IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

HI-SLAM2: Geometry-Aware Gaussian SLAM for Fast Monocular Scene Reconstruction

Wei Zhang, Qing Cheng, David Skuddis, Niclas Zeller, Daniel Cremers, and Norbert Haala

***Abstract*—We present HI-SLAM2, a geometry-aware Gaussian SLAM system that achieves fast and accurate monocular scene reconstruction using only RGB input. Existing neural SLAM or 3DGS-based SLAM methods often tradeoff between rendering quality and geometry accuracy, our research demonstrates that both can be achieved simultaneously with RGB input alone. The key idea of our approach is to enhance the ability for geometry estimation by combining easy-to-obtain monocular priors with learning-based dense SLAM, and then using 3-D Gaussian splatting as our core map representation to efficiently model the scene. Upon loop closure, our method ensures on-the-fly global consistency through efficient pose graph bundle adjustment and instant map updates by explicitly deforming the 3-D Gaussian units based on anchored keyframe updates. Furthermore, we introduce a grid-based scale alignment strategy to maintain improved scale consistency in prior depths for finer depth details. Through extensive experiments on Replica, ScanNet, Waymo Open, ETH3D SLAM and ScanNet++ datasets, we demonstrate significant improvements over existing neural SLAM methods and even surpass RGB-D-based methods in both reconstruction and rendering quality.**

***Index Terms*—Deep learning for visual perception, dense reconstruction, visual SLAM.**

I. Introduction

DENSE 3-D scene reconstruction from imagery remains one of the most fundamental challenges in computer vision, robotics, and photogrammetry. Achieving real-time and accurate 3-D reconstruction from images alone can enable numerous applications, from autonomous navigation to mobile surveying and immersive AR. While many existing solutions rely on RGB-D [1], [2], [3], [4], [5] or LiDAR sensors [6], [7], [8], [9], [10], these approaches have inherent limitations. LiDAR systems require expensive hardware setups and an additional camera for capturing color information, while RGB-D sensors suffer from limited operational range and sensitive to varying lighting conditions. Vision-based monocular scene reconstruction thus offers a promising lightweight and cost-effective alternative.

The fundamental challenge in monocular 3-D reconstruction stems from the lack of explicit scene geometry measurements [11]. Traditional visual SLAM methods [12], [13], [14], [15], [16], [17] developed over decades and typically provide only sparse or semidense map representations, proving insufficient for detailed scene understanding and complete reconstruction. While dense SLAM approaches [18], [19], [20] attempt to address this limitation through per-pixel depth estimation, they remain susceptible to significant depth noise and struggle to achieve complete, accurate reconstructions.

Recent advances in deep learning have revolutionized many key components of 3-D reconstruction, including optical flow [21], [22], depth estimation [23], [24], and normal estimation [24], [25]. These improvements have been integrated into SLAM systems through monocular depth networks [26], multiview stereo techniques [27], and end-to-end neural approaches [28]. However, even with these advancements, current systems often produce reconstructions with artifacts due to noisy depth estimates, limited generalization capability, or excessive computational requirements. The emergence of neural SLAM methods, particularly those based on neural implicit fields [5], [29], [30] and 3-D Gaussian Splatting (3DGS) [31], [32], [33], has shown promising results. Yet these approaches typically prioritize either rendering quality or geometry accuracy, creating an undesirable tradeoff. Our work addresses this limitation by simultaneously improving both aspects without compromise either. As shown in Fig. 1, our approach achieves superior performance across both geometry accuracy and rendering quality, surpassing not only RGB-based methods but also RGB-D-based approaches.

In this article, we aim to advance the state-of-the-art in dense monocular SLAM for 3-D scene reconstruction. We present HI-SLAM2, a geometry-aware Gaussian Splatting SLAM system that achieves accurate and fast monocular scene reconstruction using RGB input alone. The key idea of our approach lies in enhancing geometry estimation by combining monocular geometry priors with learning-based dense SLAM, while leveraging 3DGS as our compact map representation for efficient

Received 9 July 2025; accepted 21 September 2025. Date of publication 28 October 2025; date of current version 19 November 2025. This work was supported in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC 2120/1—project number 390831618. The work of Qing Cheng was supported in part by the DAAD program Konrad Zuse Schools of Excellence in Artificial Intelligence and in part by the Federal Ministry of Research, Technology and Space. This article was recommended for publication by Associate Editor and Editor J. Civera upon evaluation of the reviewers' comments. *(Corresponding author: Norbert Haala.)*

Wei Zhang, David Skuddis, and Norbert Haala are with the Institute for Photogrammetry and Geoinformatics, University of Stuttgart, 70174 Stuttgart, Germany (e-mail: wei.zhang@ifp.uni-stuttgart.de; david.skuddis@ifp.uni-stuttgart.de; norbert.haala@ifp.uni-stuttgart.de).

Qing Cheng is with the Technical University of Munich, 80333 Munich, Germany.

Niclas Zeller is with the Karlsruhe University of Applied Sciences, 76133 Karlsruhe, Germany.

Daniel Cremers is with the Technical University of Munich, 80333 Munich, Germany, and also with the Munich Center for Machine Learning, 80333 Munich, Germany.

The project page and source code are available at https://hi-slam2.github.io/.

This article has supplementary downloadable material available at https://doi.org/10.1109/TRO.2025.3626627, provided by the authors.

Digital Object Identifier 10.1109/TRO.2025.3626627

1941-0468 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

©2026 IEEE

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

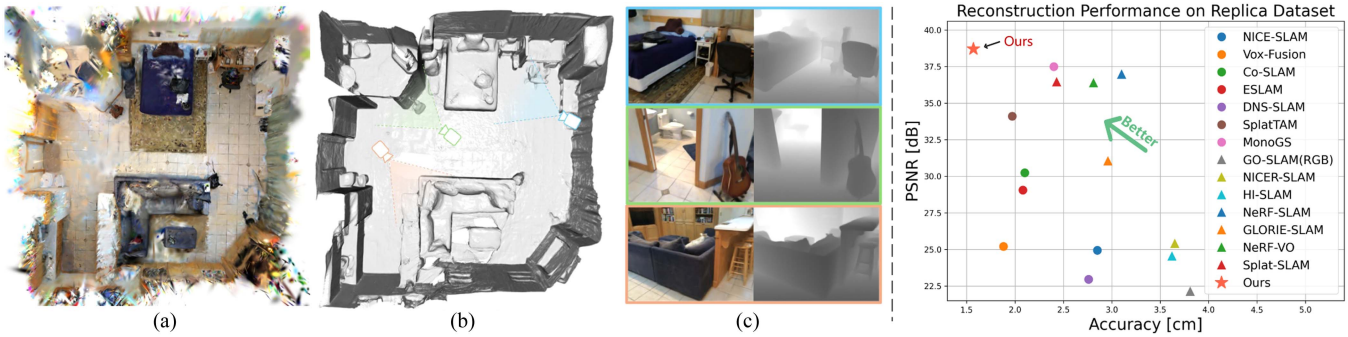

Fig. 1. Our method builds a 3-D Gaussian Splatting (3DGS) map (a) to reconstruct complex scenes using only monocular input. We are able to extract accurate and detailed mesh reconstructions (b) with high-quality renderings (c). The right figure illustrates the tradeoff between geometric accuracy and visual appearance, as some methods prioritize one aspect over the other. Compared to existing methods, our approach excels among RGB-only methods, denoted by ▲, and also surpasses recent RGB-D methods, denoted by •, in both geometry and appearance reconstruction. (a) 3DGS Map. (b) Mesh. (c) Renders.

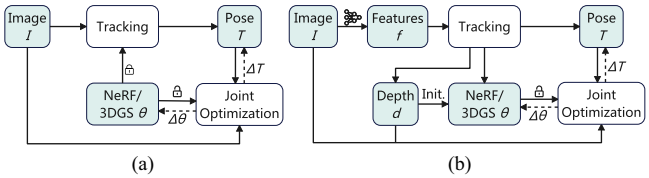

Fig. 2. Comparison of SLAM paradigms: while map-centric SLAM employs a unified map representation for both tracking and joint optimization, the hybrid design approach utilizes learning-based features and bundle adjustment (BA) for tracking, producing depth as an intermediate scene representation. This is then used to initialize the 3-D map and supervise the joint optimization of camera poses and scene geometry. (a) Map-centric e.g. MonoGS. (b) Hybrid design.

and accurate scene modeling. As depicted in Fig. 2, unlike map-centric SLAM methods, we adopt a hybrid approach that utilizes learning-based dense SLAM to generate depth as a proxy, which serves both to initialize scene geometry and to guide map optimization. This hybrid design decouples the map training from tracking while seamlessly recoupling pose and map later during joint optimization, ensuring both efficiency and accuracy.

For depth estimation, we introduce a scale-grid based alignment strategy that effectively addresses scale distortions in monocular depth priors, significantly improving depth estimation accuracy. Our surface depth rendering employs unbiased depth calculation at ray-Gaussian intersection points [34], enabling more precise surface fitting. To enhance surface reconstruction, particularly in low-texture regions, we incorporate monocular normal priors into 3DGS training, ensuring the consistency of reconstructed surfaces. By deforming 3-D Gaussian units using keyframe pose updates, we enable efficient online map updates, boosting both speed and flexibility in mapping. Furthermore, unlike hash grid-based methods [35], [36] that require a predefined scene boundary, our approach allows the map to grow incrementally as new areas are explored without any prior knowledge of scene size.

We validate our approach through extensive experiments on both synthetic and real-world datasets, including Replica [37], ScanNet [38], Waymo Open [39], ETH3D SLAM [40], and ScanNet++ [41]. Our method achieves substantial improvements in both reconstruction and rendering quality compared to existing Neural SLAM methods, surpassing even RGB-D-based methods in accuracy. Our method is particularly well-suited for real-time applications that demand rapid and reliable scene reconstruction in scenarios where depth sensors are impractical.

In summary, our work advances the state-of-the-art in dense monocular SLAM through the following contributions.

1) A geometry-aware Gaussian SLAM framework achieving high-fidelity RGB-only reconstruction through efficient online mapping and joint optimization of camera poses and Gaussian map.
2) An enhanced depth estimation approach leveraging geometry priors and improved scale alignment to compensate for monocular prior distortions and enable accurate surface reconstruction.
3) A balanced system achieving superior performance in both geometry and appearance reconstruction across synthetic and real-world datasets.

II. Related Works

A. Depth Estimation

Depth estimation can be broadly categorized into multiview and monocular approaches. Classic multiview methods rely on geometry principles, utilizing techniques such as patch matching [42] or cost aggregation [43]. Recent learning-based approaches MVSNet [23] and DeepMVS [44] have greatly improved the consistency of depth estimation across video sequences. In parallel, monocular depth estimation has seen remarkable progress, with methods like MiDaS [45] and OmniData [24] demonstrating impressive generalization across diverse datasets. However, these monocular approaches suffer from scale ambiguity, producing depth maps with inconsistent scales between frames. Our work addresses this limitation through a novel scale-grid alignment strategy that estimates spatially varying depth scales, enabling more accurate depth estimation compared to previous method [36] that relied on a single, rigid scale transformation.

B. Surface Reconstruction

Surface reconstruction typically follows a two-stage pipeline: camera pose estimation through Structure-from-Motion (SfM) [42], [46], followed by multiview stereo [47],

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

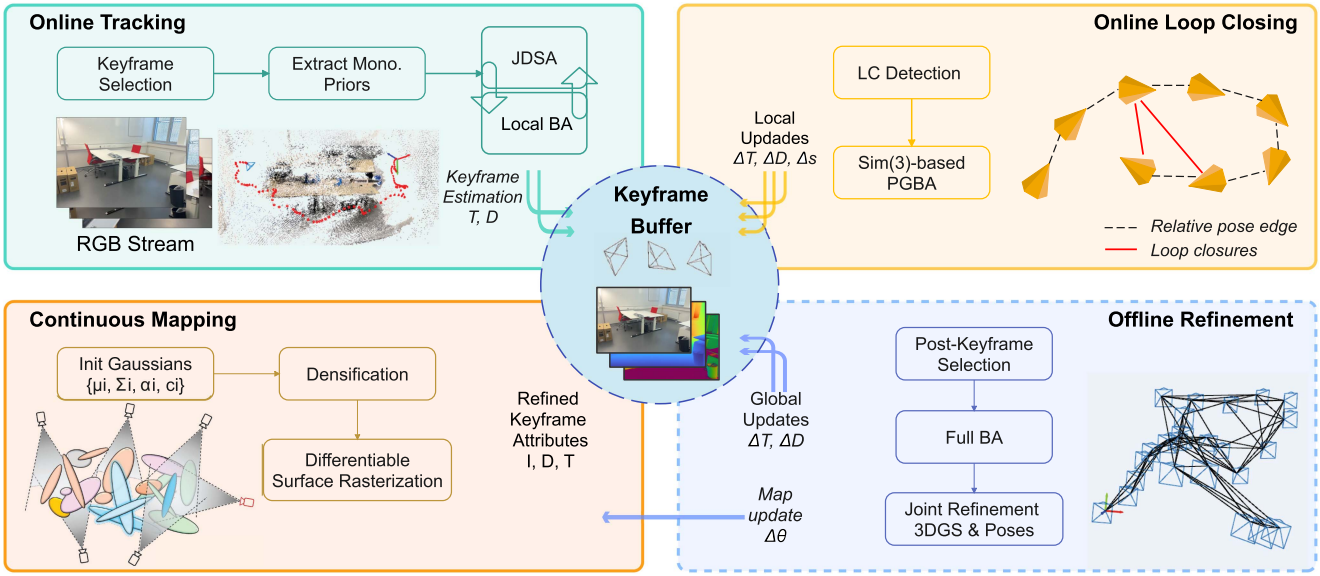

Fig. 3. *System Overview*: Our framework consists of four key stages: online camera tracking, online loop closing, online mapping, continuous mapping, and offline refinement. The camera tracking is performed using a recurrent-network-based approach to estimate camera poses $\mathbf{T}$ and generate depth maps $\mathbf{D}$ from RGB input. Depth priors are incorporated into the tracking process through our proposed joint depth and scale alignment (JDSA) strategy improving depth estimation accuracy. For 3-D scene representation, we use 3DGS to model scene geometry, enabling efficient online map updates. These updates are integrated with $Sim(3)$-based pose graph BA for online loop closing, allowing for scale drift correction via scale updates $\Delta\mathbf{s}$, and achieving both fast updates and high-quality rendering. In the offline refinement stage, camera poses and scene geometry undergo full BA, followed by joint optimization of Gaussian primitives and camera poses to further enhance global consistency.

[48] for dense reconstruction. While widely adopted, these methods are computationally intensive and often produce incomplete reconstructions due to depth estimation uncertainties. Neural implicit representations [49] and their variants [50], [51], [52] have demonstrated high-quality reconstruction capabilities but remain computationally demanding. Recent advances in 3DGS [53] offer more efficient rendering compared to NeRF-based approaches, and its variants [34], [54] show promising geometry reconstruction capabilities. Our approach leverages 3DGS for efficient scene representation while maintaining high-quality reconstruction, effectively addressing the speed-quality tradeoff inherent in previous methods.

C. Dense Visual SLAM

Dense SLAM methods traditionally relied on volumetric representations such as truncated signed distance functions (TSDF) [55] or 3-D voxel grids [56], [57] for scene geometry modeling. The emergence of neural implicit representations [49] enabled high-quality scene reconstructions within dense visual SLAM [5], [29], but at significant computational cost and often requiring RGB-D input. Recent monocular approaches like NICER-SLAM [58] and HI-SLAM [36] have demonstrated promising results using only RGB input. The 3DGS-based methods Splat-SLAM [33] and GLORIE-SLAM [59] showcase the potential of 3DGS for real-time dense reconstruction. However, these methods still face challenges in balancing computational efficiency with reconstruction quality. Our work addresses these limitations through key innovations in depth estimation, scene consistency, and computational efficiency, achieving both high-quality geometry and appearance reconstruction in real time.

III. Methods

Our system is designed to enable fast and accurate camera tracking and scene reconstruction from monocular RGB input. As illustrated in Fig. 3, the system comprises four key components: an online tracker, an online loop closing module, a continuous mapper, and an offline refinement stage. The online camera tracker (see Section III-B) leverages a learning-based dense SLAM frontend to estimate camera poses and depth maps. Global consistency and real-time performance are achieved through the online loop closure module (see Section III-C), which combines loop closure detection with efficient pose graph bundle adjustment (PGBA). For scene representation, we employ 3DGS (see Section III-D), enabling efficient online map construction, updates, and high-quality rendering. The offline refinement stage (see Section III-E) enhances reconstruction quality through full BA and joint optimization of Gaussian map and camera poses ensures optimal global consistency. The final mesh is generated by fusing rendered depth maps through TSDF fusion.

A. Comparison to HI-SLAM

The current HI-SLAM2 system represents a significant advancement over our previous work HI-SLAM [36], with improvements across multiple aspects that enhance tracking accuracy and reconstruction quality substantially. The key improvements can be summarized as follows.

1) *Depth prior integration:* We propose a novel spatially adaptive scale-grid alignment strategy that effectively addresses nonlinear scale distortions in monocular depth priors. In contrast to HI-SLAM's single scale alignment,

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

our 2-D grid-based method with bilinear interpolation accommodates spatially varying distortions. This is achieved without introducing dependencies between depth pixels, as each depth value is treated as an independent variable. This design preserves the efficiency of solving the optimization problem using the Schur complement. The improved scale alignment enhances the accuracy of depth estimation, thereby facilitating Gaussian initialization and optimizing the map reconstruction with depth supervision.

2) *Map representation:* We replace HI-SLAM's neural implicit field representation with 3DGS, transitioning from an implicit to explicit representation. This change provides several benefits:
   a) significantly faster rendering;
   b) efficient online map updates through direct Gaussian primitive updates rather than neural network weight optimization;
   c) incremental map growth without predefined scene boundaries;
   d) enhanced geometry preservation as evidenced by the 1.54 cm improvement in reconstruction accuracy on the Replica dataset (see Table IV).
3) *Hierarchical optimization:* Our system employs a multi-stage optimization pipeline that includes online tracking with local BA, online loop closure with $Sim(3)$-based PGBA, and global full BA. Last but not least, the joint refinement of both Gaussian map parameters and camera poses, which couples the tracking frontend with the mapping backend more tightly. In contrast, HI-SLAM only performs pose optimization in the tracking frontend, which can lead to inconsistencies between estimated poses and map geometry. This hierarchical approach reduces the absolute trajectory error (ATE) by 29.3% compared to online tracking alone (see Table X) on the Replica dataset, yielding a globally more consistent reconstruction.

B. Online Tracking

Our online tracking module builds upon a learning-based dense visual SLAM method [20] to estimate camera poses and depth maps of keyframes. By leveraging dense per-pixel information through a recurrent optical flow network, our system can robustly track the camera in challenging scenarios, such as low-textured environments and fast movements. To match per-pixel correspondences among all overlapping frames, we construct a keyframe graph $(\mathcal{V}, \mathcal{E})$ which represents the covisibility relationships between every pair of keyframes. The graph nodes $\mathcal{V}$ correspond to keyframes, each containing a pose $\mathbf{T} \in SE(3)$ and an estimated depth map $\mathbf{d}$. Graph edges $\mathcal{E}$ connect keyframes with sufficient overlap, determined by their optical flow correspondences. To synchronize the estimated states with other modules aiding continuous mapping and online loop closing, a keyframe buffer is maintained to store the information of all keyframes and their respective states.

The tracker begins with keyframe selection where each incoming frame is assessed to determine if it should be selected as a keyframe. This decision is based on the average flow distance relative to the last keyframe calculated through a single pass of the optical flow network [21] and a predefined threshold d_{flow}. For selected keyframe, we extract the monocular priors, including depth and normal priors, through a pretrained neural network [24]. While the depth priors are used directly by the tracker module to facilitate depth estimation, the normal priors are used by the scene representation mapper for 3-D Gaussian map optimization as extra geometry cues.

Following [20], we initialize the system state after collecting $N_{\text{init}} = 12$ keyframes. The initialization performs BA on a keyframe graph, where edges connect keyframes within an index distance of 3, ensuring sufficient overlap for reliable convergence. Since a monocular system does not have an absolute scale, we normalize the scale by setting the mean of all keyframe depths to one. This scale is then hold as the system scale by fixing the poses of the first two keyframes in subsequent BA optimizations. Afterward, each time a new keyframe is added, we perform local BA to estimate the camera poses and depth maps of the keyframes in the current keyframe graph. Edges between the new keyframe and neighboring keyframes with sufficient overlap are added to the graph. With the optical flow prediction $\mathbf{f}$, the reprojection error is minimized by using the flow-predicted target, denoted as $\check{\mathbf{p}}_{ij} = \mathbf{p}_i + \mathbf{f}$, and the current reprojection induced by camera poses and depths as source. The local BA optimization problem can be formulated as

$$\underset{\mathbf{T},\mathbf{d}}{\arg\min} \sum_{(i,j)\in\mathcal{E}} \|\check{\mathbf{p}}_{ij} - \Pi(\mathbf{T}_{ij}\Pi^{-1}(\mathbf{p}_i, \mathbf{d}_i))\|^2_{\Sigma_{ij}} \tag{1}$$

where $\mathbf{T}_{ij} = \mathbf{T}_j \cdot \mathbf{T}_i^{-1}$ denotes the rigid body transformation from keyframe i to keyframe j, and $\mathbf{d}_i$ refers to the depth map of keyframe i in inverse depth parametrization, Π and Π^{-1} represent the camera projection and back-projection functions, respectively. Σ_{ij} is a weight matrix with diagonals representing the prediction confidences from the optical flow network. The confidence effectively ensures the robustness of the optimization by reducing the influence of the outliers caused by occlusions or low-texture regions. Depth estimates in underconfident regions, where the depth cannot be accurately estimated, are further refined using monocular depth priors in the subsequent step.

Incorporate monocular depth prior: To overcome the challenge of depth estimation in difficult areas such as low-textured or occluded regions, we incorporate the easy-to-obtain monocular depth priors [24] into the online tracking process. In the RGB-D mode of [20], depth observations are directly used to compute the mean squared error during BA optimization. However, we cannot directly follow the same manner because predicted monocular depth priors exhibit inconsistent scales. To address this, Zhang et al. [36] proposed estimating a depth scale and an offset for each depth prior as optimization parameters. Although this approach helps align an overall prior scale, we found that it is not sufficient to fully correct the scale distortions inherent in monocular depth priors.

To further improve this, we propose estimating a 2-D depth scale grid with coefficients $\mathbf{s}_i$ of dimension (m, n) for each depth prior $\check{\mathbf{d}}_i$. The depth scale at every pixel can be obtained by bilinear interpolation $Bi(\mathbf{p}_i, \mathbf{s}_i)$ on the grid based on its

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

four surrounding grid coefficients. This spatially varying scale formulation makes it more flexible to align the prior depth with the estimated depth by BA and helps to reduce the influence of noise in the depth prior. Using the sampled depth scales, the scale-aligned depth prior can be obtained as $\check{\mathbf{d}}_i \cdot Bi(\mathbf{p}_i, \mathbf{s}_i)$. Then we formulate the depth prior factor r_d as follows:

$$r_d = \|\check{\mathbf{d}}_i \cdot Bi(\mathbf{p}_i, \mathbf{s}_i) - \mathbf{d}_i\|^2. \tag{2}$$

The grid resolution is set to 2×2. Higher resolutions may introduce instability, particularly in low-texture regions where optical flow predictions have low confidence, resulting in insufficient weighting in the reprojection error term for each grid tile. The scale coefficients are initially set to ones. After the system converges, the scale coefficients for new depth priors are initialized using those from the previous depth priors.

As reported in [36], directly incorporating the depth prior factor into the BA optimization, i.e., jointly optimizing the camera poses, depths, and scale coefficients, can make the system prone to scale drift and hinder convergence. To address this, similar to the approach in [36], we introduce a JDSA module to estimate the prior scales separately with the following objective:

$$\begin{aligned}\underset{\mathbf{s},\mathbf{d}}{\operatorname{argmin}} \sum_{(i,j)\in\mathcal{E}} \|\check{\mathbf{p}}_{ij} - \Pi(\mathbf{T}_{ij}\Pi^{-1}(\mathbf{p}_i, \mathbf{d}_i))\|^2_{\Sigma_{ij}} \\ + \sum_{i\in\mathcal{V}} \|\check{\mathbf{d}}_i \cdot Bi(\mathbf{p}_i, \mathbf{s}_i) - \mathbf{d}_i\|^2. \end{aligned} \tag{3}$$

By interleaving the JDSA optimization with the local BA optimization, we ensure that the system scale remains stable and the depth prior is well-aligned, providing depth estimation with a better initial guess. We use the damped Gauss–Newton algorithm to solve the optimization problem. For the sake of the optimization efficiently, we separate scale and depth variables as follows:

$$\begin{pmatrix} \mathbf{B} & \mathbf{E} \\ \mathbf{E}^T & \mathbf{C} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{s} \\ \Delta\mathbf{d} \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ \mathbf{w} \end{pmatrix} \tag{4}$$

where $\mathbf{B}$, $\mathbf{E}$, $\mathbf{C}$ are the blocks of the Hessian matrix and $\mathbf{v}$, and $\mathbf{w}$ are the gradient vector of the linearized system. Since the dimension of matrix $\mathbf{B}$ is much smaller than $\mathbf{C}$, we can solve the system efficiently by first solving for $\Delta\mathbf{s}$ and then $\Delta\mathbf{d}$ using the Schur complement

$$\begin{aligned} \Delta\mathbf{s} &= (\mathbf{B} - \mathbf{E}\mathbf{C}^{-1}\mathbf{E}^T)^{-1}(\mathbf{v} - \mathbf{E}\mathbf{C}^{-1}\mathbf{w}) \\ \Delta\mathbf{d} &= \mathbf{C}^{-1}(\mathbf{w} - \mathbf{E}^T\Delta\mathbf{s}) \end{aligned} \tag{5}$$

matrix $\mathbf{C}$ is diagonal since the scale alignment in (2) is applied to the depth prior rather than the depth variables. This preserves the independence between the depth variables allowing us to invert $\mathbf{C}$ efficiently as $\mathbf{C}^{-1} = \text{diag}\left(\frac{1}{c_1}, .., \frac{1}{c_n}\right)$. Fig. 4 shows an example of the scale alignment for monocular depth priors. Note that the estimated spatially varying scales result in well-aligned depth prior with respect to the ground truth depth.

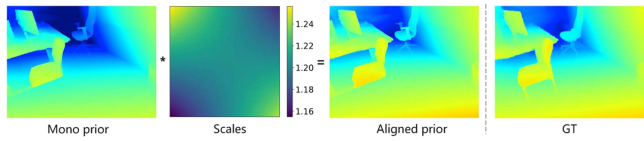

Fig. 4. Example of scale alignment of monocular depth.

C. Online Loop Closing

While our online tracker can robustly estimate camera poses, measurement noise inevitably accumulates over time and travel distance, which leads to pose drift. In addition, monocular systems are prone to scale drift due to inherent scale unobservability. To correct both pose and scale drifts and enhance the global consistency of the 3-D map, our online loop closing module searches for potential loop closures and performs global optimization on the entire history of keyframes using a $Sim(3)$-based PGBA first proposed in [36].

Loop closure detection: Loop closure detection is performed in parallel to the online tracking. For each selected new keyframe, we calculate the optical flow distances d_{of} between the new keyframe and all previous keyframes. We define three criteria to select loop closure candidates. First, d_{of} must fall below a predefined threshold τ_{flow}, ensuring sufficient covisibility for reliable convergence in recurrent flow updates. Second, orientation differences based on current pose estimations should remain within a threshold τ_{ori}. Finally, the frame index difference must exceed a minimum threshold τ_{temp} beyond the current local BA window. When all criteria are met, we add edges connecting the keyframe pairs in forward and revert reprojection directions in our keyframe graph.

$Sim(3)$-Based PGBA: When loop closure candidates are identified, inspired by the efficiency of PGBA in [36] and [60], we choose pose graph BA over full BA to balance computational efficiency with accuracy. To address scale drift, we adopt $Sim(3)$ representations for keyframe poses, enabling per-keyframe scale correction as proposed in [61]. Before each optimization run, we convert the latest pose estimates from $SE(3)$ to $Sim(3)$ group and initialize scales with ones. The pixel warping step follows (1), with the $SE(3)$ transformation replaced by a $Sim(3)$ transformation.

Constructing the pose graph involves connecting poses through relative pose edges. Following [36], [60], we derive relative poses from dense correspondences of inactive reprojection edges which are retained when their associated keyframes leave the sliding window of local BA. These dense correspondences offer a reliable basis for computing relative poses because they have been refined for multiple iterations when they are active in the sliding window. The reprojection error term from (1) is used, but here the optimization focuses solely on relative poses $\check{\mathbf{T}}_{ij}$ under the assumption that depth estimates are accurate. To incorporate uncertainty, we estimate variances Σ_{ij}^{rel} for the relative poses based on the adjustment theory [62] as

$$\Sigma_{ij}^{rel} = (\mathbf{J}\Delta\mathbf{T}_{\mathbf{ij}} - \mathbf{r})^T\Sigma_{ij}(\mathbf{J}\Delta\mathbf{T}_{\mathbf{ij}} - \mathbf{r})(\mathbf{J}^T\Sigma_{ij}\mathbf{J})^{-1} \tag{6}$$

where $\mathbf{J}$, $\mathbf{r}$, and $\Delta\mathbf{T}_{\mathbf{ij}}$ are the Jacobian, reprojection residuals, and relative pose update from the previous iteration, respectively.

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

These variances serve as weights in PGBA. The final objective of PGBA is to minimize the sum of relative pose factors and reprojection factors

$$\arg\min_{\mathbf{T},\mathbf{d}} \sum_{(i,j)\in\mathcal{E}^*} \|\check{\mathbf{p}}_{ij} - \Pi(\mathbf{T}_{ij}\Pi^{-1}(\mathbf{p}_i,\mathbf{d}_i))\|^2_{\Sigma_{ij}} + \sum_{(i,j)\in\mathcal{E}^+} \|\log(\check{\mathbf{T}}_{ij}\cdot\mathbf{T}_i\cdot\mathbf{T}_j^{-1})\|^2_{\Sigma^{rel}_{ij}} \tag{7}$$

where $\mathcal{E}^*$ represents detected loop closures, and $\mathcal{E}^+$ denotes the set of relative pose factors. To ensure the convergence of the optimization and account for potential outliers in relative pose factors, we apply a damped version of Gauss–Newton algorithm to find the optimal solution as follows:

$$\mathbf{H} = \mathbf{H} + \epsilon\cdot\mathbf{I} + \lambda\cdot\mathbf{H} \tag{8}$$

where $\mathbf{H}$ denotes the Hessian matrix. The damping factor $\epsilon = 10^{-4}$ and regularization factor $\lambda = 10^{-1}$ serve two critical functions: preventing convergence to local minima and improving numerical conditioning, while maintaining rapid convergence. Following optimization, we convert the optimized poses back to the $SE(3)$ for subsequent tracking. The depth maps are scaled according to the corresponding $Sim(3)$ pose transformations. In addition, as detailed in Section III-D, we update the 3-D Gaussian primitives based on the pose updates of their anchor keyframes.

D. 3-D Scene Representation

We adopt 3DGS [53] as our scene representation modeling scene appearance and geometry. Unlike implicit neural representations such as NeRF, 3DGS provides an explicit representation that enables efficient online map updates and high-quality rendering. The scene is represented by a set of 3-D anisotropic Gaussians $\mathcal{G} = \{g_i\}_{i=1}^M$, where each 3-D Gaussian unit is defined as

$$g_i(\mathbf{x}) = e^{-(\mathbf{x}-\boldsymbol{\mu}_i)^\top\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)} \tag{9}$$

where $\boldsymbol{\mu}_i \in \mathbb{R}^3$ denotes the Gaussian mean and $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$ represents the covariance matrix in world coordinates. The covariance matrix $\boldsymbol{\Sigma}_i$ is decomposed into orientation $\mathbf{R}_i$ and scale $\mathbf{S}_i = \text{diag}\{s_i\} \in \mathbb{R}^{3\times3}$, such that $\boldsymbol{\Sigma}_i = \mathbf{R}_i\mathbf{S}_i\mathbf{S}_i^T\mathbf{R}_i^T$. Each Gaussian also carries attributes for opacity $o_i \in [0,1]$ and color $\mathbf{c}_i \in \mathbb{R}^3$. Unlike the original 3DGS [53], we simplify the color representation by using direct RGB values instead of spherical harmonics, reducing optimization complexity. To handle view-dependent color variations, we employ exposure compensation during the offline refinement stage (see Section III-E).

The rendering process projects these 3-D Gaussians onto the image plane using perspective transformation

$$\boldsymbol{\mu}_i' = \pi(\mathbf{T}_i\cdot\boldsymbol{\mu}_i),$$
$$\boldsymbol{\Sigma}_i' = \mathbf{J}\mathbf{W}\boldsymbol{\Sigma}_i\mathbf{W}^T\mathbf{J}^T \tag{10}$$

where $\mathbf{J}$ represents the Jacobian of the perspective transformation and $\mathbf{W}$ denotes the rotation matrix of keyframe pose $\mathbf{T}_i$. After depth-based sorting of the projected 2-D Gaussians, pixel colors and depths are computed through α-blending along each ray from near to far

$$\hat{C} = \sum_{i\in\mathcal{N}} c_i\alpha_i\prod_{j=1}^{i-1}(1-\alpha_j)$$
$$\hat{D} = \sum_{i\in\mathcal{N}} d_i\alpha_i\prod_{j=1}^{i-1}(1-\alpha_j) \tag{11}$$

where $\mathcal{N}$ represents the set of Gaussians intersecting the ray, c_i is the color of the ith Gaussian, and α_i represents the pixel translucency calculated by evaluating the opacity of ith Gaussian at the intersection point.

Unbiased depth: Previous works [31], [32] directly use the depth at the Gaussian mean, which introduces estimation biases when rays intersect the Gaussian at points distant from its mean. Following [34], we compute an unbiased depth by determining the actual ray-Gaussian intersection point along the ray direction. This depth is calculated by solving the planar equation at the intersection of the ray and Gaussian surface. Since all rays from the same viewpoint that intersect a given Gaussian are coplanar, the intersection equation needs to be solved only once per Gaussian. This approach maintains the computational efficiency of splat-based rasterization while significantly improving the depth accuracy. We demonstrate the benefits of this unbiased depth computation through ablation studies in Section IV-G.

Map update: The map update process adjusts the 3-D Gaussian units based on the updates of keyframe pose to ensure global consistency of the 3-D map. This update happens both online during the $Sim(3)$-based PGBA and offline during the global full BA. To enable rapid and flexible updates to the 3-D scene representation, we deform the mean, orientation, and scale of each Gaussian unit. Specifically, means and orientations are transformed according to the relative $SE(3)$ pose change between the previous and updated keyframes, while scales are adjusted using the scale factors derived from the $Sim(3)$ pose representation.

The update equations for each Gaussian unit are

$$\boldsymbol{\mu}_j' = (\mathbf{T}_i'^{-1}\cdot\mathbf{T}_i\cdot\boldsymbol{\mu}_j)/s_i,$$
$$\mathbf{R}_j' = \mathbf{R}_i'^{-1}\cdot\mathbf{R}_i\cdot\mathbf{R}_j,$$
$$s_j' = s_i\cdot s_j \tag{12}$$

where $\boldsymbol{\mu}_j'$, $\mathbf{R}_j'$, and s_j' represent the updated mean, orientation, and scale of the jth Gaussian, respectively. This transformation ensures that the geometric relationships between Gaussians are preserved while accommodating the refined keyframe poses, maintaining the accuracy and completeness of the 3-D reconstruction.

Exposure compensation: Real-world captures exhibit varying exposures across different views due to illumination changes and view-dependent reflectance. These variations introduce color inconsistencies that can significantly impact reconstruction quality. Following [32], [63], we address this challenge by optimizing per-keyframe exposure parameters through a 3×4 affine transformation matrix. For a rendered image $\hat{I}$, the exposure

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

correction is formulated as

$$\hat{I}' = \mathbf{A} \cdot \hat{I} + \mathbf{b} \tag{13}$$

where $\mathbf{A}$ denotes the 3×3 color transformation matrix and $\mathbf{b}$ represents the 3×1 bias vector. During the offline refinement stage, these exposure parameters are jointly optimized alongside camera poses and scene geometry, as detailed in Section III-E.

Map management: To ensure that newly observed regions are well represented, we initialize a set of 3-D Gaussian primitives when each new keyframe is created to populate the Gaussian map. The initialization process begins by unprojecting the estimated depth map of the keyframe into 3-D space. Specifically, each pixel's depth value is back-projected to generate a 3-D point, which serves as the mean μ_i of a new Gaussian primitive. The orientation is initialized as the unit orientation. The initial scale $\mathbf{S}_i$ is set by finding the average distance of nearest three neighbors to adapt to the local point density, while opacity o_i is initialized to 0.5 to allow the optimization itself to update. Color attributes $\mathbf{c}_i$ are assigned based on the corresponding pixel's RGB value from the keyframe. To maintain map compactness and prevent redundancy, we apply random downsampling with a factor ψ. This downsampling ensures computational efficiency while preserving enough spatial coverage. To control map growth, we implement a pruning strategy that removes Gaussians with low opacity to eliminate redundant or insignificant primitives. We reset the opacity values every 500 iterations and perform interleaved densification and pruning every 150 iterations to balance map size and quality. A detailed analysis of map size evolution is presented in Section IV-H.

Optimization losses: The 3DGS representation is optimized using a combination of photometric, geometric, and regularization losses. The photometric loss $\mathcal{L}_c$ measures the L1 difference between the exposure-compensated rendered image $\hat{I}'$ and the observed image I. The depth loss $\mathcal{L}_d$ computes the L1 difference between the rendered depth $\hat{D}$ and the estimated depth $\bar{D}$ from the interleaved BA and JDSA optimization

$$\mathcal{L}_c = \sum_{k \in \mathcal{K}} |\hat{I}'_k - I_k|, \ \mathcal{L}_d = \sum_{k \in \mathcal{K}} |\hat{\mathbf{D}}_k - \bar{\mathbf{D}}_k| \tag{14}$$

where $\mathcal{K}$ denotes the keyframes in the local window during online mapping or all keyframes during offline refinement. To enhance the geometric supervision, we incorporate normal priors into the optimization. The estimated normals are derived from rendered depth maps using cross products of depth gradients along image plane axes. The normal loss $\mathcal{L}_n$ is defined as a cosine embedding loss

$$\mathcal{L}_n = \sum_{k \in \mathcal{K}} |1 - \hat{\mathbf{N}}_k^T \cdot \bar{\mathbf{N}}_k|. \tag{15}$$

To prevent artifacts due to excessively slender Gaussians, we apply a regularization term to the scale of the 3-D Gaussians

$$\mathcal{L}_s = \sum_{i \in \mathcal{G}} |\mathbf{s}_i - \bar{s}_i| \tag{16}$$

where $\bar{s}_i$ denotes the mean scale of the ith Gaussian, penalizing ellipsoid stretching. The final loss combines these terms with appropriate weights as follows:

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_d \mathcal{L}_d + \lambda_n \mathcal{L}_n + \lambda_s \mathcal{L}_s \tag{17}$$

where $\lambda_c, \lambda_d, \lambda_n$, and λ_s are the respective weights. We optimize Gaussian parameters using the Adam optimizer [64], performing ten iterations per new keyframe.

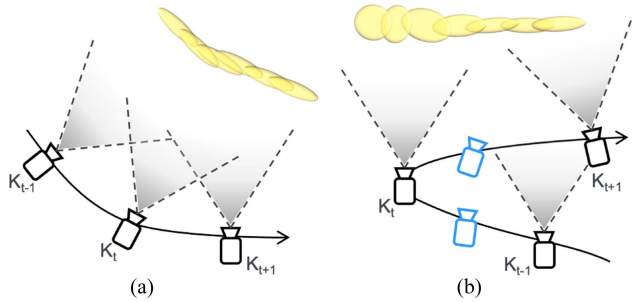

Fig. 5. View coverage analysis in two scenarios. (a) Optimal case where consecutive keyframes maintain sufficient overlap, ensuring proper multiview coverage. (b) Suboptimal case where newly observed regions in keyframe K_t lack adequate observations. Our system addresses this by inserting additional post-keyframes (shown in blue) to enhance view coverage.

E. Offline Refinement

Following the online processing, we implement three sequential offline refinement stages to enhance the global consistency and map quality: post-keyframe insertion, full BA, and joint pose and map refinement.

Postkeyframe Insertion: The first refinement stage identifies regions with insufficient view coverage, particularly areas near view frustum boundaries. These regions typically arise when forward camera motion is followed by backward rotational movement, as illustrated in Fig. 5. During online processing, keyframe selection relies on average optical flow between neighboring frames, as view coverage cannot be fully evaluated without future trajectory information. To identify under-observed regions in the offline stage, we project each keyframe's pixels onto its adjacent keyframes and quantify the percentage of pixels that fall outside the fields of view of neighboring keyframes. When this percentage exceeds a predetermined threshold, we flag the region as having insufficient observations. Additional keyframes are then inserted at these locations, and new Gaussian primitives are populated in the same manner as the keyframes inserted during the online process. This ensures more complete scene reconstruction and preserves critical details at scene boundaries.

Full BA: While our online loop-closing module achieves global consistency through efficient $Sim(3)$-based PGBA, full BA further enhances system accuracy. PGBA offers superior computational efficiency compared to full BA, but introduces minor approximation errors when abstracting dense correspondences into relative pose edges. Specifically, PGBA computes reprojection factors only for loop closure edges, while full BA performs comprehensive optimization by recomputing reprojection factors in (1) for all overlapping keyframe pairs, including both neighboring and loop closure frames. As demonstrated in Section IV-G, this improves the global consistency of camera poses and scene geometry at a finer granularity.

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

Joint pose and map refinement: The final refinement stage jointly optimizes the Gaussian map and camera poses based on the results of full BA. While the online mapping stage limits optimization iterations per keyframe to maintain real-time performance, the offline refinement enables comprehensive optimization across all keyframes. To facilitate joint pose refinement, we compute pose Jacobians during rasterization-based rendering. In addition, we also optimize per-keyframe exposure compensation parameters to ensure a better global color consistency. Unlike the full BA stage which employs the Gauss–Newton algorithm, this joint refinement step utilizes the Adam optimizer [64] with first-order gradient descent, leveraging our existing mapping pipeline.

TABLE I
COMPARISON OF CAMERA TRACKING ACCURACY FOR RGB AND RGB-D METHODS ON REPLICA DATASET WITH RESULTS IN [CM]

	Method	ro-0	ro-1	ro-2	of-0	of-1	of-2	of-3	of-4	Avg.
RGB-D input	NICE-SLAM[29]	0.97	1.31	1.07	0.88	1.00	1.06	1.10	1.13	1.07
	ESLAM[67]	0.63	0.71	0.70	0.52	0.57	0.55	0.58	0.72	0.63
	Point-SLAM[68]	0.61	0.41	0.37	0.38	0.48	0.54	0.69	0.72	0.52
	SplatTAM[31]	0.31	0.40	0.29	0.47	0.27	0.29	0.32	0.55	0.36
	MonoGS[32]	0.44	0.32	0.31	0.44	0.52	0.23	0.17	2.25	0.58
RGB input	DROID-SLAM[20]	0.34	0.13	0.27	0.25	0.42	0.32	0.52	0.40	0.33
	NICER-SLAM[58]	1.36	1.60	1.14	2.12	3.23	2.12	1.42	2.01	1.88
	GLORIE-SLAM[59]	0.31	0.37	0.20	0.29	0.28	0.45	0.45	0.44	0.35
	Splat-SLAM[31]	0.29	0.33	0.25	0.29	0.35	0.34	0.42	0.43	0.34
	MGS-SLAM[69]	0.36	0.35	0.32	0.35	0.28	0.26	0.32	0.34	0.32
	Ours	0.23	0.22	0.19	0.23	0.27	0.25	0.37	0.33	0.26

IV. EXPERIMENTS

To evaluate the performance of the proposed system, we conducted extensive experiments on several challenging datasets, including the synthetic Replica dataset [37] and the real-world datasets ScanNet [38], Waymo [39], ETH3D SLAM [40], and ScanNet++ [41]. We begin by providing implementation details and evaluation metrics, followed by quantitative and qualitative comparisons on camera tracking accuracy, geometry, and appearance reconstruction quality against state-of-the-art baselines. Subsequently, we present ablation studies to analyze the impact of different design choices. Finally, we present the runtime performance and map size analysis.

A. Implementation Details

Our system is implemented using PyTorch [65] and CUDA for GPU acceleration, with evaluations performed on an Nvidia RTX 4090 GPU and Intel Core i9-12900 K CPU. For optical flow and geometry prior prediction, we utilize pretrained models from [20] and [24], respectively. For map refinement optimization, we use 2000 iterations for the Replica dataset and 26 000 iterations for ScanNet and ScanNet++ datasets, ensuring fair comparison with existing methods. The loss weights of map optimization remain consistent across all experiments: color loss (λ_c) at 0.95, depth loss (λ_d) at 0.25, and scale loss (λ_s) at 10. The normal loss weight (λ_n) is set to 0.1 for the Replica dataset and increased to 0.5 for ScanNet and ScanNet++ datasets to enhance the geometric supervision on real-world data. The downsampling factor ψ is set to 32 across all experiments, providing an optimal balance between map size and quality.

B. Datasets

Replica Dataset [37] provides synthetic indoor scenes with high-quality reconstructions, featuring complex geometry and textures. We evaluate using eight RGB-D sequences from [5]. The sequences have perfect camera poses and reconstructions make it ideal for benchmarking dense visual SLAM methods. ScanNet Dataset [38] offers real-world RGB-D captures for 3-D scene reconstruction. Following [36], we use eight sequences for tracking evaluation and six additional sequences for geometry reconstruction assessment, using the RGB-D reconstructions as ground truth. ETH3D SLAM [40] provides a diverse set of real-world RGB-D sequences with motion-capture ground truth poses, featuring challenging conditions such as extreme lighting variations and complete darkness. Waymo Open dataset [39] provides real-world outdoor data with ground truth vehicle poses. The front-view camera images are used as input for our system. Following the evaluation protocol of [66], we evaluate the tracking accuracy and rendering quality using nine sequences. ScanNet++ [41] presents a large-scale indoor dataset with laser-scanned ground truth, enabling evaluation of dense SLAM reconstruction quality. While the dataset includes both DSLR and iPhone captures, we specifically evaluate on the iPhone sequences, which present additional challenges due to their lower image quality.

C. Evaluation Metrics

We evaluate our system's performance across three key aspects. Camera tracking accuracy is quantified using ATE, measuring the precision of estimated camera poses. For geometry reconstruction quality, we adopt three metrics from [5]: average accuracy [cm], average completeness [cm], and completeness ratio [%] (representing the percentage of reconstruction within 5 cm of ground truth). For appearance quality assessment, we evaluate keyframe renderings using standard photometric metrics: peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and learned perceptual image patch similarity (LPIPS). In all result tables, we highlight performance rankings using: first, second, and third.

D. Camera Tracking Accuracy

We evaluate the camera tracking accuracy of our system against state-of-the-art dense visual SLAM methods on indoor Replica, ScanNet, ETH3D SLAM datasets and outdoor Waymo datasets, including comparisons with RGB-D based approaches. The ATE results in Tables I and II demonstrate the superior tracking accuracy of our system. RGB-D methods SplatTAM [31] and MonoGS [32], despite having access to depth measurements for map-based tracking, achieve lower accuracy than hybrid approaches. Our system, along with Splat-SLAM [33], represents the class of hybrid methods that effectively combine dense SLAM with deep learning foundations. The global BA of DROID-SLAM [20] was enabled in all experiments. While it employs global BA, our additional global pose and map

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

TABLE II
CAMERA TRACKING ACCURACY FOR RGB AND RGB-D METHODS ON SCANNET DATASET WITH RESULTS IN [CM]

	Method	0000	0054	0059	0106	0169	0181	0207	0233	Avg.
RGB-D input	NICE-SLAM[29]	12.00	20.90	14.00	7.90	10.90	13.40	6.20	9.00	11.8
	ESLAM[67]	7.30	36.30	8.50	7.50	6.50	9.00	5.70	4.30	10.6
	Co-SLAM[35]	7.10	12.80	11.10	9.40	5.90	11.80	7.10	6.10	8.90
	Point-SLAM[68]	10.20	28.00	7.80	8.70	22.20	14.80	9.50	6.10	14.3
	LoopSplat[70]	4.20	7.50	7.50	8.30	7.50	10.60	7.90	5.20	7.70
RGB input	GO-SLAM[71]	5.90	13.30	8.30	8.10	8.40	8.30	6.90	5.30	8.10
	GLORIE-SLAM[59]	5.50	9.40	9.10	7.00	8.20	8.30	7.50	5.10	7.50
	Splat-SLAM[31]	5.57	9.50	9.11	7.09	8.26	8.39	7.53	5.17	7.58
	HI-SLAM[36]	6.43	9.97	7.22	6.56	8.53	7.65	8.43	5.23	7.47
	Ours	5.82	8.64	7.30	6.80	8.25	7.41	7.40	4.93	7.07

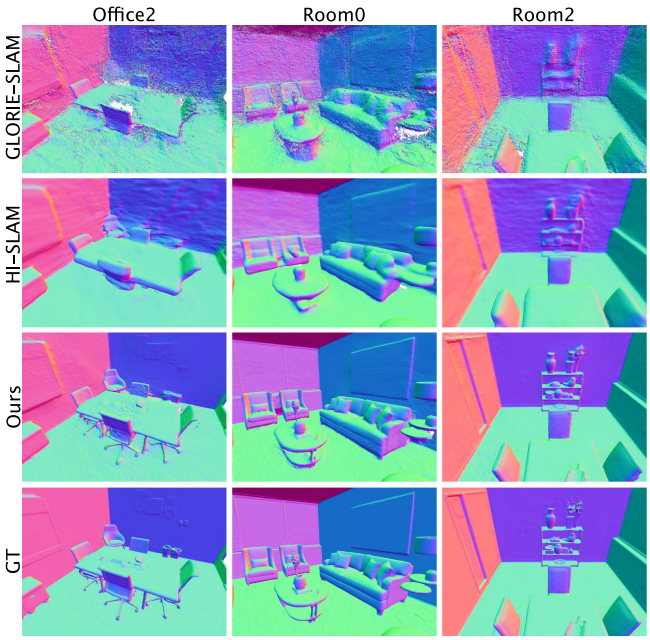

Fig. 6. Qualitative comparison on geometry reconstruction on Replica dataset.

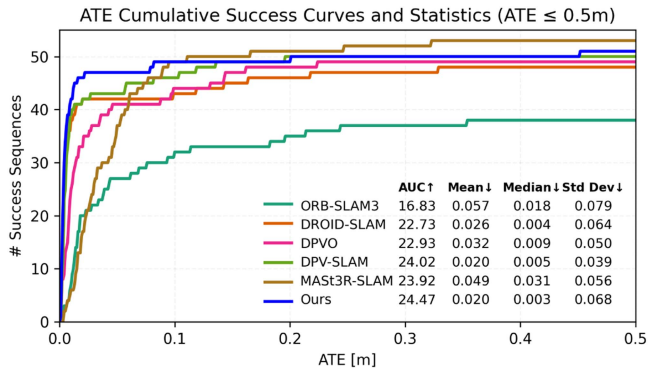

Fig. 7. Analysis of successful sequences relative to ATE error thresholds on the ETH3D SLAM dataset.

joint refinement further improves tracking accuracy beyond the baseline method. We further compare our system on the ETH3D SLAM dataset with state-of-the-art sparse and dense methods, including ORB-SLAM3 [17], DPVO [73], DPV-SLAM [74], and MASt3R-SLAM [75]. Fig. 7 illustrates the cumulative success curves based on ATE thresholds and ATE statistics. As none of the methods can successfully track all sequences, we use the area under the curve (AUC) metric to assess both accuracy and robustness with an upper ATE threshold of 0.5 m. Our system achieves the highest AUC among all methods and lowest mean and median ATE. Out of total 61 sequences, 6 sequences failed due to complete darkness, while 4 sequences encountered tracking failures caused by lighting changes and view occlusions. These limitations could potentially be addressed in the future by integrating place recognition for relocalization to reduce the standard deviation of ATE. Our evaluation on the Waymo Open dataset (Table III) further validates our approach, where we achieve the lowest ATE among all competing methods. This highlights the ability of our system to generalize effectively to challenging large-scale outdoor environments with complex scene geometries that typically pose substantial difficulties for monocular systems.

TABLE III
CAMERA TRACKING AND RENDERING RESULTS ON WAYMO OPEN DATASET AVERAGED OVER NINE SEQUENCES

Metrics	NICER-SLAM[58]	GLORIE-SLAM[59]	Photo-SLAM[72]	MonoGS[32]	OpenGS-SLAM[66]	Ours
ATE [m] ↓	19.59	0.536	19.95	8.529	0.839	0.457
PSNR ↑	12.22	18.83	17.73	21.80	23.99	28.99
SSIM ↑	0.622	0.702	0.741	0.780	0.800	0.872
LPIPS ↓	0.726	0.572	0.674	0.577	0.434	0.219

TABLE IV
RECONSTRUCTION EVALUATION ON REPLICA DATASET FOR IMPLICIT AND EXPLICIT RGB METHODS. OURS SURPASS OTHER METHODS ESPECIALLY LARGE MARGIN IN ACCURACY

	Method	Metric	ro-0	ro-1	ro-2	of-0	of-1	of-2	of-3	of-4	Avg.
NeRF-based	NICER-SLAM[58]	Acc.[cm] ↓	2.53	3.93	3.40	5.49	3.45	4.02	3.34	3.03	3.65
		Comp.[cm] ↓	3.04	4.10	3.42	6.09	4.42	4.29	4.03	3.87	4.16
		Comp.Rat[%]↑	88.75	76.61	86.10	65.19	77.84	74.51	82.01	83.98	79.37
	GO-SLAM[71]	Acc.[cm] ↓	4.60	3.31	3.97	3.05	2.74	4.61	4.32	3.91	3.81
		Comp.[cm] ↓	5.56	3.48	6.90	3.31	3.46	5.16	5.40	5.01	4.79
		Comp.Rat[%]↑	73.35	82.86	74.23	82.56	86.19	75.76	72.63	76.61	78.00
	HI-SLAM[36]	Acc.[cm] ↓	3.21	3.74	3.16	3.87	2.60	4.62	4.25	3.53	3.62
		Comp.[cm] ↓	3.25	3.08	4.09	5.29	8.83	4.42	4.06	3.72	4.59
		Comp.Rat[%]↑	86.99	87.19	80.82	72.55	72.44	80.90	81.04	82.88	80.60
3DGS-based	GLORIE SLAM[59]	Acc.[cm] ↓	2.84	3.07	3.05	2.98	2.06	3.32	3.34	2.92	2.96
		Comp.[cm] ↓	4.65	3.55	3.64	2.39	3.43	4.54	4.57	4.78	3.95
		Comp.Rat[%]↑	81.96	85.78	84.50	88.82	85.07	82.09	80.41	81.04	83.72
	Splat-SLAM[33]	Acc.[cm] ↓	1.99	1.91	2.06	3.96	2.03	3.45	2.15	1.89	2.43
		Comp.[cm] ↓	3.78	3.38	3.34	2.75	3.33	4.36	3.96	4.25	3.64
		Comp.Rat[%]↑	85.47	86.88	86.12	87.32	85.17	81.37	82.25	82.95	84.69
	Ours	Acc.[cm] ↓	1.35	1.40	1.87	1.40	1.18	1.94	1.70	1.70	1.57
		Comp.[cm] ↓	3.33	3.27	3.66	2.07	3.23	4.29	3.84	4.26	3.49
		Comp.Rat[%]↑	87.45	85.91	86.13	89.41	85.63	81.73	82.52	83.23	85.25

E. Geometry Reconstruction Quality

In Table IV, we evaluate our geometry reconstruction results against recent NeRF-based and 3DGS-based methods on the Replica dataset, demonstrating superior performance in both accuracy and completeness metrics. As illustrated in Fig. 6, our method produces smoother reconstructions while preserving fine geometric details compared to GLORIE-SLAM [59] and HI-SLAM [36]. This is particularly evident in complex scene elements such as chair legs and shelf-mounted vases, where our results more closely match the ground truth. Qualitative comparisons on the ScanNet dataset (see Fig. 8) further highlight

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

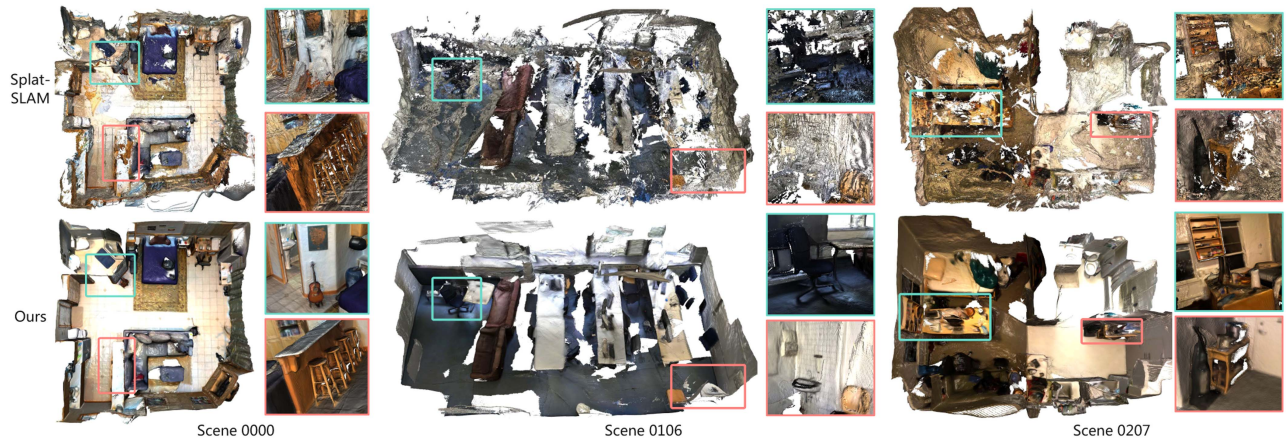

Fig. 8. Qualitative comparison on geometry and appearance reconstruction on ScanNet dataset.

Fig. 9. Rendering quality comparison on the Waymo Open Dataset in unbounded outdoor scenes. Our method captures finer details of the driving environment, while other methods produce noticeably blurrier results.

TABLE V
RENDERING QUALITY EVALUATIONS ON REPLICA DATASET FOR RGB AND RGB-D METHODS

	Method	Metric	ro-0	ro-1	ro-2	of-0	of-1	of-2	of-3	of-4	Avg.
RGB-D input	Point-SLAM[68]	PSNR ↑	32.40	34.08	35.50	38.26	39.16	33.99	33.48	33.49	35.17
		SSIM ↑	0.97	0.98	0.98	0.98	0.99	0.96	0.96	0.98	0.98
		LPIPS ↓	0.11	0.12	0.11	0.10	0.12	0.16	0.13	0.14	0.12
	Splat TAM[31]	PSNR ↑	32.86	33.89	35.25	38.26	39.17	31.97	29.70	31.81	34.11
		SSIM ↑	0.98	0.97	0.98	0.98	0.98	0.97	0.95	0.95	0.97
		LPIPS ↓	0.07	0.10	0.08	0.09	0.09	0.10	0.12	0.15	0.10
	Mono GS[32]	PSNR ↑	34.83	36.43	37.49	39.95	42.09	36.24	36.70	36.07	37.50
		SSIM ↑	0.95	0.96	0.97	0.97	0.98	0.96	0.96	0.96	0.96
		LPIPS ↓	0.07	0.08	0.08	0.07	0.06	0.08	0.07	0.10	0.07
RGB input	GLORIE -SLAM[59]	PSNR ↑	28.49	30.09	29.98	35.88	37.15	28.45	28.54	29.73	31.04
		SSIM ↑	0.96	0.97	0.96	0.98	0.99	0.97	0.97	0.97	0.97
		LPIPS ↓	0.13	0.13	0.14	0.09	0.08	0.15	0.11	0.15	0.12
	Splat-SLAM[33]	PSNR ↑	32.25	34.31	35.95	40.81	40.64	35.19	35.03	37.40	36.45
		SSIM ↑	0.91	0.93	0.95	0.98	0.97	0.96	0.95	0.98	0.95
		LPIPS ↓	0.10	0.09	0.06	0.05	0.05	0.07	0.06	0.04	0.06
	Ours	PSNR ↑	35.48	36.93	38.53	42.28	43.16	37.31	36.99	38.95	38.71
		SSIM ↑	0.96	0.97	0.97	0.98	0.98	0.97	0.97	0.97	0.97
		LPIPS ↓	0.04	0.04	0.03	0.02	0.03	0.04	0.04	0.04	0.03

our advantages over Splat-SLAM [33], showing more accurate geometry without floating artifacts and achieving better completeness. Additional qualitative results on the ScanNet++ dataset (see Fig. 10) demonstrate our system's capability to fully reconstruct challenging scenes, including low-texture surfaces like floors and walls. Notably, our reconstructions even capture glass windows that are missing in the laser scanner ground truth.

F. Appearance Reconstruction Quality

Tables III, V, and VI present our rendering quality evaluation results. On the Replica dataset, our system significantly outperforms competing methods, achieving superior PSNR and LPIPS metrics. For the ScanNet dataset, we demonstrate better performance than RGB-D methods while matching the strong baseline of Splat-SLAM [33]. As detailed in our ablation study (see Section IV-G), we could achieve even higher rendering quality by relaxing geometric constraints by normal loss, but this would compromise geometry accuracy. Instead, our system balances the tradeoff between geometry and appearance quality. On the Waymo Open dataset, Figure 9 shows qualitative comparisons of our rendered RGB images with those from other methods, and Table III presents the corresponding quantitative results. Our system achieves significantly higher rendering quality. This enhanced visual fidelity can be attributed to two key factors: first, our superior tracking accuracy; and second, the effective integration of depth and normal supervision in our mapping pipeline.

G. Ablation Study

Monocular prior integration: We first evaluate the depth estimation accuracy improved by our different modules on the Replica dataset, as depth accuracy is crucial for scene reconstruction quality. Table VII quantifies the effectiveness of different approaches. Comparing prior (one scale) with prior (scale grid) demonstrates that our grid-based scale alignment significantly outperforms the single-scale alignment from HI-SLAM [36], better addressing inherent scale distortions in

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

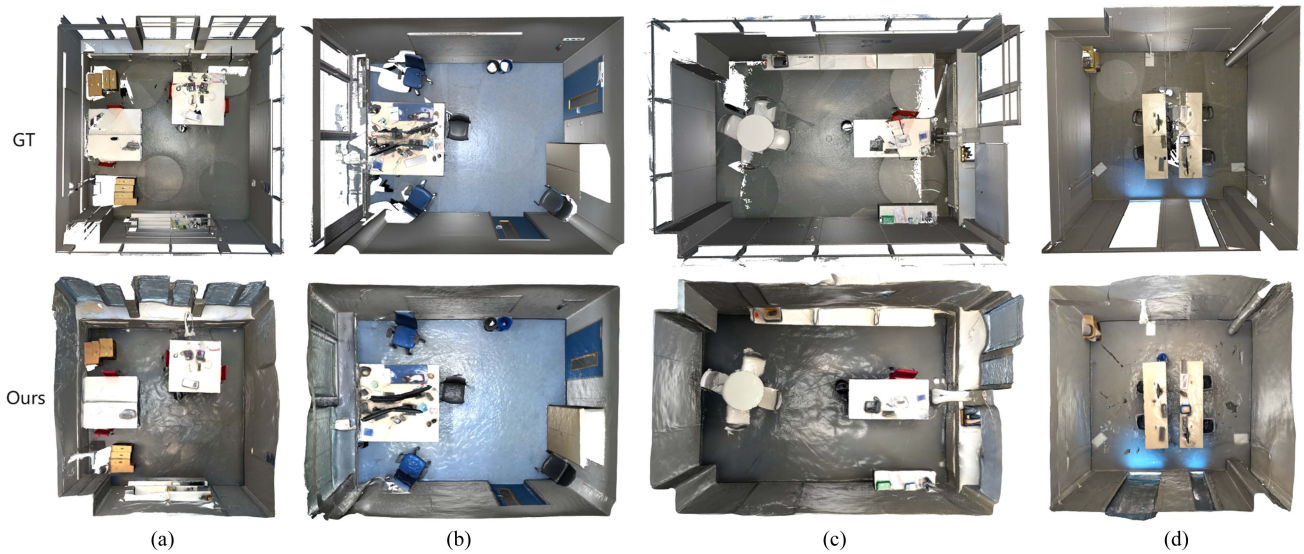

Fig. 10. Reconstructed meshes of four selected sequences on ScanNet++ Dataset. (a) f34d532901. (b) 39f36da05b. (c) 8b5caf3398. (d) b20a261fdf.

TABLE VI
RENDERING QUALITY EVALUATIONS ON SCANNET DATASET FOR RGB AND RGBD METHODS

	Method	Metric	0000	0059	0106	0169	0181	0207	Avg.
RGB-D input	Point -SLAM[68]	PSNR ↑	21.30	19.48	16.80	18.53	22.27	20.56	19.82
		SSIM ↑	0.81	0.77	0.68	0.69	0.82	0.75	0.75
		LPIPS ↓	0.48	0.50	0.54	0.54	0.47	0.54	0.51
	Splat TAM[31]	PSNR ↑	18.70	20.91	19.84	22.16	22.01	18.90	20.42
		SSIM ↑	0.71	0.79	0.81	0.78	0.82	0.75	0.78
		LPIPS ↓	0.48	0.32	0.32	0.34	0.42	0.41	0.38
	Gaussian -SLAM[76]	PSNR ↑	28.54	26.21	26.26	28.60	27.79	28.63	27.67
		SSIM ↑	0.93	0.93	0.93	0.92	0.92	0.91	0.92
		LPIPS ↓	0.27	0.21	0.22	0.23	0.28	0.29	0.25
RGB input	GLORIE -SLAM[59]	PSNR ↑	23.42	20.66	20.41	25.23	21.28	23.68	22.45
		SSIM ↑	0.87	0.83	0.84	0.91	0.76	0.85	0.84
		LPIPS ↓	0.26	0.31	0.31	0.21	0.44	0.29	0.30
	Splat-SLAM[33]	PSNR ↑	28.68	27.69	27.70	31.14	31.15	30.49	29.48
		SSIM ↑	0.83	0.87	0.86	0.87	0.84	0.84	0.85
		LPIPS ↓	0.19	0.15	0.18	0.15	0.23	0.19	0.18
	Ours	PSNR ↑	28.62	27.22	28.13	31.28	30.37	30.03	29.27
		SSIM ↑	0.85	0.87	0.90	0.90	0.90	0.86	0.88
		LPIPS ↓	0.28	0.23	0.21	0.18	0.25	0.30	0.24

TABLE VII
DEPTH ACCURACY OF OUR FINAL RENDERED DEPTH COMPARED TO THE PRIOR DEPTH ALIGNED USING DIFFERENT STRATEGIES, AS WELL AS BA DEPTH WITH AND WITHOUT JDSA ASSISTANCE, EVALUATED ON THE REPLICA DATASET

Depth Type	Abs Diff [m] ↓	Abs Rel [%] ↓	Sq Rel [%] ↓	RMSE [m] ↓	δ <1.05 [%] ↑	δ <1.25 [%] ↑
Prior(one scale)	0.147	6.70	4.62	0.18	66.69	94.85
Prior(scale grid)	0.074	3.41	0.52	0.10	77.45	99.66
BA estimate	0.059	2.86	0.37	0.09	83.52	99.74
BA with JDSA	0.046	1.99	0.51	0.11	91.84	99.32
Ours Rendered	0.015	0.67	0.10	0.04	98.65	99.81

TABLE VIII
ABLATION STUDY ON THE CHOICE OF DEPTH PRIOR MODELS, EVALUATING RENDERING QUALITY, RECONSTRUCTION ACCURACY, AND INFERENCE TIME PER FRAME ON THE REPLICA DATASET

	PSNR [dB] ↑	Acc [cm] ↓	Comp [cm] ↓	Comp Rat [%] ↑	Runtime [ms] ↓
Ours + Metric3D	38.59	1.60	3.58	85.11	61
Ours + DA V2	38.68	1.57	3.48	85.31	32
Ours + ZoeDepth	38.60	1.64	3.60	85.00	52
Ours + Omnidata	38.71	1.57	3.49	85.25	6

TABLE IX
ABLATION STUDY ON THE CHOICE OF NORMAL PRIOR MODELS

	PSNR [dB] ↑	Acc [cm] ↓	Comp [cm] ↓	Comp Rat [%] ↑	Runtime [ms] ↓
Ours + EESNU	38.60	1.70	3.64	84.61	5
Ours + DSINE	38.52	1.62	3.55	85.02	37
Ours + OmniData	38.71	1.57	3.49	85.25	6

monocular depth priors. Further analysis compares two depth estimation approaches: BA estimate (using BA alone) and BA with JDSA (using interleaved BA and JDSA optimization). The results confirm that incorporating JDSA with alternating optimization outperforms BA alone, validating our depth prior integration strategy. The final rendered depth from our Gaussian map achieves the highest accuracy, validating the effectiveness of our complete pipeline.

We further investigate the incorporation of alternative depth and normal prior predictors. For depth priors, including Metric3D [77], ZoeDepth [78], and Depth Anything (DA) V2 [79], we present results in Table VIII. Notably, ZoeDepth and DA V2 were not trained on any Replica images, yet still achieve comparable performance, indicating that our method does not rely on indirect biases from the training data. For normal priors, we evaluate EESNU [25] and DSINE [80], as shown in Table IX. The results demonstrate that our method remains compatible with these alternatives. Nevertheless, our chosen prior, OmniData, offers a more efficient tradeoff between performance and computational cost. This choice also aligns with the baseline methods, which likewise employ OmniData as the geometry prior.

Trajectory accuracy: Table X demonstrates the progressive improvement in pose estimation accuracy on the Replica

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

TABLE X
ABLATION STUDY ON THE PROGRESSIVE IMPROVEMENT IN TRAJECTORY ACCURACY ON THE REPLICA DATASET, AVERAGED OVER EIGHT SEQUENCES

	Online Tracking	Online PGBA	Offline Full BA	Joint Pose Map Refinement
ATE [cm] ↓	0.42	0.33	0.32	0.26

From left to right, each stage refines the pose estimation based on the previous stage.

TABLE XI
ABLATION STUDY ON THE IMPACT OF THE DIFFERENT PROPOSED MODULES ON THE RECONSTRUCTION PERFORMANCE ON THE REPLICA DATASET, AVERAGED OVER EIGHT SEQUENCES

	PSNR [dB] ↑	SSIM ↑	LPIPS ↓	Acc. [cm] ↓	Comp. [cm] ↓	Comp. Rat [%] ↑
w/o grid-based scale align	37.18	0.97	0.05	1.68	3.58	84.04
w/o unbiased depth render	38.23	0.97	0.04	1.73	3.92	81.16
w/o $\mathcal{L}_{normal}$	39.09	0.98	0.03	2.46	4.09	82.40
w/o joint pose map refine	37.25	0.96	0.04	1.61	3.55	84.29
Ours	38.71	0.97	0.03	1.57	3.49	85.25

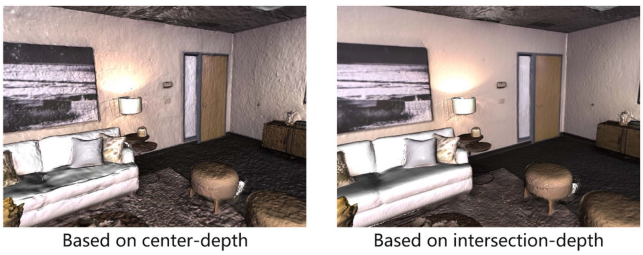

Fig. 11. Reconstruction quality comparison on the *Room0* scene of the Replica dataset, using rendered depth based on the depth at the Gaussian center versus our approach, which uses the depth at the ray-Gaussian intersection.

dataset through our system pipeline. Starting with initial estimates from the online tracking module, accuracy is first enhanced through online PGBA based loop closing. A subsequent full BA further refines these results, with the final joint pose and 3DGS map refinement achieving the highest trajectory accuracy. This systematic improvement across stages validates the effectiveness of our hierarchical optimization approach.

Component analysis: To evaluate key design components, we conduct ablation studies by removing individual components. Table XI confirms each module's contribution to system performance. The grid-based scale alignment proves crucial, as its removal significantly degrades reconstruction accuracy. Similarly, the unbiased depth rendering enhances both rendering quality and geometric accuracy, with qualitative comparison shown in Fig. 11. While the normal loss slightly affects appearance metrics, it substantially improves geometry quality. The final joint pose and map refinement further enhances accuracy through improved global consistency.

H. Performance Analysis

Our system achieves real-time performance with the online tracking, loop closing and mapping operating at 22 frames per second (FPS) on the Replica dataset, and takes only 12 s for offline refinement. On the ScanNet dataset, despite more rapid camera movements, it maintains robust performance online

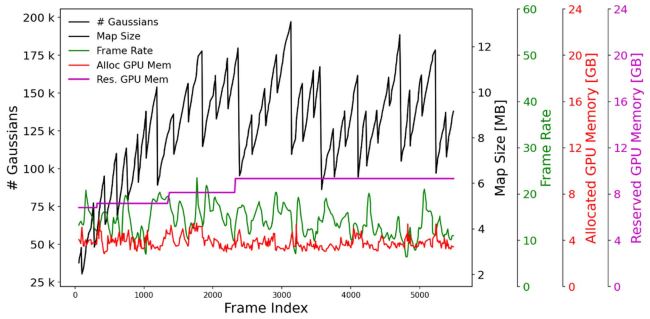

Fig. 12. Evolution of map size, GPU memory usage, and system speed over frame index for the *Scene0000* sequence from ScanNet, evaluated on an RTX 4090 GPU. The number of Gaussians grows as new areas are explored and stabilizes with the pruning strategy. System speed denotes the processing frame rate for online tracking, loop closing, and mapping. Allocated and reserved GPU memory usage are both reported to provide resource analysis.

above 10 FPS, with offline refinement taking a few minutes due to a higher number of optimization iterations. Fig. 12 illustrates the map size evolution, GPU memory consumption, and processing speed for sequence *Scene0000*. While the Gaussian count initially grows during new area exploration, our efficient map pruning strategy stabilizes the map size by eliminating redundant Gaussians in revisited regions. The memory analysis reveals that both allocated and reserved GPU memory usage remain stable throughout the sequence, with periodic fluctuations corresponding to pruning cycles. Reserved memory gradually increases as additional buffers are allocated to store keyframe states. The processing frame rate decreases during rapid motion due to more frequent keyframe insertion, and increases during stable motion, remaining consistent across different phases. This demonstrates our system's scalability and efficiency in managing large-scale environments while maintaining predictable resource consumption.

I. Evaluation on Self-Collected Robot Data

We evaluate our system on a self-collected dataset captured by our robotic platform. Fig. 13 shows the platform and the results in a large factory hall. The robot is equipped with stereo cameras. However, only the monocular input from the left camera is used for this experiment. The recorded sequence has intotal 4073 frames with a duration of 6 min and 52 s. We compare our rendering results with the baseline method DROID-SLAM + 3DGS, where the estimated camera poses and point cloud from DROID-SLAM serve as input and used to initialize the Gaussians. For our system, it completes online stage including tracking, loop closing, and mapping in 5 min and 20 s, followed by 3 min and 32 s for the offline refinement stage. In contrast, the baseline requires 4 min and 24 s for DROID-SLAM and an additional 15 min and 2 s for 3DGS mapping. Overall, our method takes only about half the runtime of the baseline while achieving comparable rendering quality and significantly better geometry, without the floaters observed in the baseline results. Furthermore, we evaluate the trajectory accuracy using the centimeter-accurate photogrammetric reference described in [81], as shown in Fig. 14. While MASt3R-SLAM [75]

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

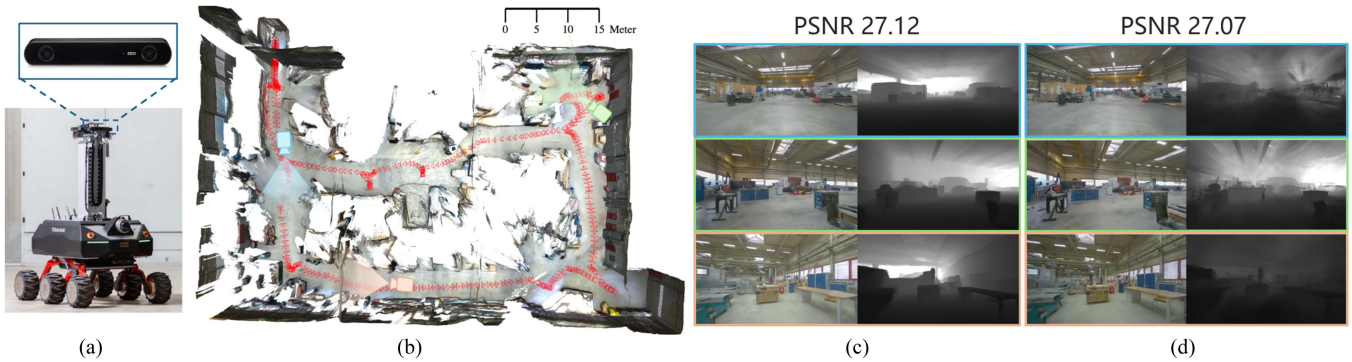

Fig. 13. Evaluation on self-collected data with the robot navigating through a large factory hall. Our robot (a) is equipped with stereo cameras. Our system operates solely on the monocular input from the left camera and reconstructs the scene along with the estimated camera trajectory (b). The rendered color and depth maps (c) achieve comparable visual quality while exhibiting significantly better geometry and free of floaters. In contrast, the baseline DROID-SLAM + 3DGS (d) suffers from geometric artifacts and floating points. We refer readers to our supplementary video, which showcases novel view renderings on out-of-sequence viewpoints. (a) Setup. (b) Reconstruction. (c) Renderings of ours. (d) DROID-SLAM + 3DGS.

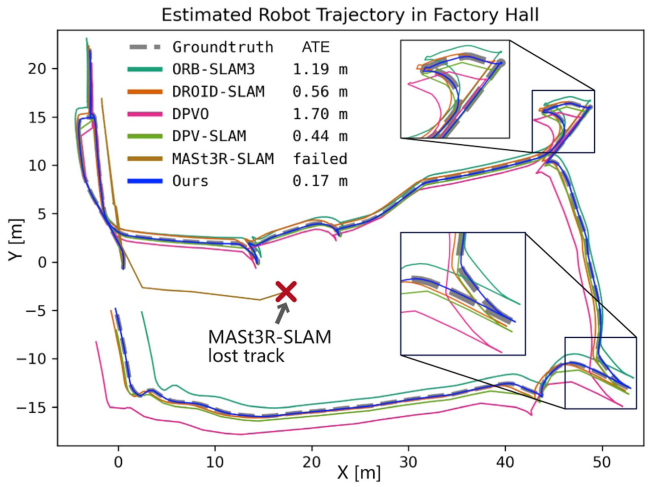

Fig. 14. Qualitative comparison of the estimated robot trajectory in the factory hall.

TABLE XII
SEMANTIC RECONSTRUCTION RESULTS EVALUATED BY MIOU METRIC ON FOUR SEQUENCES OF THE REPLICA DATASET

	Method	ro-0	ro-1	ro-2	of-0	Avg.mIoU[%]↑
RGB-D	NIDS-SLAM[82]	82.45	84.08	76.99	85.94	82.37
	DNS-SLAM[83]	88.32	84.90	81.20	84.66	84.77
	SNI-SLAM[84]	88.42	87.43	86.16	87.63	87.41
	SGS-SLAM[85]	92.95	92.91	92.10	92.90	92.72
	Hier-SLAM[86]	95.25	95.81	95.73	95.52	95.58
RGB	HI-SLAM[36]	74.93	79.55	80.90	71.53	76.72
	Ours	90.27	92.80	91.11	92.45	91.65

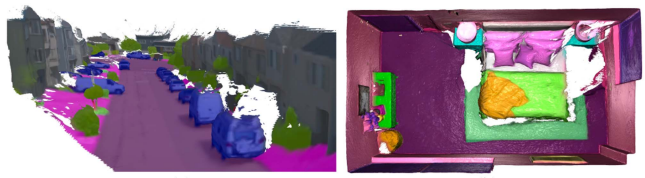

Fig. 15. Semantic reconstruction results: (a) outdoor scene from *100613* of Waymo Open dataset and (b) indoor scene from *Room1* of the Replica dataset.

struggles to track camera poses due to poor generalizability to the new environment, our system maintains stable tracking and successfully detects loop closures with the highest accuracy.

J. Extension to Semantic Reconstruction

As an extension, we demonstrate the capability of our system for semantic scene reconstruction by incorporating 2-D semantic information into the 3DGS representation. Each Gaussian primitive is augmented with semantic color channels in addition to its existing geometric and appearance attributes. These semantic channels can be efficiently rasterized to the image plane alongside color and depth, enabling simultaneous semantic colorization of the reconstructed scene. For semantic optimization, we maintain the same pipeline structure as our depth and pose optimization framework, with an additional L1 semantic RGB loss term that measures the absolute difference between rendered and ground truth semantic color maps. Following the evaluation protocol of [82], we assess our semantic reconstruction performance on the Replica dataset using mean Intersection over Union (mIoU) as the primary metric. We evaluate on four standard sequences to enable direct comparison with existing baseline methods.Table XII presents quantitative results comparing our approach against recent RGB-D and RGB-only semantic SLAM methods, including concurrent work Hier-SLAM [86]. Our system achieves competitive performance compared to state-of-the-art RGB-D methods, while significantly outperforming the RGB-only baseline HI-SLAM [36] by a margin of 14.93%. This substantial improvement can be attributed to two key factors: 1) our more accurate geometry reconstruction provides better surface boundaries for semantic label assignment and 2) the explicit 3DGS representation allows for sharper semantic boundaries compared to implicit NeRF-based approaches that often struggle with object delineation in complex scenes or regions with small objects. Fig. 15 demonstrates qualitative results: indoor scene from the Replica dataset with ground truth semantic labels, and outdoor driving scene from the Waymo Open dataset using Mask2Former [87] predictions as semantic inputs.

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

V. Conclusion

This article presents HI-SLAM2, a novel monocular SLAM system that achieves fast and accurate dense 3-D scene reconstruction through four complementary modules. The online tracking module enhances depth and pose estimation by integrating depth priors with grid-based scale alignment, while parallel PGBA in the online loop closing module corrects pose and scale drift. Our mapping approach leverages 3-D Gaussian splatting for compact scene representation, continuously refined during SLAM tracking. We enhance geometric consistency through monocular normal priors and unbiased ray-Gaussian intersection depth for splat-based rasterization. During offline refinement, we achieve high-fidelity reconstruction by incorporating exposure compensation and performing joint optimization of camera poses, 3DGS map, and exposure parameters. Extensive evaluations on challenging datasets demonstrate that HI-SLAM2 outperforms state-of-the-art methods in accuracy and completeness while maintaining superior runtime performance. Our system achieves high-quality geometry and appearance reconstruction without the typical tradeoffs observed in other methods.

Limitations: Our system has three main limitations: First, the current proximity-based loop closure detection shows limited robustness in the ETH3D dataset when encountering view occlusions and textureless regions, suggesting the need for learned feature-based place recognition. Second, in city-scale scenes, mapping quality can degrade due to limited optimization budget, indicating the need for submap optimization strategies. Third, the system assumes static environments. Incorporating dynamic object detection and tracking, along with motion segmentation, would enable robust operation in dynamic environments.

References

[1] R. A. Newcombe et al., "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Eeality*, 2011, pp. 127–136.

[2] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "ElasticFusion: Dense slam without a pose graph," in *Proc. Robot., Sci. Syst.*, vol. 11. Rome, Italy, 2015, Art. no. 3.

[3] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–18, 2017.

[4] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural RGB-D surface reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6290–6301.

[5] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "iMAP: Implicit mapping and positioning in real-time," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6229–6238.

[6] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auton. Robots*, vol. 4, pp. 333–349, 1997.

[7] J. Zhang et al., "Loam: LiDAR odometry and mapping in real-time," in *Proc. Robot.: Sci. Syst.*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[8] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LiDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 1271–1278.

[9] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.

[10] Y. Pan, X. Zhong, L. Wiesmann, T. Posewsky, J. Behley, and C. Stachniss, "PIN-SLAM: LiDAR SLAM using a point-based implicit neural representation for achieving global map consistency," *IEEE Trans. Robot.*, vol. 40, pp. 4045–4064, 2024.

[11] C. Cadena et al., "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[12] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[13] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[14] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Computer Vis.*, Springer, 2014, pp. 834–849.

[15] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 249–265, Apr. 2017.

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

[17] C. Campos, R. Elvira, J. J. G. Rodrıguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[18] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *Proc. Int. Conf. Computer Vis.*, 2011, pp. 2320–2327.

[19] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, 2017.

[20] Z. Teed and J. Deng, "DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 16558–16569.

[21] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Proc. Eur. Conf. Computer Vis.*, Springer, 2020, pp. 402–419.

[22] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, and D. Tao, "GMFlow: Learning optical flow via global matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8121–8130.

[23] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "MVSNet: Depth inference for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 767–783.

[24] A. Eftekhar, A. Sax, J. Malik, and A. Zamir, "Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3D scans," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10786–10796.

[25] G. Bae, I. Budvytis, and R. Cipolla, "Estimating and exploiting the aleatoric uncertainty in surface normal estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 13137–13146.

[26] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular slam with learned depth prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6243–6252.

[27] Z. Teed and J. Deng, "Deepv2d: Video to depth with differentiable structure from motion," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–13.

[28] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3828–3838.

[29] Z. Zhu et al., "NICE-SLAM: Neural implicit scalable encoding for SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12786–12796.

[30] A. Rosinol, J. J. Leonard, and L. Carlone, "NERF-SLAM: Real-time dense monocular SLAM with neural radiance fields," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 3437–3444.

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

[32] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison, "Gaussian splatting SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 18039–180 48.

[33] E. Sandström et al., "Splat-SLAM: Globally optimized RGB-only SLAM with 3D Gaussians," in *Proc. Comput. Vis. Pattern Recognit. Workshops*, 2025, pp. 1680–1691.

[34] B. Zhang, C. Fang, R. Shrestha, Y. Liang, X. Long, and P. Tan, "Rade-GS: Rasterizing depth in Gaussian splatting," 2024, *arXiv:2406.01467*.

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

[35] H. Wang, J. Wang, and L. Agapito, "Co-SLAM: Joint coordinate and sparse parametric encodings for neural real-time SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 13293–13302.

[36] W. Zhang, T. Sun, S. Wang, Q. Cheng, and N. Haala, "HI-SLAM: Monocular real-time dense mapping with hybrid implicit fields," *IEEE Robot. Autom. Lett.*, vol. 9, no. 2, pp. 1548–1555, Feb. 2024.

[37] J. Straub et al., "The replica dataset: A digital replica of indoor spaces," 2019, *arXiv:1906.05797*.

[38] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "SCANnet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Computer Vis. Pattern Recognit.*, 2017, pp. 5828–5839.

[39] P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Computer Vis. Pattern Recognit.*, 2020, pp. 2446–2454.

[40] T. Schöps, T. Sattler, and M. Pollefeys, "BAD SLAM: Bundle adjusted direct RGB-D SLAM," in *Proc. Conf. Computer Vis. Pattern Recognit.*, 2019, pp. 134–144.

[41] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai, "Scannet : A high-fidelity dataset of 3D indoor scenes," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 12–22.

[42] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2016, pp. 4104–4113.

[43] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, 2007.

[44] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang, "Deepmvs: Learning multi-view stereopsis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2821–2830.

[45] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1623–1637, Mar. 2022.

[46] C. Wu, "VisualSFM: A visual structure from motion system," 2011. [Online]. Available: http://www.cs.washington.edu/homes/ccwu/vsfm

[47] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. pattern Recognit.*, IEEE, 2005, vol. 2, pp. 807–814.

[48] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, 2009, Art. no. 24.

[49] 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," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[50] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–15, 2022.

[51] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *Proc. Eur. Conf. Computer Vis.*., 2022, pp. 333–350.

[52] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5501–5510.

[53] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[54] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2D Gaussian splatting for geometrically accurate radiance fields," in *Proc. ACM SIGGRAPH Conf. Papers*, 2024, pp. 1–11.

[55] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proc. 23rd Annu. Conf. Comput. Graph. Interactive Techn.*, 1996, pp. 303–312.

[56] L. Koestler, N. Yang, N. Zeller, and D. Cremers, "TANDEM: Tracking and dense mapping in real-time using deep multi-view stereo," in *Proc. Conf. Robot Learn.*, 2022, pp. 34–45.

[57] X. Zuo, N. Yang, N. Merrill, B. Xu, and S. Leutenegger, "Incremental dense reconstruction from monocular video with guided sparse feature volume fusion," *IEEE Robot. Autom. Lett.*, vol. 8, no. 6, pp. 3876–3883, 2023.

[58] Z. Zhu et al., "NICER-SLAM: Neural implicit scene encoding for RGB SLAM," in *Proc. Int. Conf. 3D Vis.*., 2024, pp. 42–52.

[59] G. Zhang, E. Sandström, Y. Zhang, M. Patel, L. Van Gool, and M. R. Oswald, "GLORIE-SLAM: Globally optimized RGB-only implicit encoding point cloud slam," 2024, *arXiv:2403.19549*.

[60] W. Zhang, S. Wang, X. Dong, R. Guo, and N. Haala, "BAMF-SLAM: Bundle adjusted multi-fisheye visual-inertial slam using recurrent field transforms," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 6232–6238.

[61] H. Strasdat, J. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular slam," in *Proc. Robot.: Sci. Syst.*, vol. 2, no. 3, 2010, Art. no. 5.

[62] W. Niemeier, "Ausgleichungsrechnung: statistische auswertemethoden," de Gruyter, 2008.

[63] B. Kerbl, A. Meuleman, G. Kopanas, M. Wimmer, A. Lanvin, and G. Drettakis, "A hierarchical 3D Gaussian representation for real-time rendering of very large datasets," *ACM Trans. Graph.*, vol. 43, no. 4, pp. 1–15, 2024.

[64] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:6628106

[65] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8024–8035.

[66] S. Yu, C. Cheng, Y. Zhou, X. Yang, and H. Wang, "RGB-only Gaussian splatting SLAM for unbounded outdoor scenes," in *Proc. 2025 IEEE Int. Conf. Robot. Autom.*, 2025, pp. 11 068–11 074.

[67] M. M. Johari, C. Carta, and F. Fleuret, "ESLAM: Efficient dense SLAM system based on hybrid representation of signed distance fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 17408–17419.

[68] E. Sandström, Y. Li, L. Van Gool, and M. R. Oswald, "Point-SLAM: Dense neural point cloud-based SLAM," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 18433–18 444.

[69] P. Zhu, Y. Zhuang, B. Chen, L. Li, C. Wu, and Z. Liu, "MGS-SLAM: Monocular sparse tracking and Gaussian mapping with depth smooth regularization," *IEEE Robot. Autom. Letters*, vol. 9, no. 11, pp. 9486–9493, 2024.

[70] L. Zhu, Y. Li, E. Sandström, K. Schindler, and I. Armeni, "Loopsplat: Loop closure by registering 3D Gaussian splats," in *Proc. 2025 Int. Conf. 3D Vis.*, 2025, pp. 156–167.

[71] Y. Zhang, F. Tosi, S. Mattoccia, and M. Poggi, "GO-SLAM: Global optimization for consistent 3D instant reconstruction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2023, pp. 3704–3714.

[72] 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," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 21584–21593.

[73] Z. Teed, L. Lipson, and J. Deng, "Deep patch visual odometry," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 39033–39051.

[74] L. Lipson, Z. Teed, and J. Deng, "Deep patch visual SLAM," in *Eur. Conf. Comput. Vis.*. Springer, 2024, pp. 424–440.

[75] R. Murai, E. Dexheimer, and A. J. Davison, "MASt3R-SLAM: Real-time dense SLAM with 3D reconstruction priors," in *Proc. Comput. Vis. Pattern Recognit. Conf.*, 2025, pp. 16 695–16 705.

[76] V. Yugay, Y. Li, T. Gevers, and M. R. Oswald, "Gaussian-SLAM: Photo-realistic dense SLAM with Gaussian splatting," 2023, *arXiv:2312.10070*.

[77] W. Yin et al., "Metric3D: Towards zero-shot metric 3D prediction from a single image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 9043–9053.

[78] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "ZoeDepth: Zero-shot transfer by combining relative and metric depth," 2023,*arXiv:2302.12288*.

[79] L. Yang et al., "Depth anything V2," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 37, 2024, pp. 21875–21911.

[80] G. Bae and A. J. Davison, "Rethinking inductive biases for surface normal estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 9535–9545.

[81] V. Ress, J. Meyer, W. Zhang, D. Skuddis, U. Soergel, and N. Haala, "3D Gaussian splatting aided localization for large and complex indoor-environments," in *Proc. Int. Arch. Photogrammetry, Remote Sens. Spatial Inform. Sci.*, 2025, vol. XLVIII-G-2025, pp. 1283–1290.

[82] Y. Haghighi, S. Kumar, J.-P. Thiran, and L. Van Gool, "Neural implicit dense semantic SLAM," 2023, *arXiv:2304.14560*.

[83] K. Li, M. Niemeyer, N. Navab, and F. Tombari, "DNS-SLAM: Dense neural semantic-informed SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 7839–7846.

[84] S. Zhu et al., "SNI-SLAM: Semantic neural implicit SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 21167–21177.

[85] M. Li et al., "SGS-SLAM: Semantic Gaussian splatting for neural dense SLAM," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 163–179.

[86] B. Li, Z. Cai, Y.-F. Li, I. Reid, and H. Rezatofighi, "HIER-SLAM: Scaling-up semantics in SLAM with a hierarchically categorical Gaussian splatting," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2025, pp. 9748–9754.

[87] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1290–1299.

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

Wei Zhang received the B.S. and M.S. degrees in geodesy and geoinformatics from the University of Stuttgart, Stuttgart, Germany, in 2015 and 2016, respectively. He is currently working toward the Ph.D. degree in photogrammetry with the Institute for Photogrammetry and Geoinformatics, University of Stuttgart.

His research interests include dense visual SLAM, 3-D scene reconstruction, and multisensor fusion for autonomous navigation and mapping.

Qing Cheng received the B.Sc. degree in electronic information engineering from Shanghai University, Shanghai, China, in 2015, and the M.Sc. degree in information technology from the University of Stuttgart, Stuttgart, Germany, in 1018. He is currently working toward the Ph.D. degree in computer vision and artificial intelligence with the School of Computation, Information and Technology, Technical University of Munich, Munich, Germany.

His research focuses on visual SLAM, 3-D scene reconstruction, scene understanding, and generative 3-D modeling.

David Skuddis received the B.S. and M.S. degrees in aerospace engineering from the University of Stuttgart, Stuttgart, Germany, in 2016 and 2019, respectively. He is currently working toward the Ph.D. degree in photogrammetry with the Institute for Photogrammetry and Geoinformatics, University of Stuttgart.

His research interests include LiDAR and visual SLAM, LiDAR-based global localization, and autonomous off-road navigation.

Niclas Zeller received the Ph.D. degree in photogrammetry and remote sensing from the Technical University of Munich (TUM), Munich, Germany, in 2018.

Subsequently, he spend three years in industry as an ADAS Perception Developer at Visteon and a Senior CV and AI Researcher at Artisense. Between 2020 and 2021, he was a Lecturer at the Chair of Computer Vision and AI at TUM. Since 2021, he has been a Professor of signal processing and mobile robotics, Karlsruhe University of Applied Sciences, Karlsruhe, Germany.

Daniel Cremers received the bachelor's degrees in mathematics and physics and the master's degree in theoretical physics from the University of Heidelberg, Heidelberg, Germany, in 1994 and 1997, respectively.

After postdoctoral research at UCLA and a position at Siemens Corporate Research, he became an Associate Professor with the University of Bonn in 2005, then the Chair of Computer Vision and AI at the Technical University of Munich in 2009. He has played key roles in the academic community as an Editor, Program Chair, an Organizer of ECCV 2018, and since 2023, the President of the European Computer Vision Association. From 2020 until 2023, he was listed among the top 10 most influential scholars in robotics of the last decade. He actively supports startups as a co-founder and advisor.

Dr. Cremers research has earned numerous honors, including multiple ERC grants, the 2016 Gottfried Wilhelm Leibniz Award and the ECCV 2024 Koenderink Test of Time Award.

Norbert Haala received the B.S. and M.Sc. degrees in geodesy and geoinformatics, the Ph.D. degree in photogrammetric computer vision from the University of Stuttgart, Stuttgart, Germany, in 1996, and the Venia Legendi (Habilitation) in photogrammetry and computer vision from the University of Stuttgart, in 2004.

He is currently a Professor with the Institute for Photogrammetry and Geoinformatics, University of Stuttgart, Faculty of Aerospace Engineering and Geodesy, where he is responsible for research and teaching in photogrammetric computer vision, image processing and SLAM.

Dr. Haala is a winner of the Carl Pulfrich Award in 2013 and is actively serving within the International Society for Photogrammetry and Remote Sensing at different positions.

Authorized licensed use limited to: UNIVERSITAETSBIBL STUTTGART. Downloaded on March 07,2026 at 16:09:36 UTC from IEEE Xplore. Restrictions apply.