(\lambda \text{DoG}(u, v) + (1 - \lambda) \overline{\|\nabla D(u, v)\|}, 1 \right), \quad (7)$$

where $\overline{(\cdot)}$ denotes min-max normalization to $[0, 1]$, and λ balances the two terms.

For subsequent keyframes, to prevent redundant insertions, we render image $\hat{I}$ and depth $\hat{D}$ to compute a rendered probability map $\hat{P}(u, v)$. This acts as a spatial penalty, defining the final spawning probability via the residual:

$$P_{kf}(u, v) = \max \left(P(u, v) - \hat{P}(u, v), 0 \right). \quad (8)$$

To align this continuous probability map with our discrete hierarchical voxels, we apply a 3×3 Non-Maximum Suppression (NMS) over P_{kf} . The selected 2D pixels are unprojected into 3D space and quantized into hierarchical voxel anchors according to the LoD rules defined in Section 3.2. Finally, anchor attributes are initialized using geometry-aware heuristics from [27].

Multi-View Consistency Management Standard 3DGS relies on positional gradients for densification, which introduces redundancy and is highly sensitive to pose noise in outdoor SLAM. Recent methods [44] introduce local heuristics for Gaussian management, but they ignore multi-view consistency, frequently

Table 2: Camera Tracking Results (ATE RMSE [m]) on the KITTI-360 Dataset. Our method consistently ranks within the top two across all sequences, securing the best performance in the majority of them.

Methods	LC	Avg.	0000	0002	0003	0004	0005	0006	0007	0009	0010
<i>Seq. frames</i>	-	7898	10505	13888	1010	11047	6289	9185	2890	13246	3022
<i>Seq. length (m)</i>	-	6984	8403	11616	1379	9975	4691	7980	4888	10580	3343
ORB-SLAM2 [28]	✓	47.1	45.5	32.3	11.5	72.1	24.8	43.9	134	-	12.6
DROID-SLAM [37]	✓	156	110	233	10.4	169	139	113	433	165	31.9
DPV-SLAM++ [23]	✓	<u>32.5</u>	<u>27.8</u>	30.6	3.01	<u>40.5</u>	20.4	39.6	36.5	<u>65.2</u>	<u>30.7</u>
VGGT-Long [7]	✓	344	305	466	17.6	703	138	248	491	241	487
DA3-Streaming [7]	✓	243	233	480	21.9	318	192	256	371	177	140
OnTheFly-NVS [27]	✗	165	164	247	32.3	179	132	182	257	-	127
Vings-Mono [44]	✓	171	168	-	77.6	-	-	-	298	-	142
S3PO-GS [5]	✗	167	-	-	55.5	-	83.5	-	260	-	271
GigaSLAM [8]	✓	64.7	42.6	35.1	24.4	206	<u>15.0</u>	<u>37.6</u>	59.5	106	56.2
Ours	✓	25.7	23.9	17.7	<u>17.3</u>	36.7	9.54	33.7	<u>50.0</u>	23.2	<u>19.5</u>

leading to erroneous pruning. Inspired by [34], we utilize multi-view reprojection errors to accurately guide anchor densification and pruning.

Since global multi-view evaluation is computationally prohibitive for large-scale SLAM, we divide the historical keyframe sequence into local chunks of size C . Within each chunk c , we compute a densification score $S_d^{(i,c)}$ (identifying under-reconstructed regions) and a pruning score $S_p^{(i,c)}$ (measuring error contribution) for each anchor i . Detailed formulations are provided in the supplementary material. To robustly update the global voxel grid, we aggregate these chunk-level scores using two distinct strategies:

Global Densification Rules. Anchor i is globally marked for densification if its score exceeds a threshold τ_d in *any* individual chunk:

$$S_d(i) = \bigvee_c \left(S_d^{(i,c)} > \tau_d \right). \quad (9)$$

Global Pruning Rules. Since S_p reflects rendering degradation, we evaluate the worst-case impact of anchor i by taking its maximum score across all chunks:

$$S_p(i) = \max_c \left(S_p^{(i,c)} \right). \quad (10)$$

Anchor i is pruned if $S_p(i) > \tau_p$, eliminating Gaussian floaters while preserving valid structures across complex views.

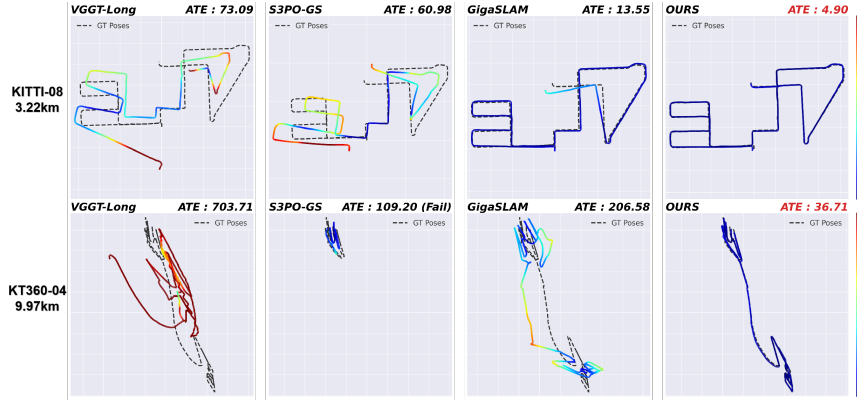
4 Experiments

4.1 Experimental Setup

Datasets and Metrics. We evaluate our system on three outdoor datasets: Waymo [36] (nine 200-frame sequences following [5]), KITTI [12] (up to 4,661

Table 3: Camera Tracking Results (ATE RMSE [m]) on the Waymo Dataset.

Methods	Avg.	13476	100613	106762	132384	152706	153495	158686	163453	405841
ORB-SLAM2 [28]	3.95	1.08	4.77	8.01	4.55	6.79	2.21	4.25	2.59	1.31
DROID-SLAM [37]	10.4	6.71	3.95	3.45	19.4	5.18	6.42	11.5	33.1	3.92
DPVO [38]	0.85	0.93	0.59	0.29	0.68	0.72	1.30	1.12	0.99	1.04
MASt3R-SLAM [29]	1.87	3.15	0.86	0.98	1.03	0.82	4.30	1.87	2.46	1.35
VGGT-SLAM [25]	5.81	5.54	6.32	3.55	4.64	5.46	6.93	10.0	5.58	4.27
VGGT-Long [7]	1.46	0.71	1.31	2.64	1.67	0.62	1.17	1.57	1.92	1.61
DA3-Streaming [7]	1.35	<u>0.61</u>	1.08	0.77	<u>0.42</u>	1.76	2.21	1.96	1.70	0.92
Splat-SLAM [35]	<u>0.59</u>	0.62	0.40	0.41	0.24	0.63	1.20	0.66	<u>0.65</u>	0.54
OnTheFly-NVS [27]	2.66	5.65	1.60	1.17	1.43	2.24	3.05	1.74	4.68	2.36
OpenGS-SLAM [46]	1.38	2.74	0.33	0.89	0.78	0.62	1.62	1.18	3.31	0.93
S3PO-GS [5]	0.92	1.04	0.76	0.37	0.62	0.79	2.11	0.69	1.42	<u>0.51</u>
Vings-Mono [44]	2.28	2.69	1.11	3.89	3.88	1.69	2.09	2.15	2.26	0.73
GigaSLAM [8]	0.87	2.54	<u>0.34</u>	0.38	0.97	<u>0.48</u>	0.64	<u>0.63</u>	1.32	0.52
Ours	0.51	0.58	0.33	<u>0.36</u>	0.55	0.46	<u>0.68</u>	0.59	0.59	0.47

**Fig. 4: Tracking trajectories.** Our method maintains robust pose estimation and global consistency across long-term sequences, whereas other baselines inevitably suffer from significant drift in localized challenging segments.

frames, 5 km), and the highly challenging KITTI-360 [22] (up to 13,888 frames, 11.6 km). We omit KITTI-01, as its extreme highway speeds and lack of trackable near-field features cause universal monocular degeneracy.

For tracking accuracy, we report the Absolute Trajectory Error (ATE) RMSE [m] on keyframes. Mapping quality is assessed via PSNR, SSIM, and LPIPS. To evaluate system efficiency for 3DGS-based methods, we report training time, peak VRAM usage, number of Gaussians, and map storage size.

Baseline Methods. We compare our approach with two distinct categories of state-of-the-art baselines. The first category is long-sequence VO/SLAM systems: ORB-SLAM2 [28], DF-VO [48], DROID-SLAM [37], DPVO [38], DPV-SLAM [23], MASt3R-SLAM [21], VGGT-SLAM [25], VGGT-Long [7] and DA3-Streaming [7]. The second category is monocular 3DGS-SLAM frameworks: Splat-SLAM [35], OnTheFly-NVS [27], Vings-Mono [44], OpenGS-SLAM [46],

Table 4: Rendering Results on three Datasets. Only 3DGS-based methods are compared. We report the average results across all evaluated scenes. Our method achieves the highest rendering quality across all three datasets.

Method	Waymo			KITTI			KITTI-360		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Splat-SLAM [35]	25.34	0.832	0.378	17.22	0.611	0.620	-	-	-
OnTheFly-NVS [27]	25.16	0.830	<u>0.349</u>	16.20	0.541	0.520	14.19	0.493	0.577
Vings-Mono [44]	19.25	0.623	0.587	15.07	0.556	0.544	14.95	0.502	0.711
OpenGS-SLAM [46]	24.83	0.816	0.415	-	-	-	-	-	-
S3PO-GS [5]	<u>26.84</u>	<u>0.844</u>	0.367	15.40	0.494	0.686	13.76	0.468	0.730
GigaSLAM [8]	26.12	0.822	0.472	<u>20.99</u>	<u>0.650</u>	<u>0.504</u>	<u>20.92</u>	<u>0.661</u>	<u>0.516</u>
Ours	28.36	0.862	0.339	22.52	0.693	0.437	21.60	0.671	0.476

Table 5: Efficiency Analysis. Results on Waymo are averaged across sequences (≈ 200 frames), while KITTI *Seq. 00* and KITTI-360 *Seq. 0005* are selected to test long-term scalability. **T**: Runtime (min); **Mem**: Peak VRAM (GB); **Size**: Map Storage (MB); **#GS**: Number of Gaussians. *OOM* denotes Out-Of-Memory.

Method	Waymo (Avg.)				KITTI (Seq. 00)				KITTI-360 (Seq. 0005)			
	T	Mem	#GS	Size	T	Mem	#GS	Size	T	Mem	#GS	Size
Splat-SLAM	40.3	16.4	420K	96.2	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
OnTheFly-NVS	1.87	8.3	528K	124	17.1	8.3	16.6M	3925	22.4	11.2	31.1M	7336
Vings-Mono	9.77	3.5	1.1M	262	229	21.5	4.18M	1018	Fail	Fail	Fail	Fail
OpenGS-SLAM	33.6	15.2	<u>306K</u>	48.6	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
S3PO-GS	31.1	12.8	320K	50.1	320	25.0	<u>2.89M</u>	<u>582</u>	462	38.5	<u>3.21M</u>	<u>560</u>
GigaSLAM	4.67	21.8	365K	75.6	125	36.9	4.01M	1207	229	47.2	6.09M	2009
Ours	<u>4.52</u>	8.6	180K	37.3	<u>70.0</u>	<u>18.1</u>	1.12M	325	<u>127</u>	<u>23.7</u>	1.14M	380

S3PO-GS [5], and GigaSLAM [8]. Particularly for GigaSLAM, which relies on an extremely time-consuming color refinement process, we restrict its iterations to an acceptable bound to ensure fair efficiency comparisons.

Implementation Details. Our system is implemented in PyTorch and evaluated on single NVIDIA RTX A6000 GPU. We employ MAST3R [21] as the dense matching model for frontend fallback mechanism, and a metric depth model [31] as geometric prior. For global consistency, we integrate DBoW2 [11] loop closure and Sim(3) pose graph optimization, followed by rigid transformations to align the 3D Gaussian map. Further details on loop closure, keyframe selection, and hyperparameters are provided in the supplementary material.

4.2 Results and Comparisons

Camera Tracking. We evaluate the tracking robustness of our system across varying sequence lengths. On the short-sequence Waymo dataset (Table 3), our method achieves the lowest average ATE (0.51m) and consistently ranks in the top two across most scenes. For the medium-scale KITTI dataset (Table 1), our approach maintains the lowest trajectory error (Avg. 3.46m). Notably, even



Fig. 5: Qualitative rendering comparisons on Waymo (top 2) and KITTI (bottom 3). Our method preserves superior structural and textural details in both near-field and distant regions, whereas other baselines struggle with noticeable blurring and artifacts.

without global loop closure (*w/o LC*), our visual odometry remains highly competitive against full SLAM systems, validating the effectiveness of our motion-adaptive hybrid tracking module.

Furthermore, on the ultra-long, highly challenging KITTI-360 dataset (Table 2), our system demonstrates unparalleled robustness. It secures the best average ATE (25.7m) and ranks within the top two across all individual sequences, successfully avoiding the catastrophic drifts that frequently plague other baselines. The qualitative trajectory visualizations (Fig. 4) further corroborate our global consistency over multi-kilometer runs.

Mapping and Efficiency. Table 4 demonstrates that our method yields the highest rendering quality across all three datasets (*e.g.*, 28.36 dB PSNR on Waymo). Qualitative results (Fig. 5) further verify our superior preservation of high-frequency details in both near-field textures and distant structures.

Table 5 reveals the underlying efficiency behind this performance. Systems like OnTheFly-NVS and Vings-Mono rely on CPU-GPU scheduling to manage memory pressure. While OnTheFly-NVS achieves extremely fast speeds, both methods suffer from massive Gaussian redundancy (*e.g.*, 31.1M primitives for OnTheFly-NVS on KITTI-360) and significantly degraded rendering quality. In contrast, our lifecycle-managed Gaussian mapping fundamentally compresses the scene representation. We maintain the strictly lowest primitive count (only 1.14M on KITTI-360) and minimal storage (380 MB), alongside highly competitive VRAM consumption (23.7 GB) and an acceptable runtime. This optimal trade-off between fidelity and resource constraints is crucial for seamlessly scaling to kilometer-scale outdoor environments.

Table 6: Ablation study of Motion-Adaptive Hybrid Tracking module on KITTI-07.

Metric	Only $\mathcal{M}_E$	Only $\mathcal{M}_P$	w/o Dyn. Filter	w/o Activation	Ours (Full)
ATE RMSE (m) ↓	2.15	3.64	1.81	2.32	1.06
PSNR (dB) ↑	21.9	21.2	22.3	22.1	22.7

Table 7: Ablation of Gaussian initialization strategies on KITTI-02.

Strategy	PSNR ↑	#GS (Millions)	VRAM (GB)	Time (min)
Random Sampling	21.84	365	44	120
Ours (Canny instead of DoG)	21.05	128	18	85
Ours (LoG instead of DoG)	21.62	142	20	90
Ours w/o Depth Grad.	21.56	131	19	86
Ours w/o NMS	21.79	202	24	100
Ours (DoG + Depth Grad.)	21.77	135	19	87

4.3 Ablation Study

To validate our system design, we conduct ablation studies on the KITTI dataset, evaluating both tracking and mapping metrics.

Motion-Adaptive Hybrid Tracking. As shown in Table 6, relying on a single geometric solver ($\mathcal{M}_E$ or $\mathcal{M}_P$) yields significant trajectory drift due to environmental degeneracies. Disabling dynamic filtering exacerbates errors induced by moving objects and severe mismatches. Crucially, removing the foundation model activation leaves the system vulnerable to extreme visual degradation, substantially increasing the ATE. This resilient tracking directly establishes a robust geometric foundation for the mapping, as the full hybrid tracking module directly translates to the highest rendering fidelity (22.7 dB).

Gaussian Initialization. Table 7 evaluates our initialization strategy against alternative operators. Canny, as a single-scale edge detector, inserts insufficient primitives, degrading the PSNR. Compared to the LoG operator, which is highly sensitive to outdoor high-frequency noise, DoG extracts multi-scale textural features more robustly. While uniform random sampling or omitting Non-Maximum Suppression (w/o NMS) yields marginal PSNR gains, they cause prohibitive explosions in primitive count, VRAM (*e.g.*, 44 GB), and runtime. Our combined strategy (DoG + Depth Grad.) demonstrates its superiority by achieving the optimal accuracy-efficiency trade-off.

Lifecycle-Managed Mapping. Table 8 demonstrates the necessity of our multi-view consistent management. Attempting to compute multi-view scores globally without historical chunking (*w/o Chunk*) immediately exhausts GPU memory (OOM) on long sequences. Densification effectively recovers under-reconstructed regions omitted during initialization, significantly boosting the PSNR. Conversely, our pruning mechanism reliably identifies and eliminates redundant floaters. This not only compresses the map substantially (from 166M to 135M Gaussians) but also refines the overall rendering quality, proving the effectiveness of our continuous optimization pipeline.

Table 8: Ablation of multi-view consistent Management on KITTI-02.

Densify	Prune	Chunk	PSNR $\uparrow$	#GS (Millions)	VRAM (GB)
×	×	-	21.27	149	20
×	✓	✓	21.16	108	17
✓	×	✓	21.71	166	21
✓	✓	×	OOM	OOM	OOM
✓	✓	✓	21.77	135	19

5 Conclusion

In this paper, we presented a novel monocular 3DGS-SLAM framework specifically engineered to tackle the complexities of kilometer-scale outdoor environments. By identifying the critical bottlenecks of pure visual tracking and volumetric rendering in unbounded scenes, we developed a system that fundamentally prioritizes both resilience and efficiency. Our motion-adaptive hybrid tracking module effectively mitigates local tracking collapses by intelligently activating dense foundation models only when geometric solvers fail. Concurrently, the proposed lifecycle-managed Gaussian mapping strategy strictly regulates the growth of Gaussian primitives, compressing the scene representation without sacrificing photorealistic fidelity. Comprehensive evaluations confirm that our method significantly outperforms existing baselines in tracking precision, rendering detail, and memory economy, demonstrating exceptional scalability for ultra-long driving sequences. We hope this joint optimization of robust tracking and memory-efficient mapping can provide valuable insights for future research on scaling high-fidelity visual SLAM to large-scale environments.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 62406267), Guangdong Provincial Project (No. 2024QN11X072), Guangzhou-HKUST(GZ) Joint Funding Program (No. 2025A03J3956) and Guangzhou Municipal Education Project (No. 2024312122).

References

1. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. *IEEE transactions on robotics* **37**(6), 1874–1890 (2021)
2. Cheng, C., Chen, X., Xie, T., Yin, W., Ren, W., Zhang, Q., Guo, X., Wang, H.: Longstream: Long-sequence streaming autoregressive visual geometry. *arXiv preprint arXiv:2602.13172* (2026)
3. Cheng, C., Hu, Y., Yu, S., Zhao, B., Wang, Z., Wang, H.: Reggs: Unposed sparse views gaussian splatting with 3dgs registration. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 8100–8109 (2025)

4. Cheng, C., Wang, Z., Yu, S., Hu, Y., Yao, N., Wang, H.: Unposed 3dgs reconstruction with probabilistic procrustes mapping. *arXiv preprint arXiv:2507.18541* (2025)
5. Cheng, C., Yu, S., Wang, Z., Zhou, Y., Wang, H.: Outdoor monocular slam with global scale-consistent 3d gaussian pointmaps. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 26035–26044 (2025)
6. Cheng, J., Cai, Z., Zhang, Z., Yin, W., Muller, M., Paulitsch, M., Yang, X.: Romeo: Robust metric visual odometry. *arXiv preprint arXiv:2412.11530* (2024)
7. Deng, K., Ti, Z., Xu, J., Yang, J., Xie, J.: Vggt-long: Chunk it, loop it, align it—pushing vggt’s limits on kilometer-scale long rgb sequences. *arXiv preprint arXiv:2507.16443* (2025)
8. Deng, K., Zhang, Y., Yang, J., Xie, J.: Gigaslam: Large-scale monocular slam with hierarchical gaussian splats. In: *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*. pp. 1–10 (2025)
9. Deng, T., Wu, W., He, J., Pan, Y., Jiang, X., Yuan, S., Wang, D., Wang, H., Chen, W.: Vpgs-slam: Voxel-based progressive 3d gaussian slam in large-scale scenes. *arXiv preprint arXiv:2505.18992* (2025)
10. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: *European conference on computer vision*. pp. 834–849. Springer (2014)
11. Gálvez-López, D., Tardos, J.D.: Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on robotics* **28**(5), 1188–1197 (2012)
12. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. *The international journal of robotics research* **32**(11), 1231–1237 (2013)
13. Homeyer, C., Begiristain, L., Schnörr, C.: DROID-Splat: Combining end-to-end SLAM with 3D gaussian splatting. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. pp. 2788–2798 (2025)
14. Hong, S., He, J., Zheng, X., Zheng, C.: Liv-gaussmap: Lidar-inertial-visual fusion for real-time 3d radiance field map rendering. *IEEE Robotics and Automation Letters* **9**(11), 9765–9772 (2024)
15. Hu, J., Chen, X., Feng, B., Li, G., Yang, L., Bao, H., Zhang, G., Cui, Z.: Cg-slam: Efficient dense rgb-d slam in a consistent uncertainty-aware 3d gaussian field. In: *European Conference on Computer Vision*. pp. 93–112. Springer (2024)
16. Hu, Y., Cheng, C., Yu, S., Guo, X., Wang, H.: Vggt4d: Mining motion cues in visual geometry transformers for 4d scene reconstruction. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*. pp. 414–424 (June 2026)
17. Huang, H., Li, L., Cheng, H., Yeung, S.K.: 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*. pp. 21584–21593 (2024)
18. Keetha, N., Karhade, J., Jatavallabhula, K.M., Yang, G., Scherer, S., Ramanan, D., Luiten, J.: Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 21357–21366 (2024)
19. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., et al.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* **42**(4), 139:1–139:14 (2023)
20. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: An accurate $O(n)$ solution to the PnP problem. *International journal of computer vision* **81**(2), 155–166 (2009)
21. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: *European conference on computer vision*. pp. 71–91. Springer (2024)

22. Liao, Y., Xie, J., Geiger, A.: Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(3), 3292–3310 (2022)
23. Lipson, L., Teed, Z., Deng, J.: Deep patch visual slam. In: *European Conference on Computer Vision*. pp. 424–440. Springer (2024)
24. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 20654–20664 (2024)
25. Maggio, D., Lim, H., Carlone, L.: Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *arXiv preprint arXiv:2505.12549* (2025)
26. Matsuki, H., Murai, R., Kelly, P.H., Davison, A.J.: Gaussian splatting slam. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 18039–18048 (2024)
27. Meuleman, A., Shah, I., Lanvin, A., Kerbl, B., Drettakis, G.: On-the-fly reconstruction for large-scale novel view synthesis from unposed images. *ACM Transactions on Graphics (TOG)* **44**(4), 1–14 (2025)
28. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics* **33**(5), 1255–1262 (2017)
29. Murai, R., Dexheimer, E., Davison, A.J.: Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 16695–16705 (2025)
30. Nistér, D.: An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence* **26**(6), 756–770 (2004)
31. Piccinelli, L., Sakaridis, C., Yang, Y.H., Segu, M., Li, S., Abbeloos, W., Van Gool, L.: Unidepthv2: Universal monocular metric depth estimation made simpler. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025)
32. Rajamani, R.: *Vehicle dynamics and control*. Springer (2006)
33. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: Sam 2: Segment anything in images and videos. In: *International Conference on Learning Representations*. vol. 2025, pp. 28085–28128 (2025)
34. Ren, S., Wen, T., Fang, Y., Lu, B.: Fastgs: Training 3d gaussian splatting in 100 seconds. *arXiv preprint arXiv:2511.04283* (2025)
35. Sandström, E., Zhang, G., Tateno, K., Oechsle, M., Niemeyer, M., Zhang, Y., Patel, M., Van Gool, L., Oswald, M., Tombari, F.: Splat-slam: Globally optimized rgb-only slam with 3d gaussians. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 1680–1691 (2025)
36. Sun, P., Kretschmar, H., Dotiwala, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 2446–2454 (2020)
37. Teed, Z., Deng, J.: Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems* **34**, 16558–16569 (2021)
38. Teed, Z., Lipson, L., Deng, J.: Deep patch visual odometry. *Advances in Neural Information Processing Systems* **36**, 39033–39051 (2023)
39. Tyszkiewicz, M., Fua, P., Trulls, E.: Disk: Learning local features with policy gradient. *Advances in neural information processing systems* **33**, 14254–14265 (2020)

40. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025)
41. Wang, L., Gong, R., Han, Y., Yang, L., Yang, L., Li, Y., Xu, B., Liu, H., Fu, R.: Towards next-generation slam: A survey on 3dgs-slam focusing on performance, robustness, and future directions. arXiv preprint arXiv:2602.04251 (2026)
42. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20697–20709 (2024)
43. Wu, C., Duan, Y., Zhang, X., Sheng, Y., Ji, J., Zhang, Y.: Mm-gaussian: 3d gaussian-based multi-modal fusion for localization and reconstruction in unbounded scenes. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 12287–12293. IEEE (2024)
44. Wu, K., Zhang, Z., Tie, M., Ai, Z., Gan, Z., Ding, W.: Vings-mono: Visual-inertial gaussian splatting monocular slam in large scenes. IEEE Transactions on Robotics (2025)
45. Yan, C., Qu, D., Xu, D., Zhao, B., Wang, Z., Wang, D., Li, X.: Gs-slam: Dense visual slam with 3d gaussian splatting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19595–19604 (2024)
46. Yu, S., Cheng, C., Zhou, Y., Yang, X., Wang, H.: Rgb-only gaussian splatting slam for unbounded outdoor scenes. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 11068–11074. IEEE (2025)
47. Yugay, V., Li, Y., Gevers, T., Oswald, M.R.: Gaussian-slam: Photo-realistic dense slam with gaussian splatting. arXiv preprint arXiv:2312.10070 (2023)
48. Zhan, H., Weerasekera, C.S., Bian, J.W., Garg, R., Reid, I.: Df-vo: What should be learnt for visual odometry? arXiv preprint arXiv:2103.00933 (2021)
49. Zhao, B., Yu, S., Yin, Z., Shen, D., Wang, H.: Mmgs: 10× compressed 3dgs through optimal transport aggregation based on multi-view ranking. arXiv preprint arXiv:2605.19304 (2026)
50. Zhao, B., Zhou, Y., Song, G., Yin, Z., Wang, H.: 3d skew gaussian splatting with any camera trajectory visualization engine. arXiv preprint arXiv:2605.18334 (2026)
51. Zhao, B., Zhou, Y., Yu, S., Wang, Z., Wang, H.: Wavelet-gs: 3d gaussian splatting with wavelet decomposition. In: Proceedings of the 33rd ACM International Conference on Multimedia. pp. 8616–8625 (2025)
52. Zhu, P., Zhuang, Y., Chen, B., Li, L., Wu, C., Liu, Z.: Mgs-slam: Monocular sparse tracking and gaussian mapping with depth smooth regularization. IEEE Robotics and Automation Letters **9**(11), 9486–9493 (2024)