MASt3R-GS: Bridging 3D Reconstruction Priors with Gaussian Splatting for Real-Time Dense SLAM

Qingze Wang
Independent Researcher
Tianjin, China
oijg47@outlook.com

***Abstract*— 3D Gaussian Splatting (3DGS) has emerged as a powerful tool for geometry and appearance representation in RGB-only dense Simultaneous Localization and Mapping (SLAM), as it provides a compact dense map representation while enabling efficient and high-quality map rendering. However, existing 3DGS-based SLAM methods suffer from poor global consistency performance. To address this limitation, we propose MASt3R-GS, a novel system architecture that generates 3D dense Gaussian map representations using only RGB images as input. Our system leverages the robust sparse point clouds and pose priors from MASt3R-SLAM, a monocular dense SLAM system, combined with the local optimization capabilities of 3D Gaussian Splatting to achieve robust localization and high-fidelity reconstruction for downstream applications. Current experiments on the TUM-RGBD dataset demonstrate the effectiveness of our globally optimized 3D Gaussians, achieving superior rendering quality while maintaining competitive tracking accuracy. Our approach successfully bridges the gap between global consistency and local reconstruction quality in RGB-only dense SLAM systems.**

***Index Terms*— SLAM, 3D Gaussian Splatting, RGB-only mapping, global consistency, dense reconstruction**

I. Introduction

Real-time dense Simultaneous Localization and Mapping (SLAM) is a fundamental capability for autonomous systems, enabling camera tracking and 3D reconstruction [1] [2] in unknown environments. Recently, 3D Gaussian Splatting (3DGS) [3] has emerged as a promising alternative to Neural Radiance Fields (NeRF) [4], providing a flexible point-based scene representation through 3D Gaussian distributions and demonstrating remarkable potential in photorealistic rendering and efficient reconstruction. SLAM methods [5] based on 3DGS have recently shown progress in constructing more accurate and dense 3D scene maps.

However, existing 3D Gaussian Splatting-based SLAM systems suffer from critical limitations. To obtain online camera poses for merging local 3D Gaussians, these methods [5] [6] [7] typically integrate pose optimization into the overall optimization process, lacking independent global trajectory and map optimization mechanisms. This tightly-coupled design not only increases the computational burden of the system but, more importantly, leads to accumulation of pose errors and geometric distortion of maps, particularly when processing long sequences or scenes with loop closures.

Meanwhile, recent advances in two-view 3D reconstruction priors [8] have achieved breakthrough progress. MASt3R-SLAM [9] demonstrates how to leverage the MASt3R [10] two-view reconstruction prior to build a plug-and-play monocular SLAM system capable of producing globally consistent poses and dense geometry. Through efficient pointmap matching, robust tracking mechanisms, and second-order global optimization, the system achieves accurate camera localization and consistent scene reconstruction.

In this work, we present MASt3R-GS, a novel framework that combines the robust localization capabilities of MASt3R-SLAM with the high-quality rendering capabilities of 3D Gaussian Splatting. Our key insight is that by employing MASt3R-SLAM as an independent localization module, we can obtain accurate and globally consistent camera poses, thereby providing a reliable geometric foundation for 3D Gaussian Splatting. This decoupled design offers multiple advantages: First, the robust pose estimation and global optimization capabilities provided by MASt3R-SLAM effectively prevent pose error accumulation, offering accurate geometric constraints for 3D Gaussian initialization and optimization. Based on this reliable pose prior, 3D Gaussian Splatting can focus on optimizing scene representation without falling into local optima. Second, the dense pointmap output from MASt3R-SLAM provides high-quality initialization for 3D Gaussians, significantly improving convergence speed and reconstruction quality. The system architecture is shown in Figure 1,

Our main contributions are:

- We propose the MASt3R-GS framework, successfully combining the robust localization capabilities of MASt3R-SLAM with the efficient rendering techniques of 3D Gaussian Splatting, achieving accurate camera tracking and high-quality scene reconstruction.
- Through systematic fusion design, we address the issues of pose error accumulation and map distortion present in existing 3DGS-based SLAM methods.
- We experimentally validate the effectiveness of our fusion framework in terms of trajectory accuracy and reconstruction quality on standard benchmark datasets.

II. Related Work

Dense Visual SLAM. Visual SLAM plays a crucial role in robot navigation and augmented reality. Traditional feature-

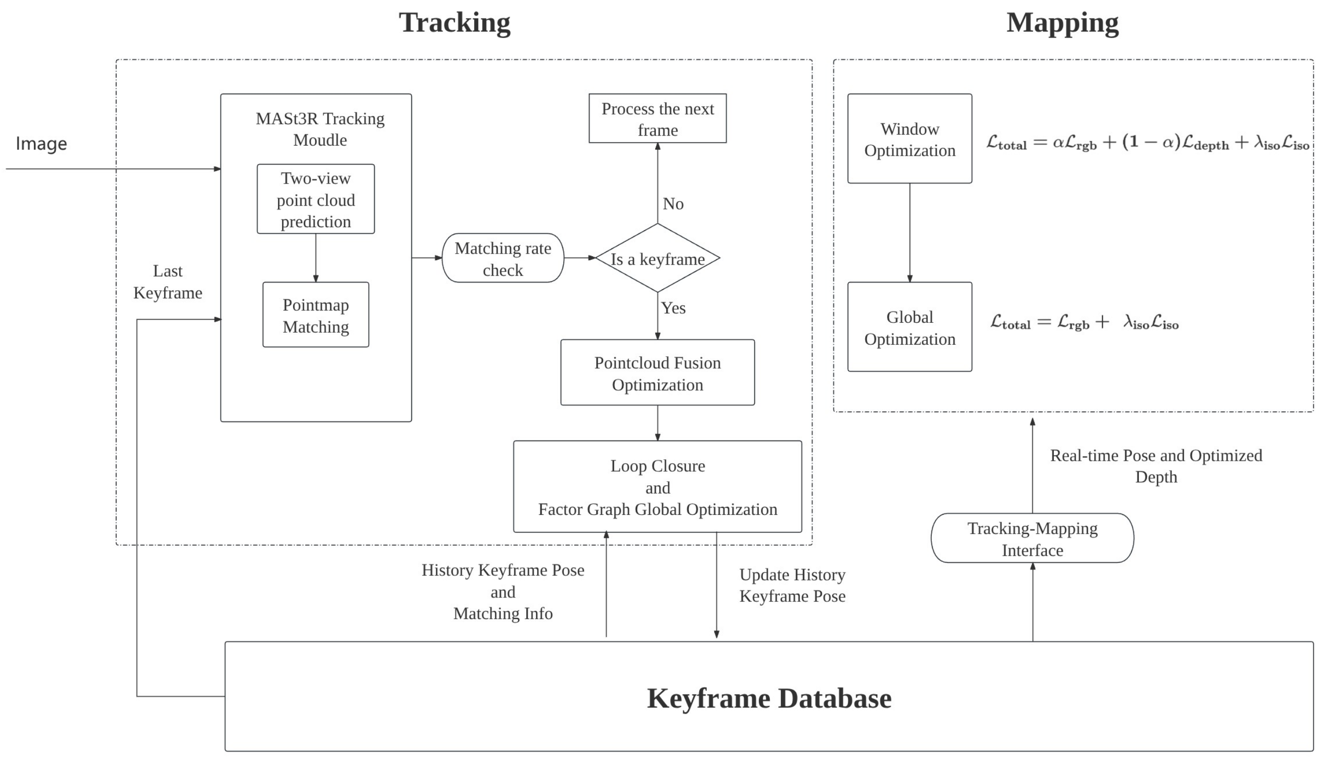
Tracking
Mapping
Image
MASt3R Tracking Moudle
Two-view point cloud prediction
Pointmap Matching
Last Keyframe
Matching rate check
Is a keyframe
No
Yes
Process the next frame
Pointcloud Fusion Optimization
Loop Closure and Factor Graph Global Optimization
Window Optimization
$\mathcal{L}_{\mathbf{total}} = \alpha\mathcal{L}_{\mathbf{rgb}} + (1-\alpha)\mathcal{L}_{\mathbf{depth}} + \lambda_{\mathbf{iso}}\mathcal{L}_{\mathbf{iso}}$
Global Optimization
$\mathcal{L}_{\mathbf{total}} = \mathcal{L}_{\mathbf{rgb}} + \lambda_{\mathbf{iso}}\mathcal{L}_{\mathbf{iso}}$
Real-time Pose and Optimized Depth
Tracking-Mapping Interface
History Keyframe Pose and Matching Info
Update History Keyframe Pose
Keyframe Database

Fig. 1: This SLAM system adopts a modular architecture design, primarily consisting of two core modules: Tracking and Mapping, with data sharing and coordination between modules achieved through a Keyframe Database. The Tracking module is responsible for real-time processing of input images, performing two-view point cloud prediction through the MASt3R tracking module, combined with Pointmap Matching technology to check matching rates. When a keyframe is detected, it triggers Pointcloud Fusion Optimization. The poses of all keyframes are maintained in real-time through Loop Closure and Factor Graph Global Optimization, ensuring global consistency of the trajectory. The Mapping module employs a sliding window optimization strategy, first performing local window optimization for rapid processing of local map information, followed by global color optimization to ensure visual consistency of the overall map. The two modules exchange Pose and Depth information in real-time through the Tracking-Mapping Interface.

based methods such as the ORB-SLAM series [1], [11], [12] achieve robust localization through sparse features but struggle to provide dense geometric information. Direct methods like DSO [13] and LSD-SLAM [14] achieve dense mapping through photometric optimization, while KinectFusion [15] pioneered real-time RGB-D dense reconstruction. Recent neural implicit methods have transformed the dense reconstruction landscape—iMAP [16] pioneered the integration of neural implicit representations into SLAM systems. Subsequently, NICE-SLAM [17] and Vox-Fusion [18] enhanced efficiency through hierarchical scene representations. In the domain of traditional dense SLAM, DROID-SLAM [19] leverages deep learning in conjunction with differentiable dense bundle adjustment to achieve state-of-the-art tracking accuracy. Nevertheless, methods based on neural implicit representations continue to face significant challenges in terms of real-time performance, memory efficiency, and global consistency maintenance.

3D Gaussian Splatting Representation. 3D Gaussian Splatting [3] serves as an explicit scene representation, achieving rendering speeds orders of magnitude faster than NeRF [4] through differentiable rasterization. This method uses anisotropic 3D Gaussians as scene primitives, with each Gaussian containing position, covariance, opacity, and appearance parameters. In SLAM applications, SplaTAM [7] and Gaussian-SLAM [6] pioneered the integration of 3D Gaussians into real-time systems, while MonoGS [5] specifically addresses monocular initialization challenges. Photo-SLAM [20] combines multi-resolution hash encoding to enhance reconstruction accuracy, and GS-SLAM [21] improves Gaussian pruning strategies. Despite significant progress, existing methods still face challenges in robust initialization under monocular settings and memory management for large-scale scenes.

Two-view Reconstruction Priors. The ability to predict 3D structure from image pairs forms the foundation of visual SLAM. Traditional methods rely on fundamental matrix estimation and triangulation [22], with COLMAP [23] achieving incremental reconstruction through robust feature matching. Deep learning has significantly improved matching quality—SuperGlue [24] uses graph neural networks for learned correspondences, while LoFTR [25] employs Transformers for dense matching without detectors. Revolutionary advances come from direct 3D prediction methods: DUSt3R [8] learns to regress pointmaps in a unified 3D space from image pairs without explicit calibration; MASt3R [10] enhances this with pixel-level matching for improved geometric consistency; Spann3R [26] extends to sequential prediction through fine-tuning. Recently, VGGT [27] achieves robust long-sequence reconstruction through geometry-guided Transformers with global optimization. While these methods provide calibration-free geometric initialization, maintaining global consistency in extended sequences remains challenging.

Global Optimization and Loop Closure. Maintaining global consistency represents a core challenge in SLAM systems. Pose graph optimization minimizes loop closure constraints through frameworks like g2o [28], while Bundle Adjustment [29] jointly optimizes cameras and scene structure. For loop detection, DBoW2 [30] uses bag-of-words models, while NetVLAD [31] leverages deep learning for robust place recognition. Global optimization of dense representations poses greater challenges—LoopSplat [32] achieves efficient loop closure and global optimization through submap partitioning and 3DGS registration, avoiding expensive reintegration. GO-SLAM [33] and HI-SLAM [34] adopt hierarchical strategies to balance accuracy and efficiency. However, efficiently propagating global corrections to dense neural representations under real-time constraints remains an open problem.

III. Methodology

Pose estimation relies on the MASt3R-SLAM framework, which is characterized by maintaining global trajectory consistency. In addition to providing real-time camera poses, MASt3R-SLAM generates optimized 3D sparse point clouds as geometric priors. Its factor graph optimization mechanism retrospectively refines historical keyframe poses, effectively suppressing cumulative errors.

Scene reconstruction is achieved through 3D Gaussian primitives. During system operation, the Gaussian collection undergoes incremental updates based on new viewpoint observations, enabling progressive refinement of scene geometry. This dynamic growth mechanism allows the system to flexibly handle scenes of varying scales and complexities.

A. Scene Representation

Our SLAM is represented in a 3DGS method which combines multiple channels. Every gaussian is defined

$$G_i(x|\mu_i,\Sigma_i) = e^{-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)} \tag{1}$$

where $\mu_i \in \mathbb{R}^3$ and $\Sigma_i \in \mathbb{R}^{3\times3}$ are the center of a point $p_i \in P$ and the corresponding 3D covariance matrix, respectively. The covariance matrix Σ_i can be decomposed into a scaling matrix $S_i \in \mathbb{R}^{3\times3}$ and a rotation matrix $R_i \in \mathbb{R}^{3\times3}$ as follows:

$$\Sigma_i = R_i S_i S_i^T R_i^T \mathbf{S}_i = \mathrm{diag}(s_{ix}, s_{iy}, s_{iz}) \tag{2}$$

The parameters of Gaussians are optimized during the mapping process. We map the Gaussians onto 2D images in a differentiable way along the depth dimension in pixel coordinates. We use fast α-blending for rendering. The parameters μ_i and Σ_i under the camera coordinate system could be projected onto 2D coordinates as follows:

$$\mu_i' = KW[\mu_i, 1]^T, \quad \Sigma_i' = JW\Sigma_i W^T J^T \tag{3}$$

where K is the intrinsic matrix of camera. W is a transformation matrix. J denotes the Jacobian matrix of the projective transformation. The pixel-level rendered color $\mathbf{C} \in \mathbb{R}^3$ could be obtained in a manner of α-blending:

$$\mathbf{C} = \sum_{i\in N_r} c_i\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j) \tag{4}$$

where $\mathbf{c}_i$ is represented by spherical harmonics and is view-dependent. During rasterization, the contributions of α_i are decayed using a Gaussian function, based on the 2D Gaussian formed by splatting a 3D Gaussian. N_r is the number of Gaussians that the ray passes through. Similarly, the depth can be rendered as:

$$\mathbf{D} = \sum_{i\in N_r} d_i\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j) \tag{5}$$

where $\mathbf{d}_i$ denotes the depth of each Gaussian. In addition to rendering its color and depth, we also need to render other finer geometric details like normals to accurately represent the surface of an actual scene.

B. Tracking and Mapping

1) Tracking: We employ MASt3R-SLAM in calibrated mode as our front-end tracking module, which is a real-time monocular dense SLAM system built upon the MASt3R two-view 3D reconstruction prior. For each RGB keyframe pair (I_i, I_j), MASt3R directly predicts pointmaps $\mathbf{X}_i^i, \mathbf{X}_j^i \in \mathbb{R}^{H\times W\times 3}$ in a common coordinate frame, along with per-pixel features $\mathbf{D}_i^i, \mathbf{D}_j^i \in \mathbb{R}^{H\times W\times d}$ for matching. The system employs efficient iterative projective matching to establish pixel correspondences $m_{i,j}$ between frames. Given the pointmap outputs from MASt3R, each point $\mathbf{x} \in \mathbf{X}_j^i$ is projected to frame I_i by iteratively optimizing pixel coordinates $\mathbf{p}$ to minimize the ray error:

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} \|\psi([\mathbf{X}_i^i]_{\mathbf{p}}) - \psi(\mathbf{x})\|^2 \tag{6}$$

where $\psi(\cdot)$ normalizes pointmaps into unit rays, enabling the system to handle generic camera models without parametric assumptions. Camera poses are refined through Gauss-Newton optimization to minimize the directional ray error:

$$E_r = \sum_{m,n\in m_{f,k}} \left\| \frac{\psi(\tilde{\mathbf{X}}_k^{k,n}) - \psi(\mathbf{T}_{kf}\mathbf{X}_f^{f,m})}{w(q_{m,n}, \sigma_r^2)} \right\|_\rho \tag{7}$$

where $q_{m,n} = \sqrt{Q_f^{f,m} Q_k^{f,n}}$ represents the match confidence weighting, $w(\cdot)$ is a per-match weighting function, and $\|\cdot\|_\rho$ denotes the Huber norm for robustness. A new keyframe K_i is added when the number of valid matches or the number of unique keyframe pixels in $m_{f,k}$ falls below a threshold ω_k. Upon adding K_i, a bidirectional edge to the previous keyframe K_{i-1} is added to the edge list $\mathscr{E}$, constraining the estimated poses sequentially in time. After solving for the relative pose $\mathbf{T}_{kf}$, the canonical pointmap $\tilde{\mathbf{X}}_k^k$ is updated via a confidence-weighted averaging filter:

$$\tilde{\mathbf{X}}_k^k \leftarrow \frac{\tilde{C}_k^k \tilde{\mathbf{X}}_k^k + C_k^f (\mathbf{T}_{kf} \mathbf{X}_k^f)}{\tilde{C}_k^k + C_k^f}, \quad \tilde{C}_k^k \leftarrow \tilde{C}_k^k + C_k^f \tag{8}$$

where C represents the confidence values. This filtering process merges information from multiple viewpoints, progressively refining the scene geometry. The system maintains global consistency through factor graph optimization. Loop closure detection is performed using encoded MASt3R features with an Aggregated Selective Match Kernel (ASMK) framework. When retrieval scores exceed threshold ω_r and the number of matches exceeds ω_l, bidirectional edges are added to the graph. Second-order global optimization is then performed over all keyframe poses to achieve global consistency:

$$E_g = \sum_{i,j \in \mathscr{E}} \sum_{m,n \in m_{i,j}} \left\| \frac{\psi(\tilde{\mathbf{X}}_i^{i,m}) - \psi(\mathbf{T}_{ij} \tilde{\mathbf{X}}_j^{j,n})}{w(q_{m,n}, \sigma_r^2)} \right\|_\rho \tag{9}$$

where $\mathbf{T}_{ij} = \mathbf{T}_{WC_i}^{-1} \mathbf{T}_{WC_j}$ represents the relative transformation between keyframes. The tracking module operates at approximately 15 FPS, providing robust camera pose estimation and dense geometric priors for subsequent 3D Gaussian Splatting optimization.

2) Tracking-Mapping Interface: The interface between MASt3R-SLAM and 3D Gaussian Splatting handles two tasks: depth map generation and pose synchronization.

a) Depth Map Generation: MASt3R-SLAM outputs point clouds with per-point confidence values. We project these points to image space considering camera distortion parameters $\mathbf{k} = [k_1, k_2, p_1, p_2, k_3]^T$ (radial: k_1, k_2, k_3; tangential: p_1, p_2), apply Z-buffering for occlusion handling, and filter low-confidence regions (threshold: 3.0) to generate depth maps for Gaussian initialization.

b) Pose Synchronization: During global optimization and loop closure, the frontend updates historical keyframe poses. The backend synchronizes these updates through a callback mechanism, ensuring consistent poses between tracking and mapping modules without interrupting ongoing operations.

C. Mapping

Following the mapping framework of [5], our mapping process is divided into two distinct stages: sliding window mapping and color refinement, leveraging the globally optimized poses from the frontend to guide the final map refinement.

a) Coarse Mapping Stage: The coarse mapping operates synchronously with the tracking module to maintain a coherent 3D structure. During this stage, we utilize the most recent keyframe poses provided by the frontend tracker while progressively inserting and optimizing newly added Gaussian distributions. The optimization is performed over keyframes within a sliding window W_k to reconstruct currently visible regions. To preserve global map consistency, we additionally sample two random historical keyframes W_r at each iteration. Importantly, this stage exclusively optimizes exposure parameters $\{a_i, b_i\}$, where a_i and b_i represent the affine transformation coefficients for exposure compensation of frame i (i.e., $I_{corrected} = a_i \cdot I_{original} + b_i$), while keeping camera poses fixed, thereby avoiding conflicts with the tracking module.

b) Fine Mapping Stage: After all frames have been processed by the frontend tracker and optimized through factor graph global optimization, we execute color refinement. This stage employs high-iteration optimization to refine the rendering quality of global keyframes.

1) Map Optimization with Depth and Geometric Regularization: The Gaussian map is optimized by minimizing the discrepancy between the rendered image $\hat{I}_i$ and the original image I_i, as well as between the rendered depth $\hat{D}_i$ and the depth D_i generated from MASt3R point clouds. Considering the uncertainty in point cloud conversion, we perform pixel-wise filtering based on depth confidence. Additionally, we introduce an isotropic regularization term to constrain the shape of Gaussians. The objective function is defined as follows:

$$L_{rgb} = \|\hat{I}_i \odot M_{rgb} - I_i \odot M_{rgb}\|_1 \tag{10}$$

$$L_{depth} = \|\hat{D}_i \odot M_{depth} - D_i \odot M_{depth}\|_1 \tag{11}$$

$$L_{iso} = \sum_{g \in \mathscr{G}} \sum_{i \in \{x,y,z\}} (s_{g,i} - \bar{s}_g)^2 \tag{12}$$

$$L_{total} = \alpha \cdot L_{rgb} + (1-\alpha) \cdot L_{depth} + \lambda_{iso} \cdot L_{iso} \tag{13}$$

where $\odot$ denotes element-wise multiplication, $M_{rgb}, M_{depth} \in \{0,1\}^{H \times W}$ are binary masks, with M_{rgb} being the RGB boundary mask and M_{depth} the confidence-based depth mask derived from MASt3R point cloud confidence. D_i is the depth map projected from MASt3R point clouds, $s_{g,i}$ denotes the scaling parameters of Gaussian g along axis i, and $\bar{s}_g = \frac{1}{3}\sum_i s_{g,i}$ is their mean value. We empirically set $\alpha = 0.95$ to prioritize photometric consistency and $\lambda_{iso} = 10$ to enforce shape regularization. The depth mask M_{depth} is activated only at pixels where depth values are non-zero and point cloud confidence exceeds the threshold $\tau_c = 3.0$, ensuring that only high-quality geometric priors contribute to the optimization. The isotropic loss L_{iso} encourages Gaussians to maintain more regular shapes by minimizing the variance among the three axial scales, contributing to a smoother and more coherent scene representation.

IV. Experiment

We currently evaluate the proposed system only on the TUM-RGBD benchmark dataset to validate its effectiveness and generalization capability. We present qualitative results from real-time monocular inputs, demonstrating the system's robustness and capability for high-quality reconstruction in practical scenarios. Finally, we conduct an ablation study to analyze the impact of different system design combinations on accuracy.

a) Implementation Details: We deploy our system on a workstation powered by an AMD Ryzen 9 9950X (4.30GHz) CPU and a single NVIDIA RTX 4090 GPU. Core computations such as rasterization and gradient evaluation are accelerated with CUDA, while high-level logic is built using PyTorch.

b) Metrics: We evaluate tracking accuracy using the RMSE of Absolute Trajectory Error (ATE), while reconstruction quality is measured via standard photometric metrics including PSNR, SSIM, and LPIPS.

c) Baseline Methods: We compare our approach against a range of recent monocular SLAM systems and Gaussian-based methods, including NICE-SLAM, SplaTam, MonoGS, MonoGS++ and Splat-SLAM. These baselines cover both traditional and neural field-based designs, providing a comprehensive reference for evaluating tracking robustness and reconstruction fidelity.

A. Quantitative Evaluation

Table 1 reveals distinct performance characteristics of our approach. By decoupling tracking and mapping, our system achieves the best rendering metrics (PSNR: 26.68 dB, SSIM: 0.86) among all RGB-only methods, with a modest trade-off in tracking accuracy (ATE: 2.92 cm). Note that MonoGS without depth input shows significant performance degradation, highlighting the importance of robust pose priors in RGB-only settings. This 0.83 cm difference in ATE compared to Splat-SLAM is offset by a significant 0.83 dB improvement in PSNR—a favorable exchange for applications prioritizing visual fidelity. The consistent PSNR improvements across all test scenes demonstrate that our global optimization strategy effectively enhances reconstruction quality without severely compromising localization performance.

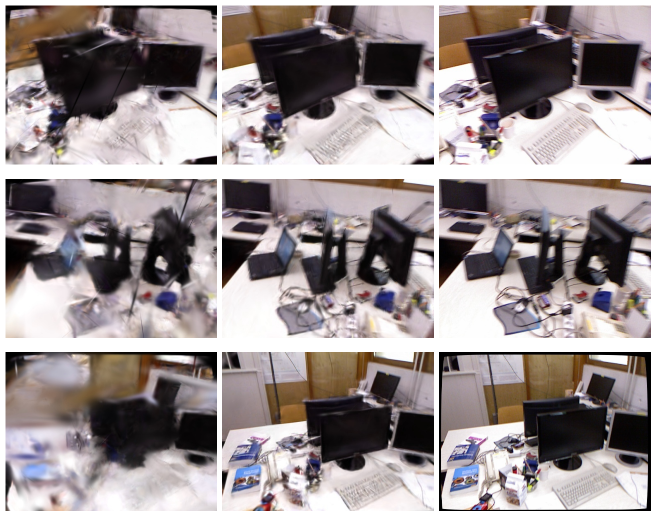

Fig. 2: Rendering results on the TUM-RGBD dataset.

Method	**Modality**	**Metric**	fr1/desk	fr1/desk2	fr1/room	fr2/xyz	fr3/office	**Avg.**
NICE-SLAM [17]	RGB-D	PSNR [dB]↑	13.83	12.00	11.75	17.64	12.02	13.45
		SSIM↑	0.56	0.50	0.37	0.71	0.54	0.54
		LPIPS↓	0.48	0.52	0.62	0.34	0.49	0.49
		ATE [cm]↓	4.46	5.12	32.17	31.85	3.66	15.45
SplaTAM [7]	RGB-D	PSNR [dB]↑	22.41	19.58	19.02	23.11	19.92	20.81
		SSIM↑	0.85	0.81	0.79	0.87	0.82	0.83
		LPIPS↓	0.24	0.33	0.31	0.20	0.34	0.28
		ATE [cm]↓	3.35	6.60	11.95	1.24	3.37	5.30
MonoGS [5]	RGB-D	PSNR [dB]↑	23.60	20.31	20.78	24.52	$\underline{24.90}$	22.82
		SSIM↑	0.78	0.70	0.70	0.79	0.83	0.76
		LPIPS↓	0.25	0.35	0.34	$\underline{0.22}$	$\underline{0.22}$	0.28
		ATE [cm]↓	$\underline{1.44}$	$\underline{0.51}$	4.97	1.38	1.31	1.92
MonoGS [5]	RGB	PSNR [dB]↑	14.90	14.31	14.60	22.19	22.32	17.66
		SSIM↑	0.52	0.51	0.51	0.72	0.77	0.61
		LPIPS↓	0.58	0.62	0.62	0.29	0.34	0.49
		ATE [cm]↓	60.70	75.62	72.63	5.06	4.26	35.66
MonoGS++ [35]	RGB	PSNR [dB]↑	22.85	20.64	22.16	26.52	25.08	23.45
		SSIM↑	0.82	0.77	0.77	0.86	0.85	0.81
		LPIPS↓	0.20	0.29	0.31	0.13	0.18	0.22
		ATE [cm]↓	1.79	5.18	21.44	$\underline{0.38}$	$\underline{0.36}$	5.83
Splat-SLAM [36]	RGB	PSNR [dB]↑	25.61	23.98	24.06	$\underline{29.53}$	26.05	25.85
		SSIM↑	0.84	0.81	0.80	$\underline{0.90}$	0.84	0.84
		LPIPS↓	0.18	0.23	0.24	$\underline{0.08}$	0.20	0.19
		ATE [cm]↓	$\underline{1.62}$	2.84	$\underline{4.31}$	0.22	1.46	$\underline{2.09}$
Ours	RGB	PSNR [dB]↑	$\underline{28.06}$	$\underline{26.13}$	$\underline{25.62}$	28.55	25.02	$\underline{26.68}$
		SSIM↑	$\underline{0.88}$	$\underline{0.85}$	$\underline{0.83}$	$\underline{0.90}$	$\underline{0.84}$	$\underline{0.86}$
		LPIPS↓	$\underline{0.16}$	$\underline{0.21}$	$\underline{0.23}$	0.12	$\underline{0.22}$	$\underline{0.19}$
		ATE [cm]↓	1.64	$\underline{2.70}$	6.26	0.63	3.38	2.92

TABLE 1: Rendering and Tracking Results on TUM-RGBD for both RGB-D Methods and RGB Methods. Our method achieves the best rendering quality with competitive tracking accuracy.

B. Ablation Study

To validate whether our system achieves the optimal combination approach and can effectively integrate global consistency priors with the local optimization of Gaussian splatting, we conducted two sets of experiments.

The first set focuses on whether global consistency priors from the front-end can better guide back-end mapping. We designed two experiments:

- **(A1)** Without using the dynamic camera pose strategy, not leveraging the frontend factor graph to maintain global pose consistency, leaving the optimization entirely to the mapping process;
- **(A2)** Leveraging the frontend factor graph to maintain global pose consistency, with the backend mapping always using the latest poses provided by the frontend.

	fr1/desk		fr1/desk2		fr1/room	
Metric	A1	A2	A1	A2	A1	A2
PSNR↑	28.56	$\underline{28.06}$	25.53	$\underline{26.13}$	24.66	$\underline{25.62}$
SSIM↑	$\underline{0.89}$	0.88	0.85	0.85	0.81	$\underline{0.83}$
LPIPS↓	$\underline{0.16}$	$\underline{0.16}$	$\underline{0.23}$	0.21	0.23	0.23
ATE [cm]↓	1.86	$\underline{1.64}$	3.14	$\underline{2.70}$	8.12	$\underline{6.26}$

TABLE 2: Comparison of global consistency prior strategies. Best results are highlighted

The second set of experiments focuses on how the continuous changes in historical keyframe poses during frontend factor graph optimization affect the system, and how to organize the system for optimal results. We simultaneously optimize poses and exposure within the optimization sliding window, but handle pose changes with the following approaches:

- **(B1)** After the Tracking completes global pose consistency maintenance, we perform global Bundle Adjustment using all keyframes, simultaneously optimizing poses and exposure;

- **(B2)** After the Tracking completes global pose consistency maintenance, we repeat the local Bundle Adjustment process from the incremental mapping stage with lower iterations and stricter learning strategies, simultaneously optimizing poses and exposure;
- **(B3)** Ignoring pose changes caused by the factor graph.

Metric	B1	B2	B3
PSNR [dB]↑	28.44	27.39	28.06
SSIM↑	0.88	0.88	0.88
LPIPS↓	0.18	0.17	0.16
ATE [cm]↓	6.70	1.88	1.64
Total Time [s]↓	167.31	153.89	140.51

TABLE 3: Comparison of Pose Change Handling Strategies

Both sets of experiments demonstrate that our current system design achieves a balance between pose estimation accuracy and rendering quality, while also offering advantages in execution time.

V. Conclusion and Future Work

We presented MASt3R-GS, an RGB-only dense SLAM framework that combines MASt3R-SLAM's global optimization with 3D Gaussian Splatting's rendering capabilities. Our decoupled architecture achieves robust pose estimation while delivering the highest rendering quality (PSNR: 26.68 dB, SSIM: 0.86) among RGB-only methods on the TUM-RGBD dataset, maintaining competitive tracking accuracy (ATE: 2.92 cm).

However, our current implementation has several limitations. Evaluation is restricted to the TUM-RGBD dataset, with performance on large-scale outdoor scenes remaining unexplored. Additionally, tracking accuracy slightly lags behind state-of-the-art methods like Splat-SLAM, and both our keyframe selection strategy and mapping approach are relatively basic.

Despite these limitations, our work demonstrates the potential of decoupled architectures in RGB-only SLAM, where global consistency and local rendering quality can be jointly optimized. Looking forward, several promising research directions emerge. First, comprehensive evaluation on other datasets (e.g., ScanNet++, EuRoC) would validate the scalability of our approach. Second, comparing with concurrent methods leveraging similar geometric priors would provide insights into different architectural choices. Third, investigating hybrid approaches that selectively enable pose refinement during mapping could potentially combine the benefits of both decoupled and tightly-coupled designs. Finally, exploring the integration of our framework with emerging foundation models for 3D vision represents an exciting frontier for robust SLAM systems.

References

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, p. 1147–1163, Oct. 2015. [Online]. Available: http://dx.doi.org/10.1109/TRO.2015.2463671

[2] E. Sandström, Y. Li, L. V. Gool, and M. R. Oswald, "Point-slam: Dense neural point cloud-based slam," 2023. [Online]. Available: https://arxiv.org/abs/2304.04278

[3] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," 2023. [Online]. Available: https://arxiv.org/abs/2308.04079

[4] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," 2020. [Online]. Available: https://arxiv.org/abs/2003.08934

[5] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, "Gaussian splatting slam," 2024. [Online]. Available: https://arxiv.org/abs/2312.06741

[6] V. Yugay, Y. Li, T. Gevers, and M. R. Oswald, "Gaussian-slam: Photo-realistic dense slam with gaussian splatting," 2024. [Online]. Available: https://arxiv.org/abs/2312.10070

[7] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten, "Splatam: Splat, track map 3d gaussians for dense rgb-d slam," 2024. [Online]. Available: https://arxiv.org/abs/2312.02126

[8] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," 2024. [Online]. Available: https://arxiv.org/abs/2312.14132

[9] R. Murai, E. Dexheimer, and A. J. Davison, "Mast3r-slam: Real-time dense slam with 3d reconstruction priors," 2025. [Online]. Available: https://arxiv.org/abs/2412.12392

[10] V. Leroy, Y. Cabon, and J. Revaud, "Grounding image matching in 3d with mast3r," 2024. [Online]. Available: https://arxiv.org/abs/2406.09756

[11] R. Mur-Artal and J. D. Tardos, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, p. 1255–1262, Oct. 2017. [Online]. Available: http://dx.doi.org/10.1109/TRO.2017.2705103

[12] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, p. 1874–1890, Dec. 2021. [Online]. Available: http://dx.doi.org/10.1109/TRO.2021.3075644

[13] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," 2016. [Online]. Available: https://arxiv.org/abs/1607.02565

[14] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 834–849.

[15] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 127–136.

[16] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," 2021. [Online]. Available: https://arxiv.org/abs/2103.12352

[17] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," 2022. [Online]. Available: https://arxiv.org/abs/2112.12130

[18] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang, "Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation," in *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, Oct. 2022, p. 499–507. [Online]. Available: http://dx.doi.org/10.1109/ISMAR55827.2022.00066

[19] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," 2022. [Online]. Available: https://arxiv.org/abs/2108.10869

[20] H. Huang, L. Li, H. Cheng, and S.-K. Yeung, "Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular, stereo, and rgb-d cameras," 2024. [Online]. Available: https://arxiv.org/abs/2311.16728

[21] C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," 2024. [Online]. Available: https://arxiv.org/abs/2311.11700

[22] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[23] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[24] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," 2020. [Online]. Available: https://arxiv.org/abs/1911.11763

[25] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers," 2021. [Online]. Available: https://arxiv.org/abs/2104.00680

[26] H. Wang and L. Agapito, "3d reconstruction with spatial memory," 2024. [Online]. Available: https://arxiv.org/abs/2408.16061

[27] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "Vggt: Visual geometry grounded transformer," 2025. [Online]. Available: https://arxiv.org/abs/2503.11651

[28] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3607–3613.

[29] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment — a modern synthesis," in *Vision Algorithms: Theory and Practice*, B. Triggs, A. Zisserman, and R. Szeliski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 298–372.

[30] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[31] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," 2016. [Online]. Available: https://arxiv.org/abs/1511.07247

[32] L. Zhu, Y. Li, E. Sandström, S. Huang, K. Schindler, and I. Armeni, "Loopsplat: Loop closure by registering 3d gaussian splats," 2024. [Online]. Available: https://arxiv.org/abs/2408.10154

[33] Y. Zhang, F. Tosi, S. Mattoccia, and M. Poggi, "Go-slam: Global optimization for consistent 3d instant reconstruction," 2023. [Online]. Available: https://arxiv.org/abs/2309.02436

[34] W. Zhang, T. Sun, S. Wang, Q. Cheng, and N. Haala, "Hi-slam: Monocular real-time dense mapping with hybrid implicit fields," 2023. [Online]. Available: https://arxiv.org/abs/2310.04787

[35] R. Li, W. Ke, D. Li, L. Tian, and E. Barsoum, "Monogs++: Fast and accurate monocular rgb gaussian slam," 2025. [Online]. Available: https://arxiv.org/abs/2504.02437

[36] E. Sandström, K. Tateno, M. Oechsle, M. Niemeyer, L. V. Gool, M. R. Oswald, and F. Tombari, "Splat-slam: Globally optimized rgb-only slam with 3d gaussians," 2024. [Online]. Available: https://arxiv.org/abs/2405.16544