Monocular Online Reconstruction with Enhanced Detail Preservation

SONGYIN WU, Meta Reality Labs Research, USA and University of California Santa Barbara, USA
ZHAOYANG LV, Meta Reality Labs Research, USA
YUFENG ZHU, Meta Reality Labs Research, USA
DUNCAN FROST, Meta Reality Labs Research, United Kingdom
ZHENGQIN LI, Meta Reality Labs Research, USA
LING-QI YAN, University of California Santa Barbara, USA
CARL REN, Meta Reality Labs Research, USA
RICHARD NEWCOMBE, Meta Reality Labs Research, USA
ZHAO DONG, Meta Reality Labs Research, USA

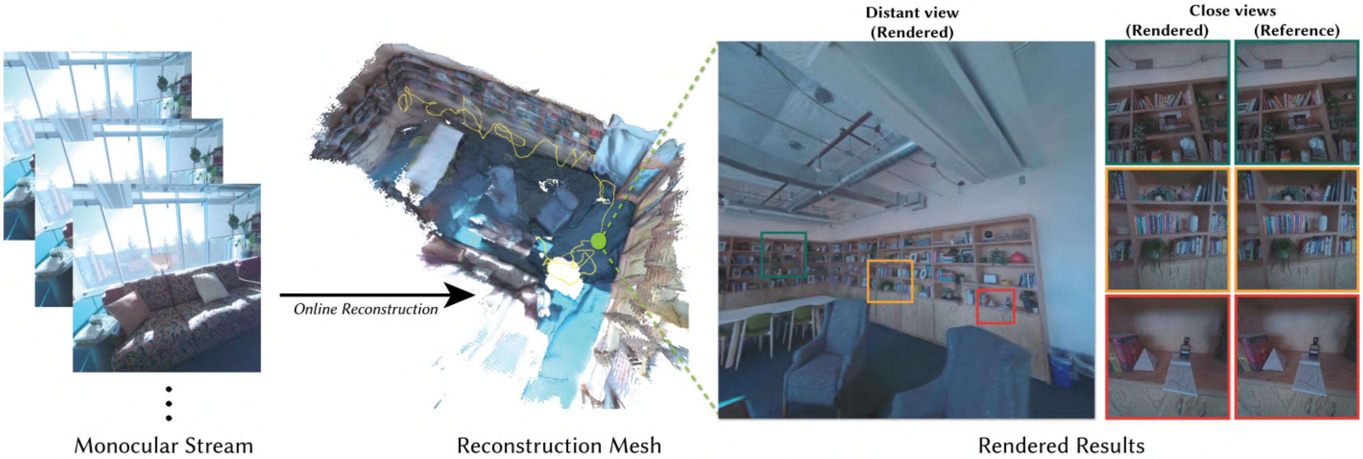


Fig. 1. Our pipeline processes a stream of monocular RGB images to reconstruct scenes with immediate feedback. Our method produces high-quality photorealistic maps with detailed reconstruction across multiple levels. The middle image illustrates our reconstructed mesh, and the right image showcases the rendered results of our reconstructed map, which captures high-quality details at both coarse and fine levels.

We propose an online 3D Gaussian-based dense mapping framework for photorealistic details reconstruction from a monocular image stream. Our approach addresses two key challenges in monocular online reconstruction: distributing Gaussians without relying on depth maps and ensuring both local and global consistency in the reconstructed maps. To achieve this, we introduce two key modules: the *Hierarchical Gaussian Management Module* for effective Gaussian distribution and the *Global Consistency Optimization Module* for maintaining alignment and coherence at all scales. In addition, we present the *Multi-level Occupancy Hash Voxels* (MOHV), a structure that regularizes Gaussians for capturing details across multiple levels of granularity. MOHV ensures accurate reconstruction of both fine and coarse geometries and textures, preserving intricate details while maintaining overall structural integrity. Compared to state-of-the-art RGB-only and even RGB-D methods, our framework achieves superior reconstruction quality with high computational efficiency. Moreover, it integrates seamlessly with various tracking systems, ensuring generality and scalability. Project page: https://poiw.github.io/MODP/.

Authors' Contact Information: Songyin Wu, Meta Reality Labs Research, Seattle, USA and University of California Santa Barbara, Santa Barbara, USA, s_wu975@ucsb.edu; Zhaoyang Lv, Meta Reality Labs Research, San Francisco, USA, zhaoyang@meta.com; Yufeng Zhu, Meta Reality Labs Research, Seattle, USA, yufengzhu@meta.com; Duncan Frost, Meta Reality Labs Research, London, United Kingdom, frost@meta.com; Zhengqin Li, Meta Reality Labs Research, Seattle, USA, zhl@meta.com; Ling-Qi Yan, University of California Santa Barbara, Santa Barbara, USA, lingqi@cs.ucsb.edu; Carl Ren, Meta Reality Labs Research, Seattle, USA, carlren@meta.com; Richard Newcombe, Meta Reality Labs Research, Seattle, USA, newcombe@meta.com; Zhao Dong, Meta Reality Labs Research, Seattle, USA, zhaodong@meta.com.

SIGGRAPH Conference Papers '25, Vancouver, BC, Canada
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1540-2/25/08
https://doi.org/10.1145/3721238.3730659

CCS Concepts: • **Computing methodologies** → **Shape modeling**; *Rendering*.

Additional Key Words and Phrases: Online reconstruction, 3D Gaussian splatting, Monocular

ACM Reference Format:
Songyin Wu, Zhaoyang Lv, Yufeng Zhu, Duncan Frost, Zhengqin Li, Ling-Qi Yan, Carl Ren, Richard Newcombe, and Zhao Dong. 2025. Monocular Online Reconstruction with Enhanced Detail Preservation. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3721238.3730659

1 Introduction

Online dense reconstruction, which generates environment models from a continuous stream of input images, is a fundamental challenge in robotics, computer vision, and computer graphics. It forms the cornerstone of interactive environmental understanding and interaction, enabling a wide range of applications such as augmented and virtual reality (AR/VR), robotics, and the emerging field of spatial AI. By processing sensor data streams to interactively reconstruct environments, it provides immediate feedback to the scanner, facilitating downstream tasks like scene understanding, active reconstruction, and more.

Previous works have explored various map representations, including point clouds [Du et al. 2011], surfels [Keller et al. 2013; Whelan et al. 2015], and signed distance functions [Newcombe et al. 2011], to achieve high-quality geometry reconstruction in Simultaneous Localization and Mapping (SLAM) systems. However, these methods often fall short in reconstructing photorealistic appearances due to the limitations of their map representations, as their primary focus is on geometric accuracy rather than visual realism. The recent success of neural radiance fields (NeRFs) in view synthesis [Mildenhall et al. 2020] has paved the way for photorealistic scene reconstruction. NeRF-based methods [Johari et al. 2023; Sandström et al. 2023; Sucar et al. 2021; Wang et al. 2023; Yang et al. 2022; Zhu et al. 2024, 2022], when integrated into SLAM systems, mark a significant breakthrough in enhancing reconstructed maps with highly realistic appearances. However, these methods face challenges in achieving interactive frame rates for both reconstruction and rendering due to the computational demands of ray marching in volumetric rendering. Additionally, their high memory requirements make it difficult to scale effectively to large scenes. In contrast, 3D Gaussian representations [Kerbl et al. 2023] model scenes as discrete Gaussian distributions, offering dramatically faster rendering and optimization speeds. This allows reconstructed scenes to be visualized at real-time frame rates. Recent works [Bai et al. 2024; Ha et al. 2025; Huang et al. 2024; Keetha et al. 2024; Matsuki et al. 2024; Sandström et al. 2024; Yan et al. 2024; Yugay et al. 2023; Zhang et al. 2024] have demonstrated that 3D Gaussian-based SLAM can produce high-quality reconstruction maps, achieving superior rendering quality compared to earlier non-radiance-field-based methods.

Dense SLAM systems can process various types of input streams. 3D Gaussian-based dense SLAMs with RGB-D inputs [Ha et al. 2025; Matsuki et al. 2024; Peng et al. 2024; Wang et al. 2024] achieve high-quality scene reconstruction, as the availability of accurate depth data allows Gaussians to be initialized near optimal locations. This facilitates rapid convergence with precise geometry and appearance. However, the reconstruction quality deteriorates significantly when relying solely on color frames (monocular input). Poor initialization of Gaussians often leads to independent optimization getting trapped in local minima, resulting in artifacts such as floaters and blurriness. In contrast to offline methods [Kerbl et al. 2023], which benefit from accurate camera poses and global optimization of the entire scene, incremental reconstruction in online SLAM systems faces challenges such as limited computational resources and the absence of global information. These issues lead to inconsistencies and lower-quality global maps. As a result, achieving photorealistic online reconstruction without depth maps remains a significant challenge.

In this paper, we aim to achieve photorealistic reconstruction using only monocular RGB frames, addressing two key challenges: the lack of dense depth maps and the need to produce globally consistent maps at interactive frame rates. Our key insight lies in controlling the distribution of Gaussians in world space based on error maps and feature complexity in image space. Furthermore, we design a multi-level occupancy hash voxel structure to regulate the distribution across different levels, ensuring coarse and fine details are recovered. In addition, we propose a view selection strategy that balances the reconstruction of newly observed local regions with the preservation of historical global maps, effectively avoiding local minima during optimization.

We evaluate our pipeline on standard datasets, including TUM [Sturm et al. 2012] and Replica [Straub et al. 2019], where it demonstrates superior reconstruction quality compared to previous monocular baselines and surpasses most RGB-D baselines. To further validate its capability to handle scenes with varying levels of detail for scenes in different scales, we capture additional indoor and outdoor sequences with complex geometries using Aria glass [Engel et al. 2023]. Moreover, our proposed mapping system is designed to be compatible with various tracking systems, highlighting its versatility. We showcase the generality of our approach by integrating it with different tracking systems, such as those in [Campos et al. 2021; Engel et al. 2023], demonstrating its scalability and effectiveness.

2 Related works

2.1 Classic Dense Visual SLAM

Over the last decade, dense visual SLAM-based 3D scene reconstruction has been a prominent research focus. For a comprehensive overview, readers are referred to detailed state-of-the-art surveys [Fuentes-Pacheco et al. 2015; Macario Barros et al. 2022; Zollhöfer et al. 2018] and foundational theses [Newcombe 2012]. Significant advancements in online 3D scene reconstruction have been achieved in RGB-D dense SLAM, employing diverse map representations such as point clouds [Du et al. 2011], Hermite radial basis functions [Xu et al. 2022], surfels [Cao et al. 2018; Keller et al. 2013; Whelan et al. 2015], and truncated signed distance functions (TSDFs) [Chen et al. 2013; Dai et al. 2017; Huang et al. 2021a; Newcombe et al. 2011; Nießner et al. 2013; Zhang et al. 2015]. For instance, ElasticFusion [Whelan et al. 2015] models scenes as collections of surfels, leveraging surfel-rendered depth and color for high-quality real-time tracking. TSDF-based BundleFusion [Dai et al. 2017] reconstructs large-scale scenes in real time through dynamic surface reintegration, achieving globally consistent 3D maps. DI-Fusion [Huang et al. 2021b] incorporates scene priors by encoding local geometry and modeling uncertainty using deep neural networks. While these methods focus primarily on geometric reconstruction, our approach simultaneously addresses surface reconstruction and photorealistic rendering, bridging the gap between accurate geometry and visually realistic output. Moreover, these prior methods rely heavily on depth maps to achieve high-quality reconstruction. In contrast, our approach operates using only a monocular RGB stream, making it more versatile and accessible.

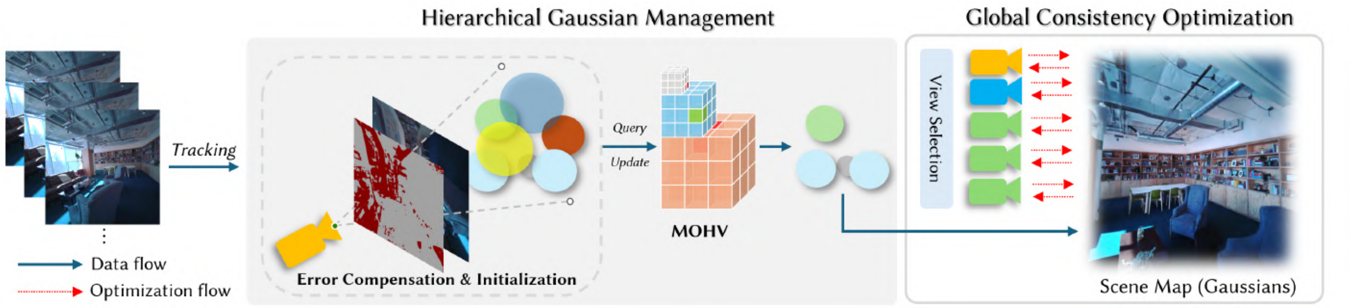

Hierarchical Gaussian Management
Global Consistency Optimization
Tracking
Query
Update
MOHV
View Selection
Error Compensation & Initialization
Data flow
Optimization flow
Scene Map (Gaussians)

Fig. 2. The figure illustrates an overview of our method. Our approach takes inputs from the tracking system and generates initial Gaussians based on regions with high geometric or texture complexity and significant errors (Sec. 3.2.2). Next, the MOHV module (Sec. 3.2.3) removes redundant Gaussians while preserving a high-quality map. Finally, the global consistency optimization module optimizes the Gaussians to produce a globally consistent map with details across various levels.

2.2 NeRF-based Dense Visual SLAM

Building upon the remarkable success of neural radiance fields (NeRF) [Mildenhall et al. 2020], recent works have integrated NeRF with RGB-D and RGB-only dense SLAM systems. In RGB-D SLAM, iMap [Sucar et al. 2021] pioneers NeRF SLAM by using a single MLP to represent the scene. NICE-SLAM [Zhu et al. 2022] introduces hierarchical feature grids decoded with pre-trained MLPs, while Vox-Fusion [Yang et al. 2022] represents scenes as voxel-based neural implicit surfaces stored in octrees. State-of-the-art methods, such as ESLAM [Johari et al. 2023] and Co-SLAM [Wang et al. 2023], adopt multi-resolution hierarchical structures to balance quality and performance, using feature grids and hash grids, respectively. Point-SLAM [Sandström et al. 2023] takes an alternative approach, leveraging neural point clouds with volumetric rendering and feature interpolation. For RGB-only SLAM, NICER-SLAM [Zhu et al. 2024] enhances accuracy and robustness by incorporating additional supervision signals, such as monocular geometric cues and optical flow, to jointly optimize camera poses and hierarchical neural implicit maps. While these methods deliver impressive results, their reliance on computationally intensive volumetric rendering limits interactive or real-time performance for online reconstruction of real-world scenes. Furthermore, their high memory requirements make reconstructing large-scale scenes impractical. In contrast, our method enables interactive reconstruction of large-scale scenes, offering significantly higher speeds and reduced memory costs.

2.3 3D Gaussian-based Dense Visual SLAM

Recently, 3D Gaussians [Kerbl et al. 2023] have gained traction as an efficient alternative for map representation in RGB-D and RGB-only dense SLAM systems. Traditional 3D Gaussian optimization, typically performed offline, requires several minutes to complete. To enable online reconstruction in RGB-D SLAM, methods such as [Wang et al. 2024; Yan et al. 2024; Yugay et al. 2023] introduce novel Gaussian seeding and optimization strategies for sequential input streams. SplaTAM [Keetha et al. 2024] incorporates Gaussian-based representations with silhouette-guided optimization via differentiable rendering. MonoGS [Matsuki et al. 2024] extends Gaussian representations for accurate tracking, mapping, and high-quality rendering in both RGB-D and RGB-only scenarios. More recently, RTG-SLAM [Peng et al. 2024] introduces a compact Gaussian representation with a highly efficient on-the-fly optimization scheme for RGB-D inputs, achieving real-time online scene reconstruction. GS-ICP SLAM [Ha et al. 2025] combines Generalized Iterative Closest Point (G-ICP) with 3D Gaussian Splatting (3DGS) to further enhance real-time RGB-D SLAM performance. For RGB-only SLAM, Photo-SLAM [Huang et al. 2024] utilizes a Gaussian-Pyramid training approach to improve mapping with multi-level features. Splat-SLAM [Sandström et al. 2024] dynamically adapts to keyframe pose and depth updates by deforming the 3D Gaussian map, ensuring globally optimized tracking and enhanced reconstruction accuracy. HI-SLAM2 [Zhang et al. 2024] combines monocular priors with learning-based dense SLAM to improve geometry estimation, achieving significant advancements in quality and performance for RGB-only SLAM. Some other concurrent works [Bai et al. 2024; Feng et al. 2024; Hu et al. 2024; Tianci et al. 2025] also attempt to reconstruct high-quality scenes from a monocular stream; however, none of them show very fine detailed reconstruction. Compared to these existing methods, our approach aims to achieve an optimal balance between reconstruction quality and performance. Quality-wise, our method excels in preserving fine details, significantly outperforming all RGB-only approaches and surpassing most RGB-D-based methods on standard datasets [Straub et al. 2019; Sturm et al. 2012]. Performance-wise, it delivers interactive online reconstruction.

3 Methods

The high-level ideas of our design are based on two key observations: the *importance of good Gaussian initialization* and the *need for global consistency optimization.* With only monocular input, achieving a well-initialized Gaussian distribution through regularization is crucial to decrease artifacts, including floater and over-blurriness, where we propose our *Hierarchical Gaussian Management* module (Sec. 3.2) with a densification strategy (Sec. 3.2.2) and a pruning mechanism through MOHV (Sec. 3.2.3). Building on this solid initialization, we propose *Global Consistency Optimization* (Sec. 3.3) to balance local rapid convergence with global consistency. An overview of our method is presented in Fig. 2.

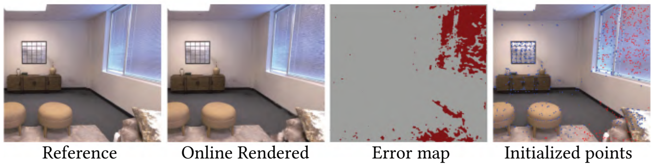


Fig. 3. A visualization of pixels used for initialization. In the error map, red regions indicate high-error areas. In the right-most image, red points represent pixels used for error compensation, while blue points correspond to pixels in geometry- or texture-complex regions.

3.1 Tracking

Our method emphasizes a high-quality mapping system while ensuring compatibility with various tracking systems. We use ORB-SLAM3 [Campos et al. 2021] as an example tracking system in our pipeline, where it provides online camera poses c_i and sparse world space feature points P_i to our system. Additional results using different tracking systems will be presented in Sec. 4.7, showcasing the generalization and robustness of our mapping system. Note that we do not discuss the quality of different tracking systems, as our focus is on the mapping system.

3.2 Hierarchical Gaussian Management

Our pipeline uses 3D Gaussians [Kerbl et al. 2023] as scene representations, where the initialization and distribution of 3D Gaussians are critical, as each Gaussian is optimized independently. Previous RGB-D SLAM works [Ha et al. 2025; Matsuki et al. 2024; Peng et al. 2024] rely on depth maps to precisely initialize Gaussians, achieving higher reconstruction quality compared to methods using only RGB inputs [Huang et al. 2024; Matsuki et al. 2024; Sandström et al. 2024; Zhang et al. 2024]. Instead, we propose a hierarchical Gaussian management module to avoid issues such as floaters and missing details caused by local minima. This module computes Gaussian and camera scales, strategically distributing Gaussians to geometry-complex and high-error regions to capture geometry and texture details better. Furthermore, to address redundancy or insufficiency in Gaussian placement across varying levels of scene detail, we propose a Multi-level Occupancy Hash Voxel (MOHV) structure. MOHV dynamically regulates Gaussian density at different scales, enabling high-quality reconstruction of fine details while maintaining computational efficiency.

3.2.1 Camera and Gaussian Scales. Gaussian scales are critical in the optimization process: excessively large scales cause over-blurred results, while excessively small scales hinder convergence. To effectively capture details at varying levels with appropriate scales, we define the scales of both the camera and Gaussians in world space. Specifically, given the camera's focal length f_i and the pixel's depth value d_t (from the tracker), the Gaussian size corresponding to such a pixel is calculated using the approach detailed in Yu et al. [2024].

$$s_t = \frac{d_t}{f_i} + \varepsilon \tag{1}$$

where s_t represents the size of the corresponding Gaussian in world space, and ε is a constant scalar. The scale of a camera view is set as the median value of the scales of all its sparse feature points.

3.2.2 Gaussian Densification. Previous works densify Gaussians mainly based on gradient magnitude [Kerbl et al. 2023] without explicitly targeting high-frequency regions with complex geometries and textures. In contrast, our approach leverages image-space information to directly initialize new Gaussians in the areas characterized by complex geometries, intricate textures, and high errors. An example of this initialization process is illustrated in Fig. 3. Specifically, we initialize new Gaussians from pixels in the following regions:

Geometry/texture complex regions. Tracking systems extract feature points from each frame to calculate the camera poses. These feature points are often located on high-contrast boundaries, which generally correspond to complex geometries or textures and require more Gaussians for accurate reconstruction. In contrast, textureless areas contain fewer feature points and consequently demand fewer Gaussians for reconstruction. Leveraging the feature point distribution from the tracking system provides a balanced approach to handling both high-frequency and low-frequency regions.

High error regions. In addition to the tracked feature points, some regions with rich textures or complex geometries may still lack sufficient Gaussians. To address this, we calculate the SSIM (structural similarity index measure) between the rendered images $\bar{I}_i$ and the observed ground truth images I_i. Additional k pixels for each keyframe are compensated in regions defined as $\bar{P}_i = \{\bar{p}_t \in \mathbb{R}^3 | SSIM(\bar{I}_i, I_i)[t] < \varepsilon_e\}$, where ε_e is a predefined threshold. The depth values for these pixels are estimated using a pre-trained model [Dexheimer and Davison 2023]. Notably, since we only require a few sparse points to compensate for these regions, computing a complete depth map for all pixels is unnecessary. This approach is significantly faster and contrasts with other methods [Sandström et al. 2024; Zhang et al. 2024] that rely on pre-trained models to predict full-depth maps.

3.2.3 Multi-level Occupancy Hash Voxel. Directly using all points in P_i and $\bar{P}_i$ often leads to redundant and overlapping Gaussians, as many tracked points are clustered in similar positions. Additionally, scenes with varying levels of detail require dynamic adjustments: increasing Gaussian density in detailed regions, such as when zooming in, while avoiding excessive Gaussians in coarser areas. To eliminate redundancy and maintain efficiency, we propose a Multi-level Occupancy Hash Voxel (MOHV) structure to effectively remove redundant Gaussians and dynamically regulate their distribution in the world space across different levels.

K-nearest neighbors remove redundant Gaussians by rejecting those within a threshold distance of each other, but this approach becomes increasingly slow as the number of Gaussians grows in larger scenes. While occupancy voxels enable fast location queries to determine whether a position is occupied, their high memory requirements limit the resolution of fine-level voxels. To overcome these challenges, we adopt a multi-level hash structure inspired by Instant-NGP [Müller et al. 2022], which significantly reduces the memory consumption of occupancy voxels by leveraging the sparsity inherent in reconstructed scenes. The MOHV module is defined by three parameters: number of levels L, initial scales S_{init} for the coarsest level scales, and number of voxels per dimension n, resulting in a total of n^3 voxels for each level. The total memory

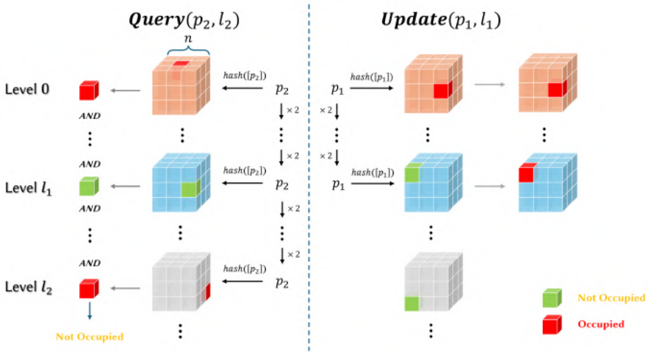


Fig. 4. The high-level concept of MOHV. It updates and queries multi-level voxels up to a given level l to maintain Gaussian distributions for capturing details at various levels.

cost of this structure is only $O(Ln^3)$. At the same time, it achieves a voxel resolution of $S_{\text{init}}/2^L$, as each successive level represents double the resolution of the previous one.

Fig. 4 illustrates the high-level concept of the query and update operations in the MOHV module. When updating a position as occupied, the module marks all corresponding voxels up to the specified level, ensuring that finer-level occupancy remains unaffected. When querying a position, the module performs an AND operation on all occupancy information from the coarsest level to the specified level. This ensures that updates to finer occupancy levels are reflected in queries at coarser levels, aligning with our design objectives. Detailed algorithmic descriptions are provided in the supplementary.

Finally, MOHV removes Gaussians $\{P_i, \bar{P}_i\}$ located in occupied positions and updates its structure for the remaining Gaussians. The remaining Gaussians are initialized and added to the scene map with corresponding world positions, scales calculated through Eq. 1, and colors with corresponding pixels.

3.3 Global Consistency Optimization

After distributing Gaussians in world space, their optimization is guided by selected views. Achieving a balance between local and global maps is challenging, as it requires both rapid convergence in newly observed regions and the preservation of previously reconstructed areas. To address this, the global consistency optimization module jointly optimizes Gaussians and online camera poses, ensuring a globally consistent reconstruction. This is accomplished through a carefully designed view selection strategy and optimization process, consisting of the following components:

3.3.1 Keyframe Selection. Using all frames for global optimization is redundant and usually yields lower quality. Instead, we select a subset of frames as keyframes for our global optimization. A frame i is added as a keyframe if the overlap ratio $\text{covis}(i, j)$ between the current frame i and the previous keyframe j is below a threshold or if t_k frames have elapsed since the last keyframe, accounting for textureless regions. The overlap ratio is calculated using tracked points instead of Gaussians to improve computational efficiency.

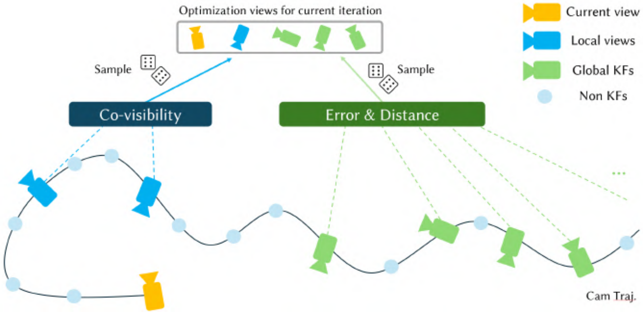


Fig. 5. High-level concept of optimization view selection. Our method samples local views based on covisibility, and global views based on their L1 error values and distance to the current view.

3.3.2 Optimization View Selection. Unlike offline 3DGS optimization [Kerbl et al. 2023], where all Gaussians are optimized simultaneously, online mapping requires incremental optimization. This makes it essential to balance newly observed views with historical ones. As shown in Fig. 5, we propose a local and global camera selection strategy to achieve fast convergence for new frames while maintaining global consistency in previously reconstructed regions. Given the current view camera c_i, the local and global cameras are defined as:

Local Cameras. Local cameras aim to optimize newly observed regions with multi-view constraints. In our experiments, we set the number of additional local views, n_{local}, to 1, in addition to the current view. To ensure sufficient overlap with the current view, we maintain a local bank of size $\bar{n}_{\text{local}}$. A new frame is added to the bank every t_{local} frames and the oldest frame is discarded once the bank exceeds its maximum size. This bank selects the n_{local} views with the largest overlap with the current view for multi-view joint optimization.

Global Cameras. Local views converge quickly but cannot ensure a globally consistent map. Relying solely on local views often results in overfitting to specific regions, leading to poor global maps due to forgetting issues and camera drift. To address this, we also select n_{global} views from historical keyframes. However, randomly sampling from all historical keyframes creates an imbalance, with earlier frames being selected more frequently than recent ones. To mitigate this issue, we sample the historical keyframes based on the following probability:

$$\text{prob}_i[j] = \text{normalize}(e^{\sigma_1 \cdot (j-i)} \cdot e^{\sigma_2 \cdot \text{err}(j)}), \tag{2}$$

where $\text{err}(j)$ refers to the Mean Absolute Error of frame j, updated every time frame j is optimized. The first term in the probability distribution adjusts the selection to prioritize newly added keyframes, while the second term emphasizes under-optimized keyframes.

3.3.3 Camera Refinement. Although the online tracking system provides reasonably accurate camera poses, they are insufficient for reconstructing high-quality maps. To enhance reconstruction quality, we further optimize the poses of keyframes during the mapping process.

The scene's Gaussian map is optimized for each keyframe using rendering losses, incorporating the current view along with the selected local and global views at every step. Further details on the optimization parameters are provided in the supplementary.

4 Experiments

4.1 Dataset

To demonstrate the robustness of our pipeline, we evaluate our method on three datasets: TUM-RGBD [Sturm et al. 2012], Replica [Straub et al. 2019], as well as our own sequences captured using Aria glasses [Engel et al. 2023]. The TUM-RGBD dataset consists of real-world RGB-D images but includes challenging sequences with severe motion blur, which often degrade reconstruction quality. The Replica dataset provides highly accurate depth maps since it is re-rendered from reconstructed 3D models. Additionally, we use Aria glasses to capture indoor and outdoor sequences with varying geometric complexity. All frames captured with Aria glasses undergo consistent pre-processing operations before being tested on our pipeline and baselines, ensuring a fair comparison.

4.2 Baselines

We compare our method against existing dense SLAM baselines, including both monocular and RGB-D approaches. Specifically, we compare with prior works [Huang et al. 2024; Matsuki et al. 2024; Sandström et al. 2024] for monocular SLAM baselines and [Ha et al. 2025; Matsuki et al. 2024; Peng et al. 2024] for RGB-D SLAM baselines. We also compare with concurrent work [Zhang et al. 2024] using their released codes. For our custom-captured sequences, where depth maps are unavailable, we compare with monocular baselines only. To ensure fairness, all evaluations are performed on the full sequence of images at their original resolution.

Note that several methods (MonoGS [Matsuki et al. 2024], Splat-SLAM [Sandström et al. 2024], Hi-SLAM2 [Zhang et al. 2024]) include a post-refinement step unrelated to the online reconstruction process. To ensure a fair comparison, we evaluate the results both before and after the post-refinement step separately. For Aria sequences, we run $26K$ steps for baselines in the post-refinement, while our method uses only $1K$ post-refinement steps for all scenes since it is not an essential step for ours.

4.3 Mapping and Rendering Quality

Our main objective is to reconstruct high-quality maps with a photorealistic appearance. Table 1 presents a quantitative comparison of our pipeline against existing monocular and RGB-D baselines. We use peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [Wang et al. 2004], and perceptual similarity (LPIPS) [Zhang et al. 2018] as our evaluation metrics. PSNR and SSIM measure overall reconstruction quality, while LPIPS emphasizes the preservation of fine details.

Table 1. Comparison of reconstruction rendering quality on different datasets. **Bold** refers to the best across all categories and green refers to the best of each category. Numbers of RTG-SLAM (Replica), GS-ICP (TUM and Replica) and Photo-SLAM (TUM and Replica) are taken from their original papers. Other numbers are calculated through their released codes.

	Methods	Metrics	TUM	Replica	Aria
RGB-D	MonoGS	PSNR	17.90	36.67	n/a
		SSIM	0.716	0.958	n/a
		LPIPS	0.322	0.072	n/a
	RTG-SLAM	PSNR	19.44	35.43	n/a
		SSIM	0.760	0.982	n/a
		LPIPS	0.408	0.109	n/a
	GS-ICP	PSNR	20.72	**38.83**	n/a
		SSIM	0.768	**0.975**	n/a
		LPIPS	0.218	**0.041**	n/a
Monocular	MonoGS	PSNR	17.54	27.38	18.34
		SSIM	0.698	0.860	0.475
		LPIPS	0.341	0.261	0.700
	Photo-SLAM	PSNR	20.54	33.30	23.40
		SSIM	0.720	0.926	0.615
		LPIPS	0.211	0.078	0.477
	Splat-SLAM	PSNR	22.34	30.37	21.58
		SSIM	0.731	0.886	0.561
		LPIPS	0.353	0.221	0.606
	Hi-SLAM2	PSNR	20.09	30.74	19.60
		SSIM	0.680	0.897	0.520
		LPIPS	0.379	0.208	0.675
	Ours	PSNR	25.45	35.85	26.15
		SSIM	0.866	0.956	0.678
		LPIPS	0.165	0.071	0.338
Monocular (w Post Refinement)	MonoGS	PSNR	22.04	30.13	21.05
		SSIM	0.737	0.900	0.555
		LPIPS	0.326	0.193	0.662
	Splat-SLAM	PSNR	25.53	33.72	24.44
		SSIM	0.801	0.938	0.618
		LPIPS	0.251	0.117	0.490
	Hi-SLAM2	PSNR	23.52	36.69	25.64
		SSIM	0.805	0.953	0.665
		LPIPS	0.242	0.113	0.414
	Ours	PSNR	**26.18**	36.89	**26.62**
		SSIM	**0.874**	0.962	**0.693**
		LPIPS	**0.154**	0.061	**0.324**

Our method outperforms all monocular baselines and surpasses RGB-D baselines in real-world captured scenarios. While RGB-D methods excel on the synthetic Replica dataset, which provides perfect depth maps, their quality degrades in real-world cases with imperfect depth data. The Aria dataset, specifically captured to test scenes with varying levels of detail, highlights our method's superior performance, particularly in LPIPS, demonstrating enhanced detail preservation compared to other baselines.

Qualitative comparisons are shown in Fig. 7, Fig. 8, and Fig. 9. In complex and challenging scenes with varying detail levels, our pipeline reconstructs significantly better details, while other baselines produce overly blurred results, even after applying a global post-refinement process.

Table 2. Tracking accuracy of different monocular baselines and our methods. Numbers represent absolute trajectory error (ATE) root mean square error (RMSE) in cm. All baseline numbers are taken from their original paper except Hi-SLAM2 (TUM) and MonoGS (Replica) since they didn't include those trajectory accuracy in their paper.

	MonoGS	PhotoSLAM	SplatSLAM
TUM	3.96	1.26	**1.1**
Replica	22.03	1.09	0.35
	Hi-SLAM2	ORB-SLAM3	Ours
TUM	1.32	1.98	1.87
Replica	**0.26**	3.88	3.75

Table 3. Speed performance of different methods. Numbers represent frames per second (FPS).

	TUM	Replica	Aria
MonoGS	2.91	1.43	1.85
Photo-SLAM	65.87	41.65	10.02
Splat-SLAM	3.62	1.01	0.46
Hi-SLAM2	13.99	15.08	3.35
Ours	11.28	9.34	4.55

Table 4. Time breakdown of each component. The reported numbers represent the percentage of time consumed by each component relative to the overall process. "Others" primarily includes data transfer and preprocessing operations.

	Optim	Cam Select	Depth Est.	MOHV	Others
PCT (%)	91.44	0.79	1.63	0.87	5.27

4.4 Tracking Accuracy

Although our framework is designed to be compatible with various tracking systems and does not specifically focus on tracking accuracy, we analyze the tracking performance to better understand its impact on the quality of our online reconstruction framework. Table 2 presents the tracking errors of different methods. While our framework exhibits slightly lower tracking accuracy, it consistently delivers higher-quality reconstruction maps, demonstrating the effectiveness of our online reconstruction pipeline. Improving integration with tracking systems is left for future work to further enhance reconstruction quality.

4.5 Speed Performance

All experiments were conducted on a Fedora machine with an AMD Ryzen Threadripper PRO 3975WX and an NVIDIA RTX 4090. As shown in Table 3, our method achieves approximately 10 FPS on small-scale scenes (TUM and Replica) and around 5 FPS on relatively larger scenes (Aria) while maintaining high reconstruction quality. Note that Photo-SLAM [Huang et al. 2024], implemented entirely in C++, is several orders of magnitude faster than other pipelines built with Python.

Table 5. Ablation studies of our Gaussian management module. Numbers are averaged from all Aria sequences. **Bold** refers to the best and underline refers to the second best.

	PSNR	SSIM	LPIPS	# Gaussians
w/o EC	24.87	0.647	0.407	123,396
w/o MOHV	26.04	**0.680**	**0.333**	445,764
w/o EC & MOHV	25.30	0.661	0.375	221,370
Ours full	**26.15**	0.678	0.338	340,962

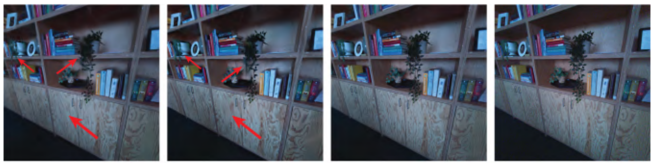

w/o EC w/o EC & MOHV Ours full Reference

Fig. 6. Ablation studies of Gaussian management module. MOHV refers to Multi-level Occupancy Hash Voxels and EC refers to Error Compensation. Zoom in for details.

A detailed time breakdown is provided in Table 4, demonstrating that our global camera selection, depth estimation, and MOHV modules are sufficiently efficient and do not constitute the pipeline's bottleneck. Additional details can be found in the supplementary material.

4.6 Ablation Studies

In this section, we perform comprehensive ablation studies to evaluate the effectiveness of individual modules in our framework. As our designs primarily focus on reconstructing details across various levels of the scene, the ablation studies are conducted on the Aria sequences, which feature scenes with diverse geometric complexity.

Error region compensation. Tracking systems without dense depth maps typically provide only sparse 3D tracked points, which are insufficient for reconstructing fine details. Our error region compensation module adds supplementary points in high-error and under-optimized regions. As illustrated in Fig. 6, incorporating error region compensation enhances detail in areas where tracked points are sparse or missing. Without it, the textures on the bookshelf and the details on the plant are noticeably missing in Fig. 6. Note that while increasing the number of tracked points by adjusting the tracking system's threshold is possible, doing so may adversely affect the overall tracking quality.

Multi-level occupancy hash voxel. The MOHV module removes redundant Gaussians within local regions based on camera scales. As shown in Table. 5 and Fig. 6, although with about 100K more Gaussians, the final reconstructed quality does not noticeably improve. This demonstrates that MOHV preserves reconstruction quality while reducing the number of Gaussians by approximately 30%. By effectively controlling the growth of Gaussians, this module enables scalability to large-scale scenes without redundant overhead.

Table 6. Ablation studies of our global consistent optimization module. **Bold** refers to the best.

	PSNR	SSIM	LPIPS
w/o global cams	17.68	0.460	0.595
w/o cam refinement	25.52	0.666	0.341
Ours full	**26.15**	**0.678**	**0.338**

Table 7. Comparison of rendering quality on reconstructed maps with different tracking systems and offline 3DGS on Aria sequences. PR. refers to the post-refinement process. PR. steps in offline 3DGS refers to the total optimization steps.

	PSNR	SSIM	LPIPS	PR. Steps	# Gs	Time
Ours	26.15	0.678	0.338	0	341K	4m1s
Ours (w PR.)	26.62	0.693	0.324	1K	341K	4m56s
Ours (w PR.)	**27.44**	**0.711**	0.297	20K	341K	9m32s
Ours (Aria)	25.07	0.659	0.330	0	700K	3m55s
Ours (Aria, w PR)	25.67	0.675	0.315	1K	700K	4m32s
Ours (Aria, w PR)	25.99	0.682	**0.294**	20K	700K	10m24s
Offline 3DGS	26.60	0.696	0.305	150K	1,781K	1h6m21s

Global view optimization. Global view optimization is essential for maintaining a globally consistent map, preventing forgetting issues, and balancing newly observed regions with previously reconstructed ones. For fairness, we add the same number of local views to ensure an equal optimization budget when global view optimization is removed. As shown in Table 6 and Fig. 11, the quality of previously reconstructed regions significantly deteriorates when the optimization focuses solely on newly observed regions.

Camera refinement. Camera refinement is used to globally optimize both the Gaussians and keyframes camera poses to improve reconstruction quality. As shown in Fig. 12, joint camera refinement improves the details in the reconstruction.

4.7 Alternative Tracking Systems

Our framework is a general online mapping system, which is not limited to the ORB-SLAM3 tracking system. In this section, we also show the results of using the Aria tracking system [Engel et al. 2023] which uses Aria's two SLAM cameras and an inertial measurement unit (IMU) to perform stereo tracking. Notably, our mapping system does not directly use SLAM cameras and IMU's information and only takes online tracked poses and sparse points as inputs to align our monocular online reconstruction settings. Table. 7 and Fig. 10 compare our method using ORB-SLAM3 and Aria tracking systems. The results demonstrate comparable performance across both systems. The slightly lower PSNR/SSIM scores with Aria tracking can be attributed to its sparse feature points, which prioritize mid-range objects over distant ones to enhance detail reconstruction. This experiment highlights the robustness and versatility of our framework, showcasing its compatibility with different tracking systems.

4.8 Post Refinement

Our method is designed for high-quality online reconstruction, with the option of a post-refinement process to further enhance reconstruction quality starting from the results of our online pipeline. Table. 7 and Fig. 10 show our method with different steps in the post-refinement process as well as the offline 3DGS baseline [Kerbl et al. 2023]. Although offline 3DGS is a global optimization approach, our methods with more post-refinement steps show better reconstruction results while requiring less time and fewer Gaussians.

5 Conclusion

In this work, we present a high-quality online reconstruction pipeline for reconstructing environments from monocular inputs. Our pipeline incorporates a hierarchical Gaussian management module and a global consistency optimization module, enabling the maintenance of Gaussians to capture details across various levels while remaining computationally efficient.

However, our method has certain limitations. One notable limitation arises when the tracking system loses tracking or when trajectory accumulation errors become significant. In the future, our method could be improved by explicitly addressing significant camera shifting issues through loop closure, extending the pipeline's applicability to even larger scenes.

Acknowledgments

We would like to thank Chieh Hubert Lin, Lei Xiao, Numair Khan, Chris Sweeney, Julian Straub, and Carl Marshall for their valuable discussions on the project. We also thank the anonymous reviewers for their insightful feedback. The pseudocode template in the supplementary is from Zhou et al. [2024]. The project was completed during Songyin Wu's internship at Meta Reality Labs Research. Ling-Qi Yan is supported by gift funds from Meta, Adobe, Linctex, and XVerse.

References

Lizhi Bai, Chunqi Tian, Jun Yang, Siyu Zhang, Masanori Suganuma, and Takayuki Okatani. 2024. RP-SLAM: Real-time Photorealistic SLAM with Efficient 3D Gaussian Splatting. arXiv:2412.09868 [cs.RO] https://arxiv.org/abs/2412.09868

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

Yan-Pei Cao, Leif Kobbelt, and Shi-Min Hu. 2018. Real-time high-accuracy three-dimensional reconstruction with consumer RGB-D cameras. *ACM Transactions on Graphics (TOG)* 37, 5 (2018), 1–16.

Jiawen Chen, Dennis Bautembach, and Shahram Izadi. 2013. Scalable real-time volumetric surface reconstruction. *ACM Trans. Graph.* 32, 4 (2013), 113–1.

Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. 2017. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 1.

Eric Dexheimer and Andrew J. Davison. 2023. Learning a Depth Covariance Function. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

Hao Du, Peter Henry, Xiaofeng Ren, Marvin Cheng, Dan B Goldman, Steven M Seitz, and Dieter Fox. 2011. Interactive 3D modeling of indoor environments with a consumer depth camera. In *Proceedings of the 13th international conference on Ubiquitous computing.* 75–84.

Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, et al. 2023. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561* (2023).

Dapeng Feng, Zhiqiang Chen, Yizhen Yin, Shipeng Zhong, Yuhua Qi, and Hongbo Chen. 2024. CaRtGS: Computational Alignment for Real-Time Gaussian Splatting SLAM. arXiv:2410.00486 [cs.CV] https://arxiv.org/abs/2410.00486

Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. 2015. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review* 43 (2015), 55–81.

Seongbo Ha, Jiung Yeon, and Hyeonwoo Yu. 2025. Rgbd gs-icp slam. In *European Conference on Computer Vision*. Springer, 180–197.

Yan Song Hu, Nicolas Abboud, Muhammad Qasim Ali, Adam Srebrnjak Yang, Imad Elhajj, Daniel Asmar, Yuhao Chen, and John S. Zelek. 2024. MGSO: Monocular Real-time Photometric SLAM with Efficient 3D Gaussian Splatting. arXiv:2409.13055 [cs.RO] https://arxiv.org/abs/2409.13055

Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. 2024. 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*. 21584–21593.

Jiahui Huang, Shi-Sheng Huang, Haoxuan Song, and Shi-Min Hu. 2021b. Di-fusion: Online implicit 3d reconstruction with deep priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8932–8941.

Shi-Sheng Huang, Haoxiang Chen, Jiahui Huang, Hongbo Fu, and Shi-Min Hu. 2021a. Real-time globally consistent 3D reconstruction with semantic priors. *IEEE transactions on visualization and computer graphics* 29, 4 (2021), 1977–1991.

Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. 2023. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17408–17419.

Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. 2024. SplaTAM: Splat Track & Map 3D Gaussians for Dense RGB-D SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21357–21366.

Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. 2013. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *3DV 2013*. IEEE, 1–8.

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

Andréa Macario Barros, Maugan Michel, Yoann Moline, Gwenolé Corre, and Frédérick Carrel. 2022. A comprehensive survey of visual slam algorithms. *Robotics* 11, 1 (2022), 24.

Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. 2024. Gaussian Splatting SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, Article 102 (2022), 15 pages.

Richard A. Newcombe. 2012. *Dense Visual SLAM*. Ph. D. Dissertation. Imperial College London. https://rapiderobot.bitbucket.io/papers/Newcombe-RA-Thesis-2014-compressed.pdf

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

Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. 2013. Real-time 3D reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)* 32, 6 (2013), 1–11.

Zhexi Peng, Tianjia Shao, Yong Liu, Jingke Zhou, Yin Yang, Jingdong Wang, and Kun Zhou. 2024. Rtg-slam: Real-time 3d reconstruction at scale using gaussian splatting. In *ACM SIGGRAPH 2024 Conference Papers*. 1–11.

Erik Sandström, Yue Li, Luc Van Gool, and Martin R Oswald. 2023. Point-slam: Dense neural point cloud-based slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 18433–18444.

Erik Sandström, Keisuke Tateno, Michael Oechsle, Michael Niemeyer, Luc Van Gool, Martin R Oswald, and Federico Tombari. 2024. Splat-SLAM: Globally Optimized RGB-only SLAM with 3D Gaussians. *arXiv preprint arXiv:2405.16544* (2024).

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. 2019. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797* (2019).

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. 2012. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 573–580.

Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. 2021. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF international conference on computer vision*. 6229–6238.

Wen Tianci, Liu Zhiang, Lu Biao, and Fang Yongchun. 2025. Scaffold-SLAM: Structured 3D Gaussians for Simultaneous Localization and Photorealistic Mapping. arXiv:2501.05242 [cs.CV] https://arxiv.org/abs/2501.05242

Hengyi Wang, Jingwen Wang, and Lourdes Agapito. 2023. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13293–13302.

Meng Wang, Junyi Wang, Changqun Xia, Chen Wang, and Yue Qi. 2024. OG-Mapping: Octree-based Structured 3D Gaussians for Online Dense Mapping. arXiv:2408.17223 [cs.CV] https://arxiv.org/abs/2408.17223

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

Thomas Whelan, Stefan Leutenegger, Renato F Salas-Moreno, Ben Glocker, and Andrew J Davison. 2015. ElasticFusion: Dense SLAM without a pose graph.. In *Robotics: science and systems*, Vol. 11. Rome, Italy, 3.

Yabin Xu, Liangliang Nan, Laishui Zhou, Jun Wang, and Charlie CL Wang. 2022. Hrbf-fusion: Accurate 3d reconstruction from rgb-d data using on-the-fly implicits. *ACM Transactions on Graphics (TOG)* 41, 3 (2022), 1–19.

Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. 2024. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19595–19604.

Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. 2022. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 499–507.

Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. 2024. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19447–19456.

Vladimir Yugay, Yue Li, Theo Gevers, and Martin R Oswald. 2023. Gaussian-slam: Photo-realistic dense slam with gaussian splatting. *arXiv preprint arXiv:2312.10070* (2023).

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 586–595.

Wei Zhang, Qing Cheng, David Skuddis, Niclas Zeller, Daniel Cremers, and Norbert Haala. 2024. HI-SLAM2: Geometry-Aware Gaussian SLAM for Fast Monocular Scene Reconstruction. *arXiv preprint arXiv:2411.17982* (2024).

Yizhong Zhang, Weiwei Xu, Yiying Tong, and Kun Zhou. 2015. Online structure analysis for real-time indoor scene reconstruction. *ACM Transactions on Graphics (TOG)* 34, 5 (2015), 1–13.

Yang Zhou, Songyin Wu, and Ling-Qi Yan. 2024. Unified Gaussian Primitives for Scene Representation and Rendering. arXiv:2406.09733 [cs.GR] https://arxiv.org/abs/2406.09733

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

Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. 2022. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12786–12796.

Michael Zollhöfer, Patrick Stotko, Andreas Görlitz, Christian Theobalt, Matthias Nießner, Reinhard Klein, and Andreas Kolb. 2018. State of the art on 3D reconstruction with RGB-D cameras. In *Computer graphics forum*, Vol. 37. Wiley Online Library, 625–652.

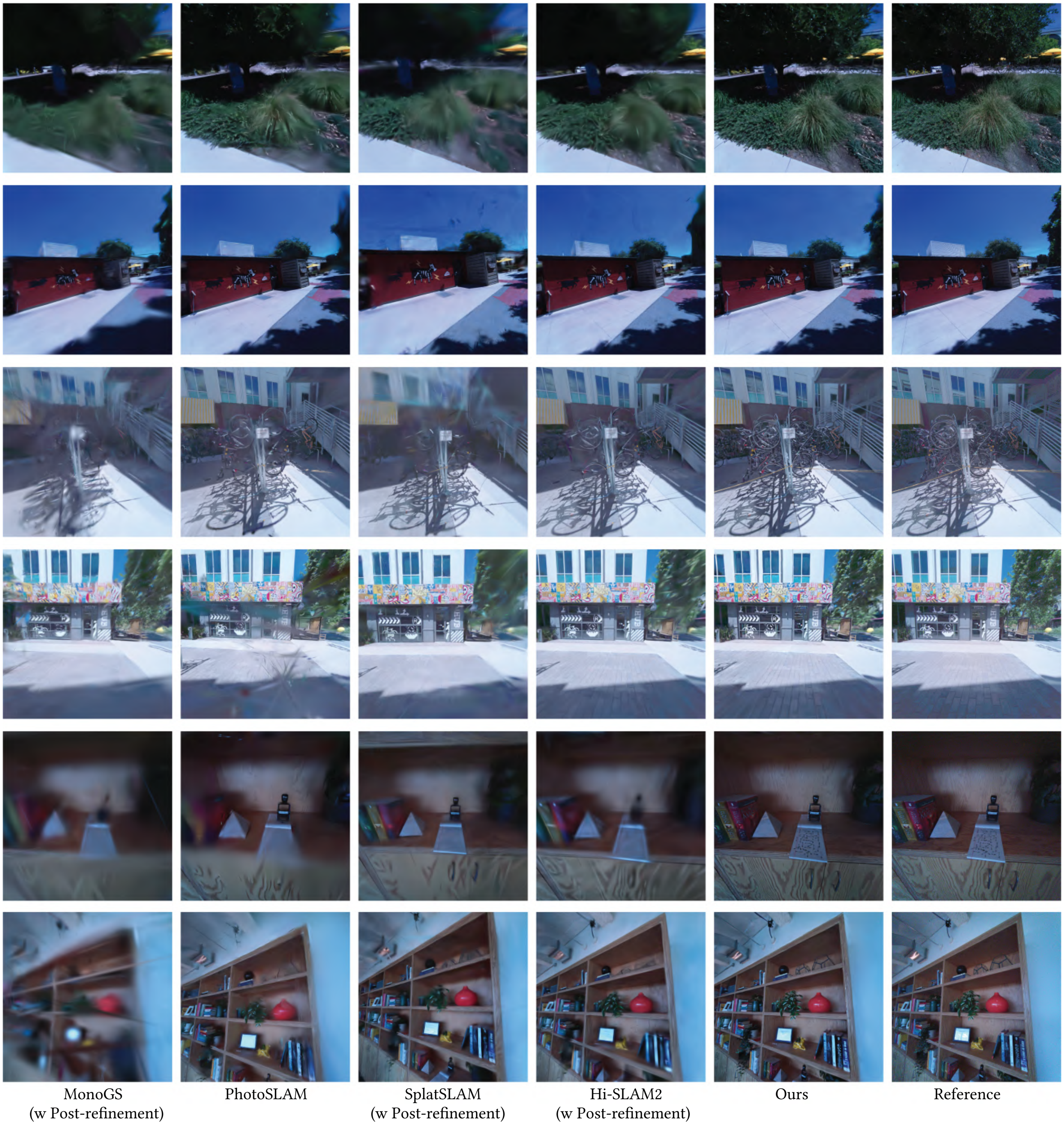
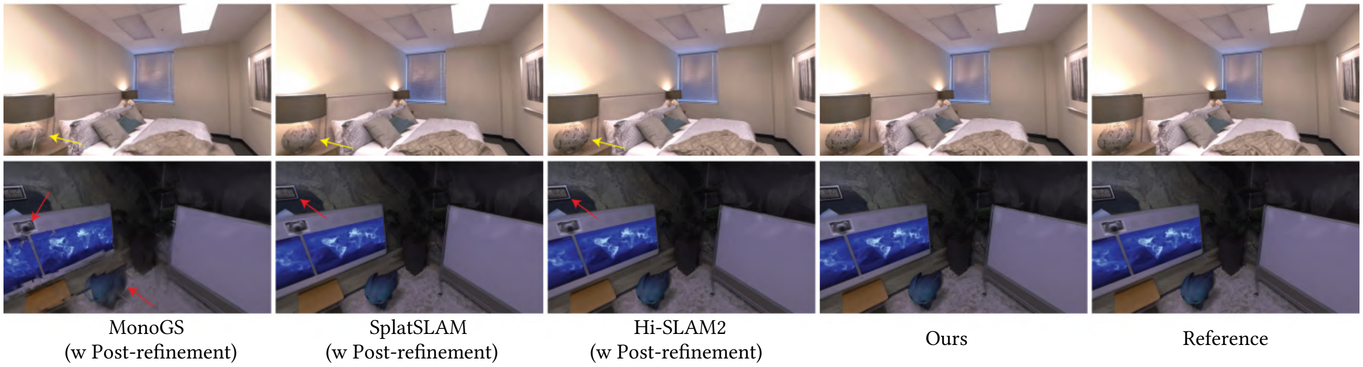
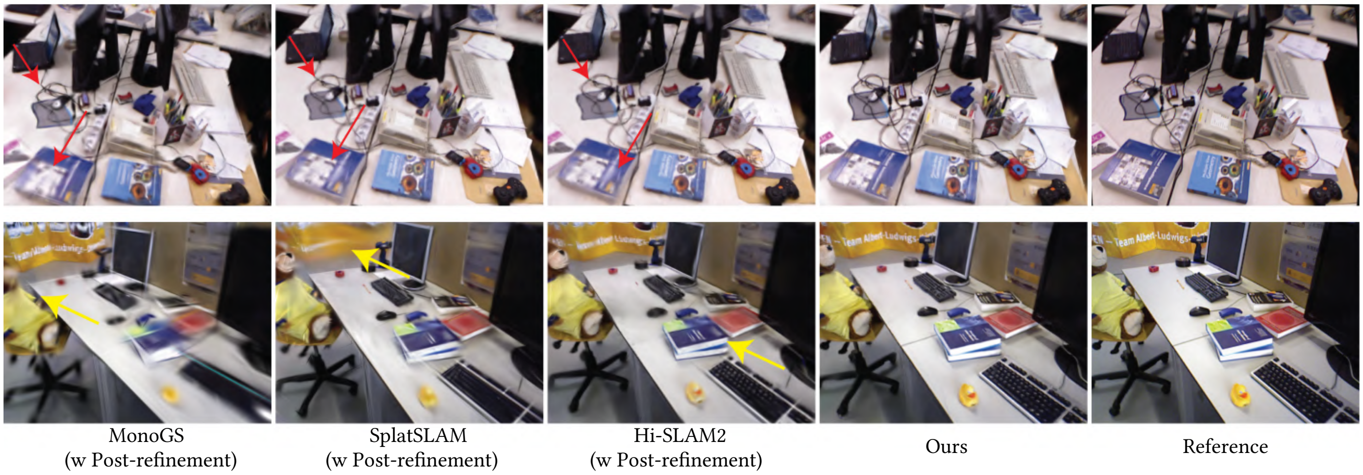


Fig. 7. Qualitative comparison on Aria captured sequences. Our method captures finer details, while other baselines produce over-blurred results. Notably, even after applying 26K post-refinement steps for the baselines, they still exhibit poorer details compared to our method. This demonstrates that high-quality, fine-level details cannot be achieved solely by increasing the number of optimization iterations. Zoom in for more details.



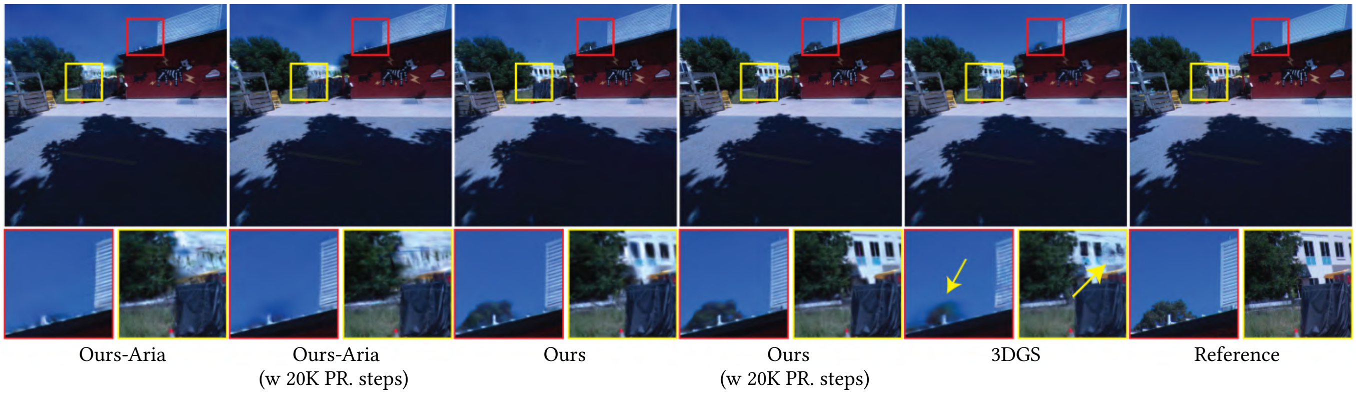
MonoGS (w Post-refinement) | SplatSLAM (w Post-refinement) | Hi-SLAM2 (w Post-refinement) | Ours | Reference

Fig. 8. Qualitative comparison on Replica dataset.



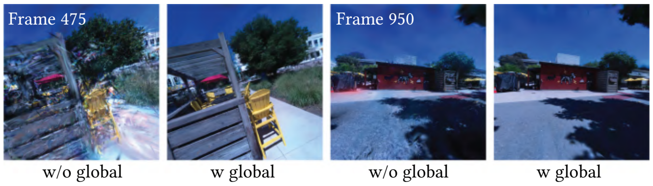
MonoGS (w Post-refinement) | SplatSLAM (w Post-refinement) | Hi-SLAM2 (w Post-refinement) | Ours | Reference

Fig. 9. Qualitative comparison on TUM dataset.



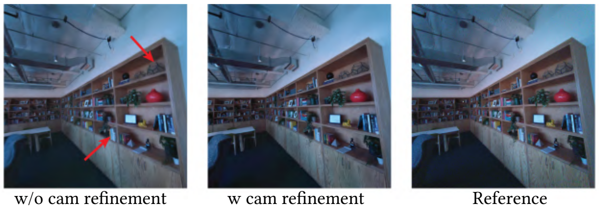
Ours-Aria | Ours-Aria (w 20K PR. steps) | Ours | Ours (w 20K PR. steps) | 3DGS | Reference

Fig. 10. Qualitative comparison of our methods with different tracking systems and offline 3DGS method. PR. refers to the post-refinement process.



w/o global | w global | w/o global | w global

Fig. 11. Ablation studies of global view optimization.



w/o cam refinement | w cam refinement | Reference

Fig. 12. Ablation studies of camera refinement.