MHED-SLAM: Multi-Scale Hybrid Encoding-Based Decoupled SLAM

Dengfang Feng[1], Wenyang Qin[1], Zhongchen Shi[*2,3,4], Wei Chen[2,3,4], Yanhui Duan[1], Liang Xie[2,3,4], Erwei Yin[*2,3,4]

[1]School of Systems Science and Engineering, Sun Yat-sen University, China
[2]Defense Innovation Institute, Academy of Military Sciences (AMS), China
[3]Intelligent Game and Decision Laboratory, China
[4]Tianjin Artificial Intelligence Innovation Center (TAIIC), China
shizhongchen@buaa.edu.cn, yinerwei1985@gmail.com

Abstract

Neural Radiance Fields (NeRF)-based Visual Simultaneous Localization and Mapping (SLAM) achieve superior scene geometric modeling and robust camera tracking by leveraging neural representations. Existing methods typically relied on multi-resolution hash encoding with truncated signed distance fields (TSDF) to achieve high frame rates. However, unavoidable hash collisions can lead to artifacts, and multi-view color inconsistencies in indoor scenes can result in shape-radiance ambiguity, adversely affecting geometric quality and tracking accuracy. To address these issues, we propose a novel Multi-scale Hybrid Encoding-based Decoupled SLAM (MHED-SLAM). First, to mitigate the adverse effects of hash collisions and reduce the number of learnable parameters, we innovatively fuse a coarse-scale hash triplane with a fine-scale hash grid within a single latent volume. Second, to enable precise geometric reconstruction and camera tracking, we decouple the reconstruction and rendering processes, independently learning a TSDF field for reconstruction and a density field for rendering. Third, we devise a Symmetric Kullback-Leibler (SKL) strategy based on ray termination distributions to align the probability distributions derived from the TSDF and density fields for their synchronous convergence. Extensive experimental evaluations demonstrate that our approach surpasses the state-of-the-art (SOTA) methods by utilizing a faster frame rate of 20 Hz and fewer parameters, while achieving higher tracking and reconstruction accuracy.

Introduction

Visual Simultaneous Localization and Mapping (SLAM) is a fundamental task in computer vision and robotics, supporting many applications in navigation (Wang et al. 2025), autonomous systems, and virtual or augmented reality (Wang et al. 2024). Classic SLAM approaches, such as ORB-SLAM (Mur-Artal, Montiel, and Tardos 2015) and VINS-mono (Qin, Li, and Shen 2018), achieve high-precision localization and real-time performance. However, these methods focus on camera tracking with sparse point clouds, limiting their practicality in dense 3D reconstruction and rendering. The emergence of Neural Radiance Fields (NeRF) introduced a feed-forward model for scene representation and consequently infers 3D geometry and rendering in a single forward pass(Zhang et al. 2025).

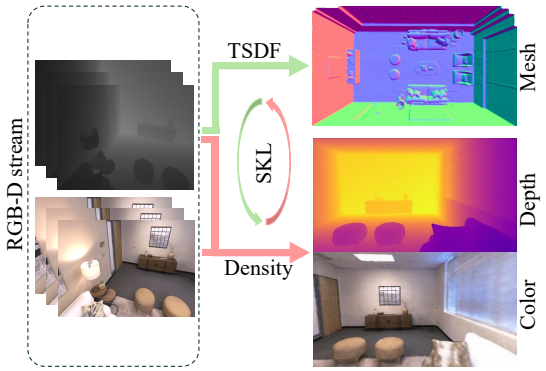

Figure 1: The proposed MHED-SLAM incorporates a multi-scale hybrid encoding and utilizes a parallel architecture to learn the TSDF field for capturing scene geometry and the density field for rendering. When combined with the SKL self-supervised strategy for synchronous convergence, the system achieves tracking and reconstruction performance that exceeds SOTA methods, while utilizing the fewest model parameters and attaining the fastest runtime.

Recent NeRF-based SLAM has investigated diverse scene representations, such as voxel grid (Zhu et al. 2022), feature tri-plane (Johari, Carta, and Fleuret 2023), multi-resolution hash encoding (Müller et al. 2022), and neural point clouds (Sandström et al. 2023). Among these representations, multi-resolution hash encoding has been widely adopted due to its outstanding convergence. Approaches such as Co-SLAM (Wang, Wang, and Agapito 2023), EC-SLAM (Li et al. 2025), and QQ-SLAM (Jiang, Hua, and Han 2025) encode scene information with a multi-resolution hash grid and achieve real-time performance. However, hash collisions are intrinsic to this encoding, which frequently generate artifacts.

Truncated Signed Distance Functions (TSDF) have emerged as a promising alternative to volumetric density in NeRF frameworks because they delineate object surfaces precisely and converge quickly. Techniques such as NeuS (Wang et al. 2021) and VolSDF (Yariv et al. 2021) in-

*Corresponding authors.

corporate Signed Distance Functions (SDF) into NeRF and achieve high-fidelity object-level 3D reconstruction. However, their performance declines in indoor scenes due to their dependence on multi-view color consistency for geometry supervision, which causes shape–radiance ambiguity in regions with textureless or sparse observations. In NeRF-based SLAM, imprecise pose estimation further exacerbates these issues, thereby degrading both reconstruction quality and tracking accuracy.

To address these challenges, we introduce MHED-SLAM, which jointly optimizes camera poses and map representations. First, we present a multi-scale hybrid encoding that architecturally interleaves coarse-scale hash tri-plane with fine-scale hash grids in a single latent volume. This cross-scale fusion not only suppresses hash-collision artifacts inherent in multi-resolution encodings, but also curbs parameter growth, safeguarding real-time performance. Second, we innovatively designed a decoupled architecture that independently learns a TSDF field for surface reconstruction and a density field for volumetric rendering (Figure 1). This disentangled design mitigates geometric degradation arising from multi-view color inconsistencies, enhancing the tracking robustness of the SLAM system. Third, we introduce a novel self-supervised Symmetric Kullback-Leibler (SKL) strategy for MHED-SLAM, grounded in ray termination distributions. By aligning the distributions derived from the TSDF and density fields, this approach synergistically optimizes both the reconstruction and rendering processes. Extensive evaluations and ablation studies on multiple synthetic and real-scan datasets show that MHED-SLAM outperforms state-of-the-art (SOTA) methods while running in real time at 20 Hz.

Overall, our contributions are as follows.

- We present a NeRF-based SLAM framework, MHED-SLAM, that decouples the reconstruction and rendering processes, achieving SOTA performance in both tracking and reconstruction at a speed of 20 Hz.
- We devise a novel multi-scale hybrid scene-encoding scheme that mitigates artifacts induced by hash collisions while minimizing the overall model parameters.
- We introduce an SKL self-supervised strategy tailored to MHED-SLAM's decoupled architecture, which synergistically optimizes reconstruction and rendering to ensure their synchronous convergence.

Related Work

Traditional vSLAM. Traditional vSLAM methods mainly depend on geometric constraints to achieve scene reconstruction through feature point matching. Notable contributions, such as MonoSLAM (Davison et al. 2007), pioneered real-time monocular SLAM and parallel tracking. ORB-SLAM (Mur-Artal, Montiel, and Tardos 2015) refined these approaches, providing a robust system that is able to adapt to diverse camera configurations. DTAM (Newcombe, Lovegrove, and Davison 2011) emerged as an early direct method that uses photometric error minimization for semi-dense reconstruction. KinectFusion (Newcombe et al. 2011) used RGB-D cameras for real-time dense surface modeling. Subsequent research, including ElasticFusion (Whelan et al. 2016) and BundleFusion (Dai et al. 2017b), improved the robustness of dense vSLAM systems. However, traditional methods continue to face challenges in representing complex geometries and effectively processing large-scale environmental reconstructions.

NeRF-based vSLAM. iMAP (Sucar et al. 2021) was a pioneering NeRF-based SLAM framework that approximated the global scene map with a single MLP. However, the limited representational capacity of the MLP resulted in a catastrophic forgetting problem. NICE-SLAM (Zhu et al. 2022) alleviated this limitation by introducing a hierarchical voxel-grid storage scheme. Co-SLAM (Wang, Wang, and Agapito 2023) further advanced the field by proposing joint coordinate and sparse grid encoding, improving computational efficiency. Other approaches improved system performance by incorporating additional information. HERO-SLAM (Xin et al. 2024) extracted and matched features using SuperPoint and LightGlue. PLGSLAM (Deng et al. 2024) exploited SIFT keypoints to assist pose optimization. SNI-SLAM (Zhu et al. 2024) and NIS-SLAM (Zhai et al. 2024) integrated semantic cues to improve camera tracking and scene reconstruction.

3DGS-based vSLAM. 3D Gaussian Splatting (3DGS) was extended to SLAM systems due to its outstanding rendering quality and speed. Photo-SLAM (Huang et al. 2024) and RTG-SLAM (Peng et al. 2024) utilized 3DGS in the backend to optimize map representations, while the frontend employed ORB-SLAM3 and ORB-SLAM2 for camera tracking, respectively. Furthermore, methods such as SplaTAM (Keetha et al. 2024) and MonoGS (Matsuki et al. 2024) leveraged the differentiability of 3DGS to directly optimize camera poses, achieving a fully end-to-end 3DGS-based SLAM. GS-SLAM (Yan et al. 2024) introduced an adaptive expansion strategy that added or removed noisy 3D Gaussians. Although 3DGS-based methods made significant advances in image rendering, they still faced challenges related to real-time performance and accurate surface reconstruction, which limited their ability to meet the demands of real-time SLAM.

Method

Figure 2 shows the MHED-SLAM pipeline. We employ a stream of RGB-D data $\{I_i\}_{i=1}^{N}\{D_i\}_{i=1}^{N}$ with known intrinsics of the camera $K \in \mathbb{R}^{3\times3}$ as input. The SLAM predicts camera poses $\{R_i|t_i\}_{i=1}^{N}$ and a composite implicit map representation F_τ, which integrates color $\boldsymbol{c}$, truncated signed distance s, and volume density σ:

$$F_\tau \rightarrow (\boldsymbol{c}, \sigma, s) \tag{1}$$

Similarly to previous work, our system is partitioned into two threads: a tracking thread and a mapping thread. All network parameters are initialized randomly, the first frame is fixed, and multiple training iterations are carried out to initialize the map. For each subsequent image frame, the tracking thread initially estimates the camera pose using a constant-velocity motion model. Subsequently, a limited subset of pixels is consistently sampled and scene parameters remain unchanged. These sampled pixels are aligned

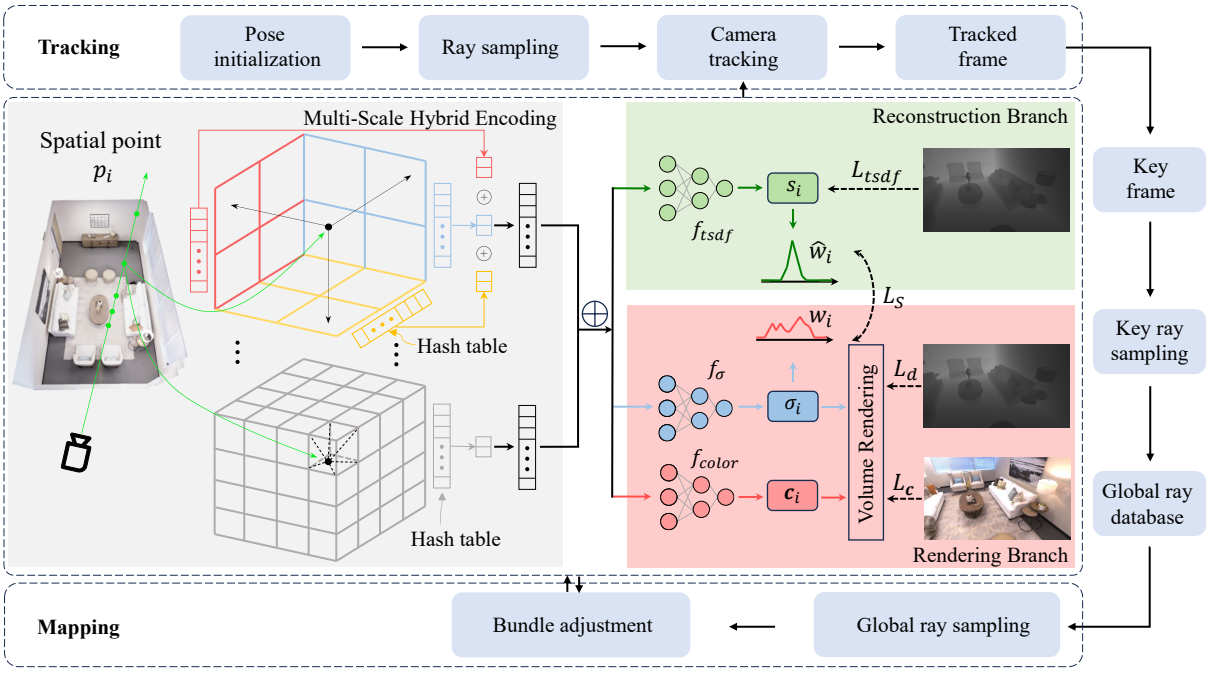

Figure 2: **Overview of MHED-SLAM.** Our approach comprises three primary components: 1) Scene Representation: The entire scene is represented using a multi-scale hybrid encoding (the hash tri-plane for coarse scale and the hash grid for fine scale), and three lightweight MLPs within the reconstruction and rendering branches map spatial coordinates to TSDF values, volume density, and RGB values. 2) Tracking: This module optimizes per-frame camera poses by minimizing the objective function. 3) Mapping: This module randomly samples global rays and subsequently performs joint optimization of the implicit map and pose parameters through bundle adjustment. The SKL self-supervised strategy encourages synchronized convergence by aligning the ray termination distributions of two branches that share underlying features.

with the scene to refine the pose. In the mapping thread, keyframes are selected at predetermined intervals, where joint optimization is performed to optimize the implicit map representation and the associated camera poses.

Multi-Scale Hybrid Encoding

Previous work (Hua and Wang 2024) has demonstrated that different encoding schemes involve trade-offs among mapping accuracy, image rendering quality, and computational efficiency. Consequently, no single scheme can satisfy all requirements, restricting either robustness or efficiency. For example, relying solely on a multi-resolution hash encoding introduces noticeable artifacts. To address this challenge, we propose a novel multi-scale hybrid encoding approach, as illustrated in Figure 3. Based on a multi-resolution hash encoding structure with L layers (ranging from L_{low} to L_{top}), we partition the architecture into two levels: coarse scale and fine scale.

For the coarse-scale layers from L_{low} to $L/2$, we employ three orthogonal multi-resolution 2D hash grids, also referred to as hash planes. Each 3D point is projected onto these three 2D planes and encoded via bilinear interpolation. The three resulting encoded features are concatenated to form a tri-plane feature vector g_{coarse}. This design mitigates the impact of hash collisions occurring in any single plane by leveraging the complementary encoding from the other two planes, reducing the overall negative effect of collisions and enhancing the robustness of the encoding.

Due to the characteristics of hash encoding, high-resolution grids result in the number of encoding parameters exceeding the maximum hash table capacity. Under these conditions, using three hash planes leads to a parameter count that is three times greater than that of a single 3D hash grid at the same level. Therefore, for the fine-scale layers from $L/2$ to L_{top}, we utilize a single multi-resolution 3D hash grid to suppress excessive growth in the total num-

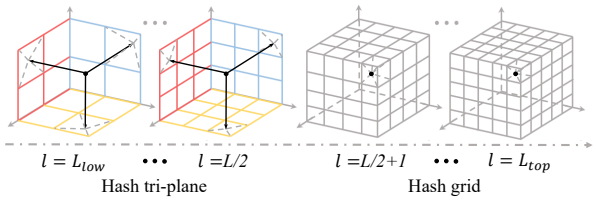

Figure 3: A multi-scale hybrid encoding comprising coarse-scale hash tri-plane at $L/2$ layers and fine-scale hash grids at $L/2$ layers.

ber of parameters. The feature vector g_{fine} for a 3D point is obtained by trilinear interpolation from the grid vertices.

Finally, the encoded features of both scales are concatenated, which are then fed into three small MLPs for decoding. In our method, the number of multi-resolution hash encoding layers L is set to 16.

Decoupled Representation Architecture

Existing NeRF-based SLAM systems typically learn the TSDF of the scene and use TSDF-based rendering to ensure convergence speed. This approach often results in geometric information being corrupted by color noise, leading to the generation of noisy surfaces.

Considering the substantial distinction between the tasks of reconstruction and rendering in SLAM systems, a decoupled architecture is introduced. The proposed framework consists of two parallel branches dedicated to reconstruction and rendering, respectively; the reconstruction branch aims to accurately capture the geometric structure of the scene. This objective is achieved using two compact MLPs, f_{tsdf} and f_σ, to extract the TSDF and σ of the scene, respectively. In contrast, the rendering branch learns the appearance of the scene using an MLP, f_{color}, which shares a similar architecture to predict the colors applied in rendering. Specifically, spatial sample points, characterized by features derived from the proposed multi-scale hybrid encoding, including hash tri-plane features and hash grid features, are concatenated and fed into the MLPs to decode scene information. The supervision of the TSDF is accomplished by employing depth sensor measurements as an approximation, while color information is governed by observed images.

$$f_{sdf}(g_c, g_f) \rightarrow s, f_{color}(g_c, g_f) \rightarrow \boldsymbol{c}, f_\sigma(g_c, g_f) \rightarrow \sigma \quad (2)$$

Volume Rendering. According to NeRF, the system retrieves the ray $\boldsymbol{r}$ for each frame from the optical center $\boldsymbol{o}$ to the pixel coordinates using the intrinsic and extrinsics of the camera. The radiation field samples N points along each ray. For each point $p_i = \boldsymbol{o} + z_i\boldsymbol{r}, i \in \{1, \ldots, N\}$, the radiation field extracts colors $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\}$ and densities $\{\sigma_1, \ldots, \sigma_N\}$. The final pixel color is obtained by accumulating these values along each ray using the volume rendering integral formula. Similarly, the depth value is derived by combining the distance and density values:

$$\hat{\mathbf{C}} = \sum_{i=1}^{N} w_i \boldsymbol{c}_i, \quad \hat{D} = \sum_{i=1}^{N} w_i z_i \quad (3)$$

Where z_i denotes the distance from the sampling points to the camera center; w_i represents the contribution weights of each sampling point to color and depth, calculated as:

$$w_i = \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \quad (4)$$

$$\alpha_i = 1 - exp\big(-\sigma_i(z_{i+1} - z_i)\big) \quad (5)$$

SKL Self-Supervised Strategy

MHED-SLAM enables its two branches to exploit their respective strengths: one in scene reconstruction and the other in image rendering. However, optimizing the branches independently often introduces geometric and photometric inconsistencies. To mitigate these inconsistencies, we propose an SKL self-supervised strategy that aligns the branches via the ray termination distribution. The ray termination distribution (the probability that a ray meets the object surface) resembles a spike-like or narrow bell curve centered at the measured depth (Deng et al. 2022). Therefore, the sampling weights along each ray during volumetric rendering should correspond to this distribution. The rendering branch learns this distribution from per-pixel color and depth supervision, whereas the reconstruction branch approximates it using sdf values sampled along each ray (Eq. 7). Consequently, we devise an SKL loss. Specifically, we apply two KL divergences, $KL(w||\hat{w})$ and $KL(\hat{w}||w)$, to the weights w from the rendering branch and $\hat{w}$ from the reconstruction branch, to quantify the ray termination distributions of the two branches and encourage their alignment through the SKL loss. The specific definition is as follows:

$$\mathcal{L}_S = \mathrm{SKL}(w, \hat{w}) = \frac{1}{2} \sum_{i=1}^{M} \left[w_i \ln \frac{w_i}{\hat{w}_i} + \hat{w}_i \ln \frac{\hat{w}_i}{w_i} \right] \quad (6)$$

$$\hat{w}_i = \hat{\alpha}_i \prod_{j=1}^{i-1} (1 - \hat{\alpha}_j) \quad (7)$$

$$\hat{\alpha}_i = 1 - exp\big(-\beta \cdot \mathrm{sigmoid}(-\beta \cdot s_i)\big) \quad (8)$$

Here, M is a set of sampled pixels. β represents a learnable parameter that regulates the shape of the ray termination distribution curve. The SKL loss encourages both branches to converge towards a unified solution.

As illustrated in Figures 4(a) and 4(b), independent training of the reconstruction and rendering branches results in distinct ray termination distributions. With the self-supervised SKL strategy, the rendering branch assists the

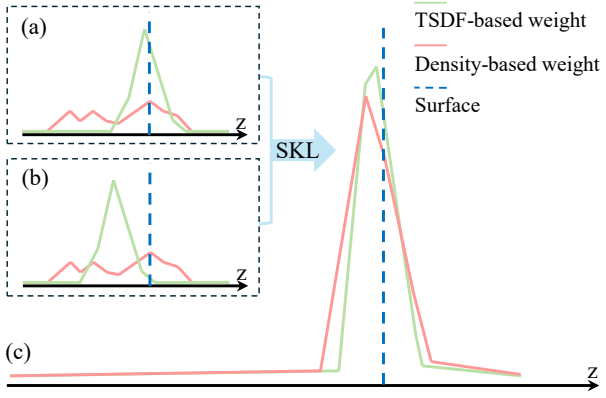

Figure 4: (a) The differing convergence rates of the two branches lead to distinct ray termination distributions. (b) The absence of depth information results in divergent ray termination distributions. (c) The SKL loss constrains the ray termination distributions of both branches to become consistent.

reconstruction branch in filling void regions, whereas the reconstruction branch guides the density convergence within the rendering branch. This bidirectional information flow enables the two branches, sharing fundamental features, to be jointly optimized, allowing their ray termination distributions to converge toward similar patterns rather than being forced into complete uniformity.

Objective Functions

We execute our tracking and mapping process by minimizing an objective function with respect to the learnable parameters τ and the camera parameters $\{R_i|t_i\}$. The objective function primarily comprises three components: color loss, depth loss, and TSDF loss. The color and depth rendering losses are formulated as $L2$ errors between the rendered results and observations:

$$\mathcal{L}_c = \frac{1}{M}\sum_{m=1}^{M}\left(\hat{C}(m) - C(m)\right)^2 \tag{9}$$

$$\mathcal{L}_d = \frac{1}{|M_d|}\sum_{m\in M_d}\left(\hat{D}(m) - D(m)\right)^2 \tag{10}$$

Where M denotes the set of sampled pixels and M_d is the subset of M with valid depth measurements. TSDF computation of the loss for each sampled point facilitates rapid convergence. We select a batch of rays with valid depth measurements. Within the truncation region tr, depth sensor measurements approximate the SDF to calculate the corresponding loss of TSDF :

$$\mathcal{L}_{tsdf} = \frac{1}{|M_d|}\sum_{m\in M_d}\frac{1}{|R_m^{tr}|}\sum_{i\in R_m^{tr}}\left(z_i + s_i\cdot tr - D(m)\right)^2 \tag{11}$$

Where R_m^{tr} is the set of sampled points along a pixel's ray within the truncation region tr, i.e., $|z_i - D(m)| < tr$. We apply distinct importance weights within the central section ($|z_i - D(m)| < 0.4tr$) and the tail section of the truncated section using loss terms λ_{tsdf}^c and λ_{tsdf}^t, respectively. For points prior to the truncation region, we calculate the discrepancy between the predicted TSDF value and 1. This loss constrains the predicted TSDF values of sampled points within free space to approach 1:

$$\mathcal{L}_{fs} = \frac{1}{|M_d|}\sum_{m\in M_d}\frac{1}{|r_m^{fs}|}\sum_{i\in r_m^{fs}}(s_i - 1)^2 \tag{12}$$

We define the final objective function as the sum of the aforementioned losses and modulate their relative contributions using the coefficients $\{\lambda_{fs}, \lambda_{tsdf}^c, \lambda_{tsdf}^t, \lambda_d, \lambda_c, \lambda_S\}$. The SKL loss is applied exclusively during the mapping stage.

Experiments

In this section, we evaluate our proposed method using widely used datasets, comparing its performance against existing implicit representation-based approaches across surface reconstruction, pose estimation, and computational efficiency.

Experimental Setup

Baselines. We evaluate our method against several recent SOTA NeRF-based visual SLAM approaches: iMAP (Sucar et al. 2021), NICE-SLAM (Zhu et al. 2022), Co-SLAM (Wang, Wang, and Agapito 2023), DF Prior (Hu and Han 2023), QQ-SLAM (Jiang, Hua, and Han 2025) and two 3DGS-based methods: MonoGS (Matsuki et al. 2024), SplaTAM (Keetha et al. 2024). Because the surfaces produced by 3DGS-based methods are blurred, reliable mesh extraction becomes difficult. To ensure a fair comparison, we evaluate only the tracking results against these systems.

Metric. We evaluate the quality of reconstruction using Depth L1(cm), Accuracy(cm), Completion(cm), and Completion ratio(%) with a threshold of 5 cm. To ensure a fair comparison across all methods, we adopt the mesh culling strategy employed by Co-SLAM. For camera trajectory assessment, we utilize the ATE RMSE metric, which is reported in centimeters across all datasets.

Dataset. We evaluate our method on three widely used datasets for 3D reconstruction and camera tracking: Replica (Straub et al. 2019), ScanNet (Dai et al. 2017a), and TUM RGB-D (Sturm et al. 2012). We evaluate reconstruction quality using eight synthetic indoor scenes from Replica, encompassing small to medium sized room configurations with an average of 2,000 images per scene. For camera tracking, we evaluate in real-world scenarios by testing our method on six ScanNet scenes and three TUM RGB-D scenes, with ScanNet representing large-scale indoor environments comprising over 5,000 images.

Implementation Details. Our system runs on a desktop PC equipped with an Intel Core i7-12700K processor and an NVIDIA RTX 3090 GPU. The number of iterations in the mapping and tracking processes, as well as the number of ray samples for corresponding pixels, varies across different datasets.

In the mapping thread, we select a keyframe every 5 frames and sample 5% of the total rays, storing them in the keyframe set. For the default configuration, we perform 10 iterations, random sampling $M = 2,000$ rays in each iteration. The weights for different loss functions are assigned as follows:$\lambda_{fs} = 1, \lambda_{tsdf}^c = 50, \lambda_{tsdf}^t = 5, \lambda_d = 0.1, \lambda_c = 1, \lambda_S = 5e\text{-}4$, where λ_S decreases to 0 with the number of iterations. For the TUM RGB-D dataset, we set $\lambda_{tsdf}^c = 250$ and perform 20 iterations.

In the tracking thread, we track every frame, initially estimating the pose of the tracking frame using a constant velocity motion model. During the tracking iterations, we consistently sample 1,000 rays. For the Replica dataset, pose optimization is conducted over 8 iterations. For the ScanNet dataset, we execute 10 iterations. For the TUM RGB-D dataset, we execute 12 iterations.

For ray sampling, we employ both uniform spatial sampling and depth-guided sampling along the rays. In the Replica and TUM RGB-D datasets, we set $N_u = 32$ points for uniform sampling per ray and $N_d = 8$ for depth-guided samples. For the ScanNet dataset, we set $N_u = 48$ and $N_d = 8$.

Evaluation on Replica. Table 1 presents the performance

Methods	Metric	room0	room1	room2	office0	office1	office2	office3	office4	Avg.	FPS
iMAP	Depth L1↓	5.08	3.44	5.78	3.79	3.76	3.97	5.61	5.71	4.64	
	Acc.[cm]↓	4.01	3.04	3.84	3.34	2.10	4.06	4.20	4.34	3.62	
	Comp.[cm]↓	5.84	4.40	5.07	3.62	3.62	4.73	5.49	6.65	4.93	9.9
	Comp. Ratio[%]↑	78.34	85.85	79.40	83.59	83.59	79.73	73.90	74.77	80.50	
	ATE RMSE[cm]↓	5.43	3.22	2.85	2.60	1.30	5.94	5.18	2.42	3.62	
NICE	Depth L1[cm]↓	1.79	1.33	2.20	1.43	1.58	2.70	2.10	2.06	1.90	
	Acc.[cm]↓	2.44	2.10	2.17	1.85	1.56	3.28	3.01	2.54	2.37	
	Comp.[cm]↓	2.60	2.19	2.73	1.84	1.82	3.11	3.16	3.61	2.63	13.7
	Comp. Ratio[%]↑	91.81	93.56	91.48	94.93	94.11	88.27	87.68	87.23	91.13	
	ATE RMSE[cm]↓	1.69	2.04	1.55	0.99	0.90	1.39	3.97	3.08	1.95	
DF Prior	Depth L1[cm]↓	1.44	1.90	2.75	1.43	2.03	7.73	4.81	1.99	3.01	
	Acc.[cm]↓	2.54	2.70	2.25	2.14	2.80	3.58	3.46	2.68	2.77	
	Comp.[cm]↓	2.41	2.26	2.46	1.76	1.94	2.56	2.93	3.27	2.45	-
	Comp. Ratio[%]↑	93.22	94.75	93.02	96.04	94.77	91.89	90.17	88.46	92.79	
	ATE RMSE[cm]↓	1.39	1.55	2.60	1.09	1.23	1.61	3.61	1.42	1.81	
Co-SLAM	Depth L1[cm]↓	1.05	0.85	2.37	1.24	1.48	1.86	1.66	1.54	1.51	
	Acc.[cm]↓	**2.11**	**1.68**	1.99	1.57	**1.31**	2.84	3.06	2.23	2.10	
	Comp.[cm]↓	2.02	1.81	1.96	1.56	1.59	2.43	2.72	2.52	2.08	17.2
	Comp. Ratio[%]↑	95.26	95.19	93.58	96.09	94.65	91.63	90.72	90.44	93.44	
	ATE RMSE[cm]↓	0.72	1.32	1.27	0.62	0.52	2.07	1.47	0.84	1.10	
QQ-SLAM	Depth L1[cm]↓	1.09	**0.69**	2.48	1.18	**0.99**	1.76	1.54	1.68	1.42	
	Acc.[cm]↓	2.38	2.62	2.00	**1.55**	1.37	3.43	3.94	2.16	2.43	
	Comp.[cm]↓	**1.76**	1.77	1.82	1.57	1.39	2.14	2.55	2.46	1.93	3.4
	Comp. Ratio[%]↑	96.39	95.49	94.28	96.10	95.40	94.07	91.78	91.53	94.38	
	ATE RMSE[cm]↓	**0.58**	1.16	0.87	**0.52**	**0.48**	1.74	1.22	0.73	0.91	
Ours	Depth L1[cm]↓	**0.70**	**0.69**	**1.43**	**0.86**	1.28	**1.18**	**1.01**	**0.85**	**1.00**	
	Acc.[cm]↓	2.33	2.67	**1.73**	1.64	1.40	**2.73**	**2.63**	**2.01**	**2.14**	
	Comp.[cm]↓	1.80	**1.63**	**1.72**	**1.41**	**1.33**	**1.97**	**2.23**	**2.10**	**1.77**	**20.0**
	Comp. Ratio[%]↑	**97.04**	**95.98**	**95.72**	**97.77**	**96.61**	**94.45**	**93.84**	**93.94**	**95.29**	
	ATE RMSE[cm]↓	0.59	**0.54**	**0.60**	0.58	0.52	**1.05**	**0.77**	**0.65**	**0.66**	

Table 1: Comparison of reconstruction metrics of MHED-SLAM and existing SLAM methods on Replica.

of multiple methods across eight different scenarios. Our method demonstrates superior performance compared to the most recent NeRF-based systems across all reconstruction metrics. Figure 5 illustrates the visualization results, indicating that our method effectively mitigates artifacts in partially unobserved areas (e.g., the room0 scene) by utilizing the tri-plane, achieving smoother reconstructions over larger regions. Regarding tracking, our method also demonstrates a tracking speed that is 6 times faster than QQ-SLAM, with accuracy surpassing that of QQ-SLAM by 27.5%.

Evaluation on ScanNet. We further evaluate our system on ScanNet, a large-scale real-world indoor dataset. Table 2 presents quantitative results that demonstrate our system's superiority over NeRF-based SLAM and 3DGS-based SLAM in camera tracking accuracy, achieving a 6.7 times improvement in tracking speed compared to QQ-SLAM. Figure 6 shows the qualitative results of reconstruction. QQ-SLAM produces scenes of suboptimal quality, whereas Co-SLAM tends to over-smooth geometric details (e.g., a row of chairs in scene 0059). In contrast, our system achieves a better balance between preserving geometric details and ensuring surface smoothness through multi-scale encoding.

Evaluation on TUM RGB-D. To further evaluate the robustness of our method, we examine its tracking performance on the real-world TUM RGB-D dataset. As shown in Table 3, our method outperforms the latest QQ-SLAM in two scenarios, achieving competitive tracking performance. Additionally, its runtime efficiency surpasses that of the latest NeRF-based SLAM and 3DGS-based SLAM, reaching 15.6 Hz.

	NeRF					3DGS	
Scene	NICE	Co.	Co. †	QQ.	Ours	Spla.	Mono.
0000	8.64	7.18	7.13	6.99	**6.03**	12.81	9.58
0059	12.25	12.29	11.14	9.47	**8.85**	10.13	6.16
0106	8.09	9.57	9.36	8.82	**7.94**	17.68	7.05
0169	10.28	6.62	5.90	6.48	**4.96**	12.08	10.66
0181	12.93	13.43	11.81	13.30	**9.67**	11.14	18.23
0207	**5.59**	7.13	7.14	5.86	6.41	7.49	7.46
AVG.	9.63	9.37	8.75	8.49	**7.31**	11.89	9.86
FPS	1.6	12.8	6.4	2.5	**16.5**	0.2	2.3

Table 2: Comparison of tracking metrics of MHED-SLAM with existing SLAM methods on ScanNet.

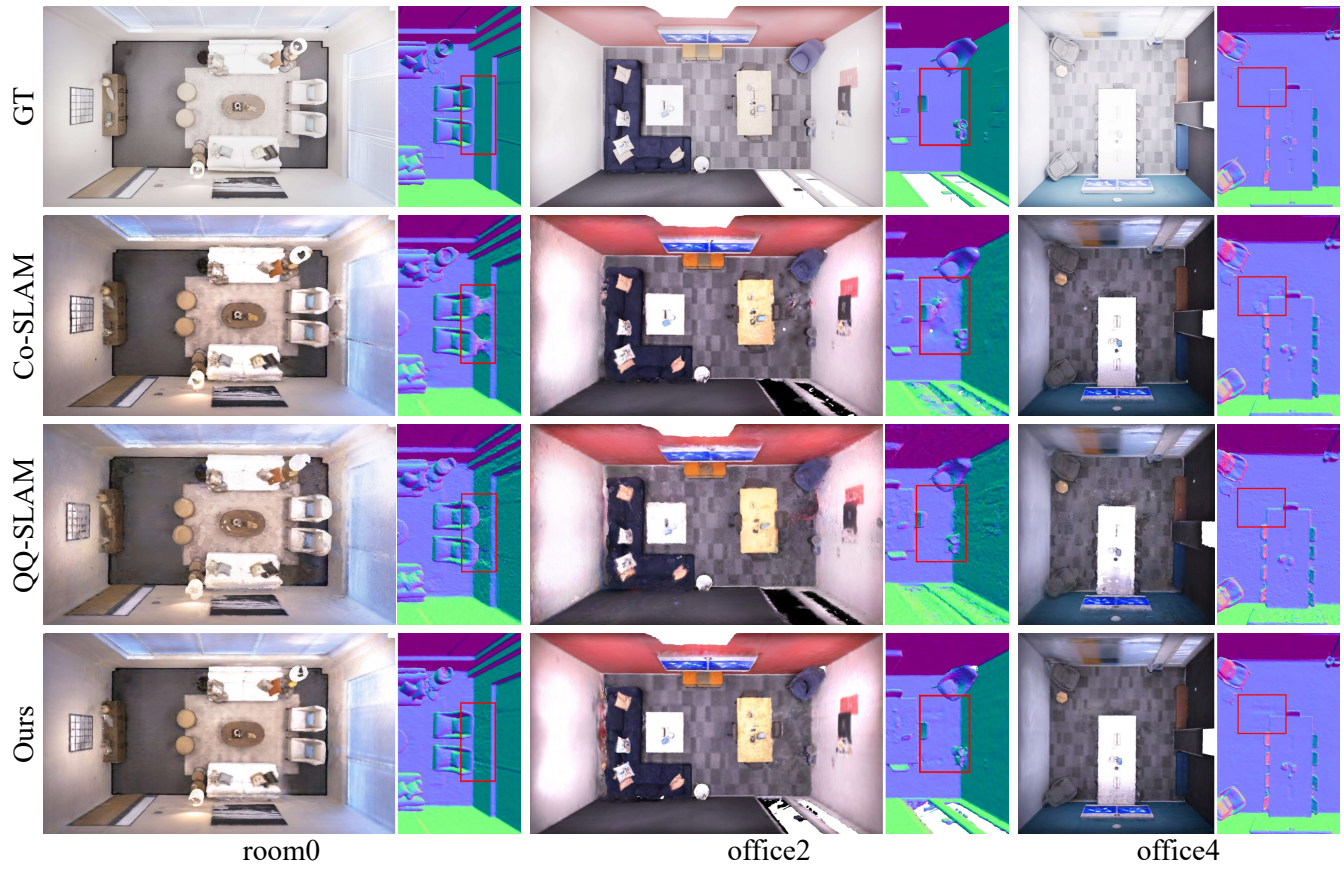

Figure 5: Qualitative comparison MHED-SLAM with existing NeRF-based SLAM approaches on Replica.

Performance Analysis

We evaluate the model parameters and runtime performance (FPS) of all systems on 8 scenes from Replica and 6 scenes from ScanNet. Table 4 presents the average results across all scenes for each dataset, obtained on a PC equipped with a 3.60 GHz Intel Core i7-12700K CPU and an NVIDIA RTX 3090 GPU. For Replica scenes, MHED-SLAM achieves a real-time performance of nearly 20.0 Hz. For the more challenging ScanNet scenes, MHED-SLAM maintains a high frame rate exceeding 16.5 Hz. Our method achieves the smallest model parameters among all compared systems, benefiting from the hash-based sparse grid encoding at fine scales and the space-efficient hash tri-plane at coarse scales.

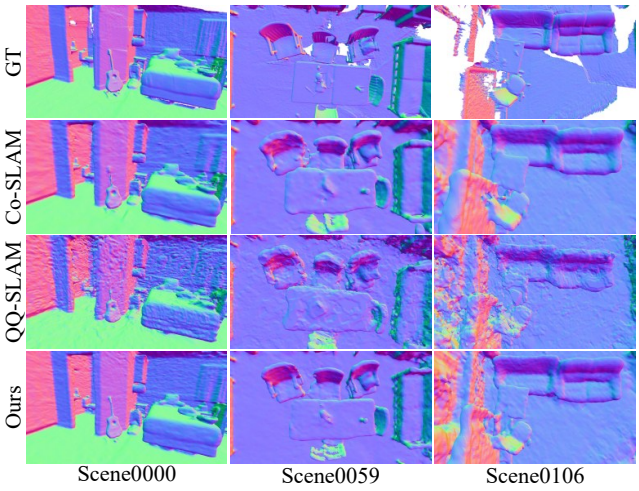

Figure 6: Qualitative comparison of MHED-SLAM with existing SLAM methods on ScanNet.

	Methods	fr1/desk	fr2/xyz	fr3/office	FPS
NeRF	iMAP	4.90	2.00	5.80	0.1
	NICE-SLAM	2.80	2.16	3.14	0.1
	Co-SLAM	2.73	2.02	2.63	13.3
	QQ-SLAM	2.61	**1.70**	2.70	2.6
	Ours	**2.57**	1.91	**2.47**	**15.6**
3DGS	SplaTAM	3.36	1.25	5.14	0.1
	monoGS	1.52	1.58	1.65	1.5

Table 3: Comparison of tracking metrics of MHED-SLAM with existing SLAM methods on TUM RGB-D.

	Method	FPS↑	Param.↓
Replica	NICE-SLAM	13.7	66.4 M
	Co-SLAM	17.2	1.00 M
	QQ-SLAM	3.4	1.00 M
	Ours	**20.0**	**0.8** M
ScanNet	NICE-SLAM	1.6	41.0 M
	Co-SLAM	12.8	3.1 M
	QQ-SLAM	2.5	3.1 M
	Ours	**16.5**	**1.4** M

Table 4: Compare the model sizes and runtime efficiency of different methods on Replica and ScanNet.

Ablations

Effects of Multi-scale Hybrid Encoding. We conducted ablation studies on the scene encoding module in MHED-SLAM using room1 from Replica (Straub et al. 2019). The experiments considered three settings: (i) a single multi-resolution hash grid, (ii) a single multi-resolution hash tri-plane, and (iii) our proposed multi-scale hybrid encoding. Table 5 reports the quantitative results. The multi-scale hybrid encoding attained reconstruction accuracy comparable to that of the multi-resolution hash tri-plane while requiring only 46% of its parameters. Figure 7 visualizes the mesh of the different encodings; due to the introduced tri-plane structure, our method does not produce artifacts on the back of the chair.

Methods	Hash grid	Hash tri-plane	Ours
Acc.↓[cm]	4.06	**2.66**	2.67
Comp.↓[cm]	1.78	1.70	**1.63**
Comp. Ratio↑[%]	95.14	95.40	**95.98**
Depth L1↓[cm]	0.87	0.71	**0.69**
Param.↓[M]	0.98	1.79	**0.82**

Table 5: Ablation study on scene encoding module.

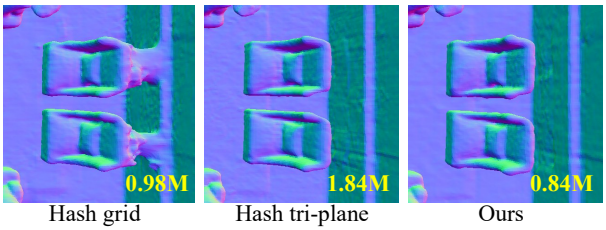

Figure 7: Visual comparison in the room0 scene of Replica.

Effects of Decoupled Architecture. We conducted an ablation study on room1 from Replica (Straub et al. 2019) to evaluate the impact of the decoupled architecture in MHED-SLAM. We first evaluated conventional NeRF-based SLAM with a coupled rendering pipeline (Coup.), and then evaluated the decoupled architecture of MHED-SLAM under different weights λ_S of SKL self-supervised loss (Decoup).

Methods	Coup.	Decoup.		
λ_S	-	0	5e-4	5e-3
Acc.↓[cm]	2.97	2.91	**2.67**	2.82
Comp.↓[cm]	1.75	1.68	**1.63**	1.74
Comp. Ratio↑[%]	95.32	95.62	**95.98**	95.53
Depth L1↓[cm]	0.94	0.85	**0.69**	0.73

Table 6: The impact of different λ_S values on the reconstruction results in the room1 scene.

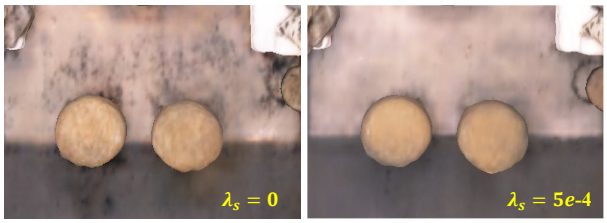

Figure 8: The impact of different λ_S values on the reconstruction results in the room0 scene.

As shown in Table 6, the decoupled architecture achieved the best reconstruction accuracy when $\lambda_S = 5\text{e}{-4}$. The proposed SKL strategy aligns the ray termination distributions of the two pipelines, enabling their synchronous convergence. When $\lambda_s = 0$, black surfaces are observed in the scene (Figure 8 left). This occurrence results from the inconsistent convergence speeds of the two branches, which are unable to adequately represent the colors of the scene. If λ_s is large, it interferes with other loss terms, leading to a decline in reconstruction accuracy (Table 6).

Conclusion

We presented a multi-scale hybrid encoding-based decoupled SLAM system, MHED-SLAM, which achieved SOTA accuracy and real-time performance. Our key innovations included multi-scale hybrid encoding, decoupled reconstruction and rendering processes, with the SKL self-supervised strategy for synchronous convergence. Using the characteristics of NeRF-based representations, we achieved precise camera tracking and complete scene reconstruction without additional supervision, while maintaining minimal model parameters and real-time performance of 20 Hz.

Acknowledgments

This work was supported in part by the grants from the National Natural Science Foundation of China under Grant 62332019 and 62076250, the National Key Research and Development Program of China (2023YFF1203900, 2023YFF1203903).

References

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017a. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the*

IEEE conference on computer vision and pattern recognition, 5828–5839.

Dai, A.; Nießner, M.; Zollhöfer, M.; Izadi, S.; and Theobalt, C. 2017b. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4): 1.

Davison, A. J.; Reid, I. D.; Molton, N. D.; and Stasse, O. 2007. MonoSLAM: Real-time single camera SLAM. *IEEE transactions on pattern analysis and machine intelligence*, 29(6): 1052–1067.

Deng, K.; Liu, A.; Zhu, J.-Y.; and Ramanan, D. 2022. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12882–12891.

Deng, T.; Shen, G.; Qin, T.; Wang, J.; Zhao, W.; Wang, J.; Wang, D.; and Chen, W. 2024. Plgslam: Progressive neural scene represenation with local to global bundle adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19657–19666.

Hu, P.; and Han, Z. 2023. Learning neural implicit through volume rendering with attentive depth fusion priors. *Advances in Neural Information Processing Systems*, 36: 33012–33026.

Hua, T.; and Wang, L. 2024. Benchmarking Implicit Neural Representation and Geometric Rendering in Real-Time RGB-D SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21346–21356.

Huang, H.; Li, L.; Cheng, H.; and Yeung, S.-K. 2024. Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular stereo and rgb-d cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21584–21593.

Jiang, S.; Hua, J.; and Han, Z. 2025. Query Quantized Neural SLAM. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 4057–4065.

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

Keetha, N.; Karhade, J.; Jatavallabhula, K. M.; Yang, G.; Scherer, S.; Ramanan, D.; and Luiten, J. 2024. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21357–21366.

Li, G.; Chen, Q.; Yan, Y.; and Pu, J. 2025. EC-SLAM: Effectively constrained neural RGB-D SLAM with TSDF hash encoding and joint optimization. *Pattern Recognition*, 170: 112034.

Matsuki, H.; Murai, R.; Kelly, P. H.; and Davison, A. J. 2024. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18039–18048.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4): 1–15.

Mur-Artal, R.; Montiel, J. M. M.; and Tardos, J. D. 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics*, 31(5): 1147–1163.

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

Newcombe, R. A.; Lovegrove, S. J.; and Davison, A. J. 2011. DTAM: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, 2320–2327. IEEE.

Peng, Z.; Shao, T.; Liu, Y.; Zhou, J.; Yang, Y.; Wang, J.; and Zhou, K. 2024. Rtg-slam: Real-time 3d reconstruction at scale using gaussian splatting. In *ACM SIGGRAPH 2024 Conference Papers*, 1–11.

Qin, T.; Li, P.; and Shen, S. 2018. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE transactions on robotics*, 34(4): 1004–1020.

Sandström, E.; Li, Y.; Van Gool, L.; and Oswald, M. R. 2023. Point-slam: Dense neural point cloud-based slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 18433–18444.

Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijmans, E.; Green, S.; Engel, J. J.; Mur-Artal, R.; Ren, C.; Verma, S.; et al. 2019. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*.

Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; and Cremers, D. 2012. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 573–580. IEEE.

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

Wang, D.; Wang, J.; Tian, Y.; Fang, Y.; Yuan, Z.; and Xu, M. 2024. PAL-SLAM2: Visual and visual–inertial monocular SLAM for panoramic annular lens. *ISPRS Journal of Photogrammetry and Remote Sensing*, 211: 35–48.

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

Wang, P.; Liu, L.; Liu, Y.; Theobalt, C.; Komura, T.; and Wang, W. 2021. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*.

Wang, Z.; Shi, D.; Qiu, C.; Jin, S.; Li, T.; Qiao, Z.; and Chen, Y. 2025. VecMapLocNet: Vision-based UAV localization using vector maps in GNSS-denied environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 225: 362–381.

Whelan, T.; Salas-Moreno, R. F.; Glocker, B.; Davison, A. J.; and Leutenegger, S. 2016. ElasticFusion: Real-time dense SLAM and light source estimation. *The International Journal of Robotics Research*, 35(14): 1697–1716.

Xin, Z.; Yue, Y.; Zhang, L.; and Wu, C. 2024. Hero-slam: Hybrid enhanced robust optimization of neural slam. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 8610–8616. IEEE.

Yan, C.; Qu, D.; Xu, D.; Zhao, B.; Wang, Z.; Wang, D.; and Li, X. 2024. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19595–19604.

Yariv, L.; Gu, J.; Kasten, Y.; and Lipman, Y. 2021. Volume rendering of neural implicit surfaces. *Advances in neural information processing systems*, 34: 4805–4815.

Zhai, H.; Huang, G.; Hu, Q.; Li, G.; Bao, H.; and Zhang, G. 2024. Nis-slam: Neural implicit semantic rgb-d slam for 3d consistent scene understanding. *IEEE Transactions on Visualization and Computer Graphics*.

Zhang, J.; Li, Y.; Chen, A.; Xu, M.; Liu, K.; Wang, J.; Long, X.-X.; Liang, H.; Xu, Z.; Su, H.; Theobalt, C.; Rupprecht, C.; Vedaldi, A.; Pfister, H.; Lu, S.; and Zhan, F. 2025. Advances in Feed-Forward 3D Reconstruction and View Synthesis: A Survey. arXiv:2507.14501.

Zhu, S.; Wang, G.; Blum, H.; Liu, J.; Song, L.; Pollefeys, M.; and Wang, H. 2024. Sni-slam: Semantic neural implicit slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21167–21177.

Zhu, Z.; Peng, S.; Larsson, V.; Xu, W.; Bao, H.; Cui, Z.; Oswald, M. R.; and Pollefeys, M. 2022. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12786–12796.