NICER-SLAM: Neural Implicit Scene Encoding for RGB SLAM

Zihan Zhu[1*] Songyou Peng[1,2*] Viktor Larsson[3] Zhaopeng Cui[4] Martin R. Oswald[1,5]
Andreas Geiger[6] Marc Pollefeys[1,7]

[1]ETH Zürich [2]MPI for Intelligent Systems, Tübingen [3]Lund University
[4]State Key Lab of CAD&CG, Zhejiang University [5]University of Amsterdam
[6]University of Tübingen, Tübingen AI Center [7]Microsoft

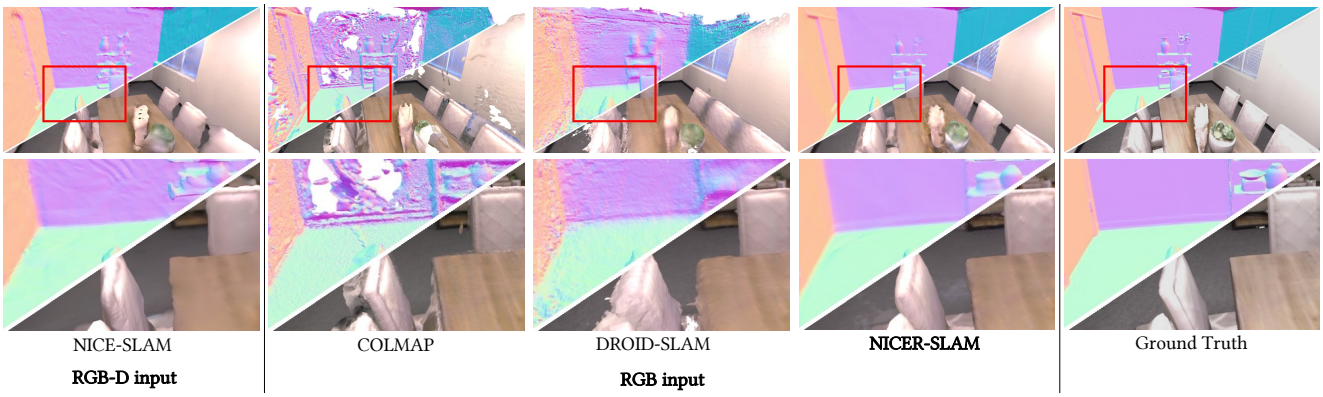

Figure 1. **3D Dense Reconstruction and Rendering from Different SLAM Systems.** On the Replica dataset [50], we compare to dense RGB-D SLAM method NICE-SLAM [75], and monocular SLAM approaches COLMAP [47], DROID-SLAM [58], and our proposed NICER-SLAM.

Abstract

Neural implicit representations have recently become popular in simultaneous localization and mapping (SLAM), especially in dense visual SLAM. However, existing works either rely on RGB-D sensors or require a separate monocular SLAM approach for camera tracking, and fail to produce high-fidelity 3D dense reconstructions. To address these shortcomings, we present NICER-SLAM, a dense RGB SLAM system that simultaneously optimizes for camera poses and a hierarchical neural implicit map representation, which also allows for high-quality novel view synthesis. To facilitate the optimization process for mapping, we integrate additional supervision signals including easy-to-obtain monocular geometric cues and optical flow, and also introduce a simple warping loss to further enforce geometric consistency. Moreover, to further boost performance in complex large-scale scenes, we also propose a local adaptive transformation from signed distance functions (SDFs) to density in the volume rendering equation. On multiple challenging indoor and outdoor datasets, NICER-SLAM demonstrates strong performance in dense mapping, novel view synthesis, and tracking, even competitive with recent RGB-D SLAM systems. Project page: https://nicer-slam.github.io/.

1. Introduction

Simultaneous localization and mapping (SLAM) is a fundamental computer vision problem with wide applications in autonomous driving, robotics, mixed reality, and more. Numerous dense visual SLAM methods have been developed over the years [35, 36, 48, 62, 63], offering real-time dense reconstructions of indoor scenes. However, most of these approaches rely on RGB-D sensors and fail on outdoor scenes or when depth sensors are not available. Moreover, these systems struggle with estimating plausible geometry in unobserved regions. A handful of dense monocular SLAM systems [2, 9, 73] have emerged in the deep learning era, taking sorely RGB sequences as input. They leverage their monocular depth prediction networks to somewhat fill in unobserved regions. Nevertheless, these systems are typically only applicable to small indoor scenes with limited camera movements.

The rapid advancements in neural implicit representations or neural fields [65] have demonstrated powerful performance in end-to-end differentiable dense visual SLAM. iMAP [53] first shows the potential of neural implicit representations in dense RGB-D SLAM, but it is only limited to room-size datasets. NICE-SLAM [75] introduces a hierarchical implicit encoding to perform mapping and camera

tracking in much larger indoor scenes. Although follow-up works [20, 23, 27, 31, 40, 67] build upon NICE-SLAM and iMAP from different angles, these methods still rely heavily on the depth input from RGB-D sensors, limiting their applicability to outdoor scenes.

Very recently, a handful of concurrent works (available as pre-prints) attempt to apply neural implicit representations for RGB-only SLAM [7, 46]. However, their tracking and mapping pipelines are independent of each other as they rely on different scene representations for these tasks. Both approaches directly depend on the state-of-the-art visual odometry methods [33, 58] for camera tracking, while using neural radiance fields (NeRFs) only for mapping. Moreover, they both only output and evaluate the rendered depth maps and color images, so no dense 3D model of a scene is produced. This raises an interesting research question:

Can we build a unified dense SLAM system with a neural implicit scene representation for both tracking and mapping from a monocular RGB video?

Compared to RGB-D SLAM, RGB-only SLAM is more challenging for multiple reasons. **1)** Depth ambiguity: Often multiple potential correspondences align well with the color observations, especially in textureless regions. Hence, stronger geometric priors are required for both mapping and tracking optimizations. **2)** Harder 3D reconstruction: The presence of ambiguity causes surface estimation to be less localized, leading to harder optimization and increased sampling efforts. **3)** Optimization convergence: The optimization is less constrained and more complex - resulting in slower convergence.

To tackle these challenges, we introduce *NICER-SLAM*, an implicit-based RGB SLAM system that is end-to-end optimizable for both accurate dense reconstruction and tracking in both indoor and outdoor environments. Additionally, our system also excels in novel view synthesis, but unlike NeRF, no camera poses (e.g., from separate SfM/SLAM systems like COLMAP) are required. Our key ideas are outlined as follows. First, we present coarse-to-fine hierarchical feature grids with small MLPs to model SDFs and colors, which yields detailed 3D reconstructions and high-fidelity renderings. Second, to facilitate the optimization of neural implicit map representations, we integrate additional supervision signals, including easy-to-obtain monocular geometric cues and optical flow. We also introduce a simple warping loss to further enhance geometry consistency. We observe that these regularizations significantly disambiguate optimization, enabling our framework to work robustly with only RGB input. Third, to better fit the sequential input for large-scale scenes, we propose a locally adaptive transformation from SDF to density.

In summary, we make the following contributions:

- We present NICER-SLAM, one of the first dense RGB-only SLAM that is end-to-end optimizable for both dense mapping and tracking, and also allows for high-quality novel view synthesis.
- We introduce a hierarchical neural implicit encoding for SDF representations, various geometric and motion regularizations, along with a locally adaptive SDF to volume density transformation. We demonstrate strong performances in mapping, and novel view synthesis and tracking on both indoor and outdoor datasets, even competitive with recent RGB-D SLAM methods.

2. Related Work

Dense Visual SLAM. SLAM is an active field in both industry and academia, especially in the past two decades. While sparse visual SLAM algorithms [14, 21, 33, 34] estimate accurate camera poses and only have sparse point clouds as the map representation, dense visual SLAM approaches focus on recovering a dense map of a scene. In general, dense map representations are categorized as either view-centric or world-centric. The first often represents 3D geometry as depth maps for keyframes, including the seminal work DTAM [36], and many follow-ups [2, 9, 22, 52, 56–59, 73, 74]. On the other hand, world-centric maps anchor the 3D geometry of a full scene in uniform world coordinates and represent as surfels [48, 63] or occupancies/TSDF values in the voxel grids [3, 11, 35, 38]. Our work also uses a world-centric map representation, but instead of explicitly representing surfaces, we store latent codes in multi-resolution voxel grids. This allows us to not only obtain high-quality geometry at low grid resolutions, but also attain plausible geometry estimation for unobserved regions.

Neural Implicit-based SLAM. Neural implicit representations [65] have delivered impressive results in numerous tasks, including reconstruction [4, 18, 28, 29, 37, 41, 43, 68], scene completion [19, 26, 42], novel view synthesis [30, 32, 45, 64, 72], etc. Regarding SLAM-related applications, some works [1, 6, 8, 25, 61, 70] attempt to jointly optimize a NeRF and camera poses, but they are limited to small objects or minor camera movements. A series of recent works [7, 46] relax such constraints, but rely on state-of-the-art SLAM systems like ORB-SLAM and DROID-SLAM to obtain camera poses, primarily focusing on novel view synthesis without producing 3D dense reconstruction.

iMAP [53] and NICE-SLAM [75] are the first two unified SLAM pipelines using neural implicit representations for both mapping and camera tracking. iMAP's application is limited to small scenes due to a single MLP as the scene representation, whereas NICE-SLAM handles much larger indoor environments using hierarchical feature grids and tiny MLPs. Many follow-up works improve upon these two works from various perspectives, including effi-

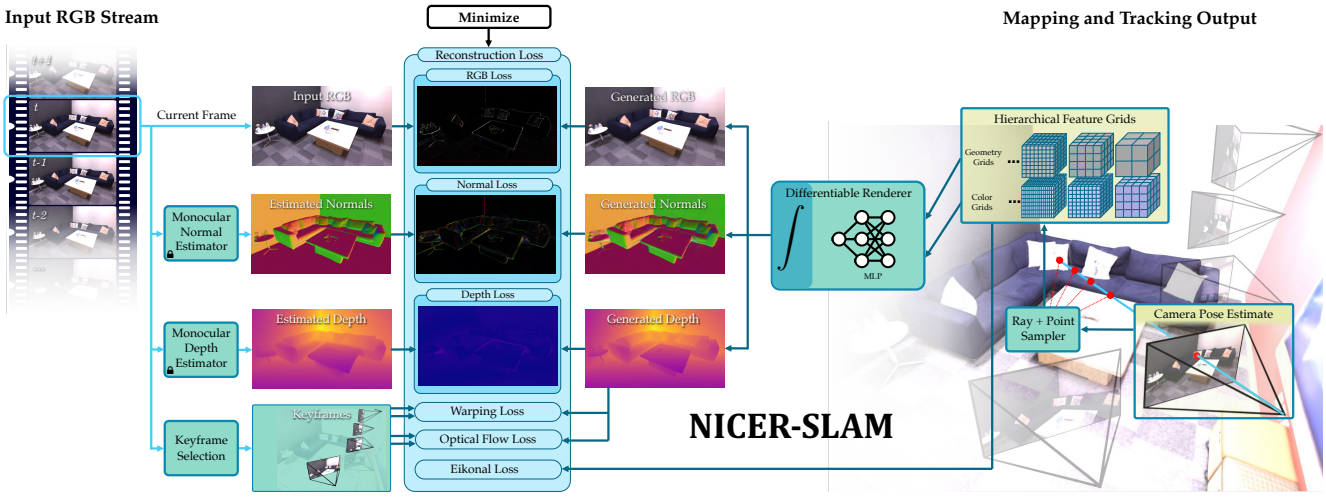

Figure 2. **System Overview.** Our method takes only an RGB stream as input and outputs both the camera poses as well as a learned hierarchical scene representation for geometry and colors. To realize an end-to-end joint mapping and tracking, we render predicted colors, depths, normals and optimize wrt. the input RGB and monocular cues. Moreover, we further enforce the geometric consistency with an RGB warping loss and an optical flow loss.

cient scene representation [20, 23], fast optimziation [67], add IMU measurements [27], or different shape representations [31, 40]. However, all of them require RGB-D inputs, limiting their outdoor applications or when only RGB sensors are accessible. In contrast, given only RGB sequences as input, our system provides high-quality 3D reconstruction and accurate camera poses simultaneously.

A concurrent work DIM-SLAM [24] presents a neural implicit-based RGB SLAM system in a similar spirit to ours. However, their use of simple color and warping losses leads to less robust results, requiring per-dataset parameter tuning and repeated experiment runs to achieve satisfactory results. In contrast, NICER-SLAM facilitates optimization by incorporating additional supervision signals, eliminating the need for any tuning across different datasets. In addition, this enhancement also enables us to achieve superior 3D reconstruction and novel view synthesis results.

3. Method

We provide an overview of the NICER-SLAM pipeline in Fig. 2. Taking an RGB video as input, we simultaneously estimate accurate 3D scene geometry and colors, as well as camera tracking all through an end-to-end optimization process. The scene geometry and appearance are represented using hierarchical neural implicit representations (Sec. 3.1). Leveraging the NeRF-like differentiable volume rendering, we render color, depth, and normal values for every pixel (Sec. 3.2), they facilitate end-to-end joint optimization for camera pose, scene geometry, and color (Sec. 3.3).

3.1. Hierarchical Neural Implicit Representations

Coarse-level Geometric Representation. The design of the coarse-level geometric representation is to efficiently model the coarse scene geometry (objects without geometric details) and the scene layout (e.g. walls, floors), even under partial observations. To achieve this, we represent the normalized scene as a dense voxel grid with a $32 \times 32 \times 32$ resolution, and maintain an optimizable 32-dim feature in each voxel. Given a 3D point $\mathbf{x} \in \mathbb{R}^3$ in the space, we use a small MLP f^{coarse} with a single 64-dim hidden layer to determine its base SDF value $s^{\text{coarse}} \in \mathbb{R}$ and a geometric feature $\mathbf{z}^{\text{coarse}} \in \mathbb{R}^{32}$ as:

$$s^{\text{coarse}},\ \mathbf{z}^{\text{coarse}} = f^{\text{coarse}}\big(\gamma(\mathbf{x}), \Phi^{\text{coarse}}(\mathbf{x})\big), \tag{1}$$

where γ is a fixed positional encoding [30, 55] that maps the coordinate to higher dimension. Following [68, 69, 71], we set the level for positional encoding to 6. $\Phi^{\text{coarse}}(\mathbf{x})$ represents the feature grid Φ^{coarse} tri-linearly interpolated at the point $\mathbf{x}$.

Fine-level Geometric Representation. Moving beyond the coarse-level representation, capturing the high-frequency geometric details of a scene is vital. We model these details as residual SDF values, utilizing multi-resolution feature grids and an MLP decoder [5, 32, 54, 75]. Specifically, we apply multi-resolution dense feature grids $\{\Phi_l^{\text{fine}}\}_1^L$ with respective resolutions R_l, as detailed in Eq. (2). These resolutions are sampled in geometric space [32] to combine features at different frequencies:

$$R_l := \lfloor R_{\min} b^l \rfloor, \quad b := \exp\left(\frac{\ln R_{\max} - \ln R_{\min}}{L-1}\right), \tag{2}$$

where $R_{\text{min}}, R_{\text{max}}$ correspond to the lowest and highest resolution, respectively. Here we consider $R_{\text{min}} = 32$, $R_{\text{max}} = 128$, in total $L = 8$ levels, with a feature dimension of 4 at each level.

Now, to model the residual SDF values for a point $\mathbf{x}$, we extract and concatenate the tri-linearly interpolated features at each level, and input them to an MLP f^{fine} with 3 hidden layers of size 64:

$$(\Delta s,\ \mathbf{z}^{\text{fine}}) = f^{\text{fine}}\big(\gamma(\mathbf{x}),\ \{\Phi_l^{\text{fine}}(\mathbf{x})\}\big)\ , \tag{3}$$

where $\mathbf{z}^{\text{fine}} \in \mathbb{R}^{32}$ is the geometric feature for $\mathbf{x}$ at the fine level. The final predicted SDF value $\hat{s}$ for $\mathbf{x}$ is obtained by adding the coarse-level base SDF value s^{coarse} to the fine-level residual SDF Δs:

$$\hat{s} = s^{\text{coarse}} + \Delta s\ . \tag{4}$$

Color Representation. Besides 3D geometry, we also predict color values such that our mapping and camera tracking can be optimized also with color losses. Moreover, as an additional application, we can also render images from novel views. Inspired by [32], we encode colors with another multi-resolution feature grid $\{\Phi_l^{\text{color}}\}_1^L$ and a decoder f^{color} parameterized with a 2-layer MLP of size 64. The number of feature grid levels is now $L = 16$, with a feature dimension of 2 at each level. We adjust the minimum and maximum resolution to $R_{\text{min}} = 16$ and $R_{\text{max}} = 2048$. The per-point color value is modelled as:

$$\hat{\mathbf{c}} = f^{\text{color}}\big(\mathbf{x}, \hat{\mathbf{n}}, \gamma(\mathbf{v}), \mathbf{z}^{\text{coarse}}, \mathbf{z}^{\text{fine}}, \{\Phi_l^{\text{color}}(\mathbf{x})\}\big)\ . \tag{5}$$

where $\hat{\mathbf{n}}$ refers to the normal at point $\mathbf{x}$ calculated from $\hat{s}$ in Eq. (4) and $\gamma(\mathbf{v})$ is the viewing direction with positional encoding with a level of 4, following [69, 71].

3.2. Volume Rendering

Following recent works on implicit-based 3D reconstruction [39, 60, 69] and dense visual SLAM [53, 75], we optimize our scene representation from Sec. 3.1 using differentiable volume rendering. For rendering a pixel, we cast a ray $\mathbf{r}$ from the camera center $\mathbf{o}$ through the pixel along its normalized view direction $\mathbf{v}$. N points are then sampled along the ray, denoted as $\mathbf{x}_i = \mathbf{o} + t_i\mathbf{v}$, and their predicted SDFs and color values are $\hat{s}_i$ and $\hat{\mathbf{c}}_i$. We transform the SDFs $\hat{s}_i$ to density values σ_i as in [69]:

$$\sigma_\beta(s) = \begin{cases} \frac{1}{2\beta}\exp\left(\frac{s}{\beta}\right) & \text{if } s \le 0 \\ \frac{1}{\beta}\left(1 - \frac{1}{2}\exp\left(-\frac{s}{\beta}\right)\right) & \text{if } s > 0 \end{cases}\ , \tag{6}$$

where $\beta \in \mathbb{R}$ is a learnable parameter. As in [30], we calculate the color $\hat{C}$ for the current ray $\mathbf{r}$ as:

$$\hat{C} = \sum_{i=1}^{N} T_i\,\alpha_i\,\hat{\mathbf{c}}_i \qquad T_i = \prod_{j=1}^{i-1}(1-\alpha_j)$$
$$\alpha_i = 1 - \exp\left(-\sigma_i\delta_i\right)\ , \tag{7}$$

where T_i and α_i correspond to transmittance and alpha value of sample point i along ray $\mathbf{r}$, and δ_i is the distance between neighboring sample points. In a similar manner, we also compute the depth $\hat{D}$ and normal $\hat{N}$ of the surface intersecting the ray as:

$$\hat{D} = \sum_{i=1}^{N} T_i\,\alpha_i\,t_i \qquad \hat{N} = \sum_{i=1}^{N} T_i\,\alpha_i\,\hat{\mathbf{n}}_i\ . \tag{8}$$

Locally Adaptive Transformation. The parameter β in Eq. (6) serves as a modifier of the smoothing amount around an object's surface during the volume rendering process. As the network gains certainty about the object's surface, the value of β gradually decreases, leading to sharper and faster reconstructions. VolSDF [69] models β as a global parameter for small object-level scenes. However, for our application involving sequential input within complex indoor and outdoor scenes, a globally optimizable β proves to be sub-optimal (ablation study in supplementary).

Instead, we propose to assign β values *locally* to model locally adaptive transformation in Eq. (6). More specifically, we partition the scene into a voxel grid and maintain a counter to track the number of point samples in it during mapping. We set the grid size to 64^3 (see ablation in supplementary). Next, we employ a heuristic approach to convert the local point counter T_p into the β value:

$$\beta = c_0 \cdot \exp(-c_1 \cdot T_p) + c_2\ . \tag{9}$$

This transformation was derived by correlating the decreasing trend of β wrt. the voxel count under the setting of global input as in [69, 71], and fitting an exponential curve. The curve fitting results are illustrated in the supplemental.

3.3. End-to-End Joint Mapping and Tracking

From purely sequential RGB input, end-to-end joint mapping and tracking present significant challenges. This is due to the high degrees of ambiguity particularly in large complex scenes with many textureless and sparsely covered regions. To enable this process under our neural scene representation, we propose to constrain the optimization with the following losses.

RGB Rendering Loss. Eq. (7) connects the 3D neural scene representation with 2D observations, allowing us to optimize the scene representation with a simple RGB reconstruction loss:

$$\mathcal{L}_{\text{rgb}} = \sum_{\mathbf{r}\in\mathcal{R}} \left\|\hat{C}(\mathbf{r}) - C(\mathbf{r})\right\|_1\ , \tag{10}$$

$\mathcal{R}$ are randomly sampled pixels, and C is the pixel color.

RGB Warping Loss. To enforce geometry consistency from only color inputs, we utilize a simple per-pixel warping loss. For any pixel in frame m, denoted as $\mathbf{r}_m$, we first

render its depth value using Eq. (8) and unproject it to 3D. We then project it to the nearby keyframe n using intrinsics and extrinsics of frame n. The projected pixel in frame n is denoted as $\mathbf{r}_{m\to n}$. The warping loss is defined as:

$$\mathcal{L}_{\text{warp}} = \sum_{\mathbf{r}_m \in \mathcal{R}} \sum_{n \in \mathcal{K}_m} \| C(\mathbf{r}_m) - C(\mathbf{r}_{m\to n}) \|_1 \ , \quad (11)$$

where $\mathcal{K}_m$ denotes the keyframe list for the current frame m, excluding frame m itself. We mask out the pixels that are projected outside the image boundary of frame n. Note that unlike [12] that optimize neural implicit surfaces with patch warping, we observe that simply performing warping on randomly sampled pixels is more efficient without performance drop.

Optical Flow Loss. The RGB rendering and warping loss are only point-wise terms that are prone to local minima. Therefore, we incorporate regional smoothness priors via optical flow estimates. Suppose the sample pixel in frame m as $\mathbf{r}_m$ and the projected pixel as $\mathbf{r}_n$, the loss will be:

$$\mathcal{L}_{\text{flow}} = \sum_{\mathbf{r}_m \in \mathcal{R}} \sum_{n \in \mathcal{K}_m} \| (\mathbf{r}_m - \mathbf{r}_n) - \text{GM}(\mathbf{r}_{m\to n}) \|_1 \ , \quad (12)$$

$\text{GM}(\mathbf{r}_{m\to n})$ denotes the estimated optical flow from [66].

Monocular Depth Loss. Given RGB input, one can easily obtain geometric cues (such as depths or normals) via an off-the-shelf monocular predictor [13]. Inspired by [71], we also include this information in the optimization to guide the neural implicit surface reconstruction. More specifically, to enforce depth consistency between our rendered expected depths $\hat{D}$ and the monocular depths $\bar{D}$, we use the loss [44]:

$$\mathcal{L}_{\text{depth}} = \sum_{\mathbf{r} \in \mathcal{R}} \| (w\hat{D}(\mathbf{r}) + q) - \bar{D}(\mathbf{r}) \|^2 \ , \quad (13)$$

where $w, q \in \mathbb{R}$ are the scale and shift used to align $\hat{D}$ and $\bar{D}$, since $\bar{D}$ is only known up to an unknown scale. We solve for w and q per image with least squares, which has a closed-form solution.

Monocular Normal Loss. Another geometric cue that is complementary to the monocular depth is surface normal, a local cue that captures more geometric details. Similar to [71], we impose consistency on the volume-rendered normal $\hat{N}$ and the monocular normals $\bar{N}$ from [13]:

$$\begin{aligned} \mathcal{L}_{\text{normal}} = & \sum_{\mathbf{r} \in \mathcal{R}} \| \hat{N}(\mathbf{r}) - \bar{N}(\mathbf{r}) \|_1 \\ & + \| 1 - \hat{N}(\mathbf{r})^\top \bar{N}(\mathbf{r}) \|_1 \ . \end{aligned} \quad (14)$$

Eikonal Loss. In addition, we add the Eikonal loss [16] to regularize the output SDF values $\hat{s}$:

$$\mathcal{L}_{\text{eikonal}} = \sum_{\mathbf{x} \in \mathcal{X}} (\| \nabla \hat{s}(\mathbf{x}) \|_2 - 1)^2 \ , \quad (15)$$

where $\mathcal{X}$ represents a set of uniformly sampled near-surface points.

Optimization Scheme. We provide details on how to optimize the scene geometry and appearance in the form of our hierarchical representation, and also the camera poses.

***Mapping*:** To optimize the scene representation mentioned in Sec. 3.1, we uniformly sample M pixels/rays in total from the current frame and selected keyframes. The optimization is performed in a 3-stage process similar to [75] but employs the following loss:

$$\begin{aligned} \mathcal{L} = & \mathcal{L}_{\text{rgb}} + 0.5\mathcal{L}_{\text{warp}} + 0.001\mathcal{L}_{\text{flow}} \\ & + 0.1\mathcal{L}_{\text{depth}} + 0.05\mathcal{L}_{\text{normal}} + 0.1\mathcal{L}_{\text{eikonal}} \end{aligned} \quad (16)$$

At the first stage, we treat the coarse-level base SDF value s^{coarse} in Eq. (1) as the final SDF value $\hat{s}$, and optimize the coarse feature grid Φ^{coarse}, coarse MLP parameters of f^{coarse}, and color MLP parameters of f^{color} with Eq. (16). Upon reaching 25% of the total number of iterations, we switch to using Eq. (4) as the final SDF value, soenabling the joint optimization of the fine-level feature grids $\{\Phi_l^{\text{fine}}\}$ and fine-level MLP f^{fine}. At 75% mark, we conduct a local bundle adjustment (BA) with Eq. (16), extending the optimization to color feature grids $\{\Phi_l^{\text{color}}\}$ and the extrinsic parameters of K selected mapping frames.

***Camera Tracking*:** In parallel to mapping, we optimize the camera pose (rotation and translation) of the current frame, while keeping the hierarchical scene representation fixed. This is achieved by sampling M_t pixels from the current frame and use purely the RGB rendering loss in Eq. (10) for 100 iterations.

4. Experiments

We evaluate qualitative and quantitative comparisons against state-of-the-art (SOTA) SLAM frameworks on both synthetic and real-world datasets in Sec. 4.1. A comprehensive ablation study supporting our design choices is provided in the supplementary material.

Datasets. We evaluate on the synthetic Replica dataset [50], where RGB-(D) images are rendered with the official renderer. To assess the performance in real-world indoor/outdoor scenarios, we also compare on the challenging dataset 7-Scenes [49] known for its low-resolution images with severe motion blur, and a self-captured outdoor (SCO) dataset, captured with Azure Kinect, comprising 6 diverse scenes, ranging from 800 to 2700 frames. COLMAP is used to obtain intrinsic parameters for the 7-Scenes and SCO datasets. For comparisons and discussions on ScanNet and TUM RGB-D dataset, see the supplementary material.

Baselines. We compare NICER-SLAM with 10 methods. (a) SOTA neural implicit RGB-D SLAM system

		rm-0	rm-1	rm-2	off-0	off-1	off-2	off-3	off-4	Avg.
RGB-D input										
NICE-SLAM	Acc.[cm]↓	3.53	3.60	**3.03**	5.56	3.35	4.71	3.84	3.35	3.87
	Comp.[cm]↓	3.40	3.62	**3.27**	**4.55**	**4.03**	**3.94**	3.99	4.15	**3.87**
	Comp.Rat.[<5cm %]↑	86.05	80.75	**87.23**	79.34	**82.13**	80.35	80.55	82.88	82.41
	Normal Cons.[%]↑	91.92	91.36	90.79	89.30	88.79	88.97	87.18	91.17	89.93
Vox-Fusion	Acc.[cm]↓	**2.53**	**1.69**	3.33	**2.20**	**2.21**	**2.72**	4.16	**2.48**	**2.67**
	Comp.[cm]↓	**2.81**	**2.51**	4.03	8.75	7.36	4.19	**3.26**	**3.49**	4.55
	Comp.Rat.[<5cm %]↑	**91.52**	**91.34**	86.78	**81.99**	82.03	**85.45**	**87.13**	**86.53**	**86.59**
	Normal Cons.[%]↑	**94.14**	**93.28**	91.71	**90.52**	**88.95**	**91.54**	91.03	**92.67**	**91.73**
RGB input										
COLMAP	Acc.[cm]↓	3.87	27.29	5.41	5.21	12.69	4.28	5.29	5.45	8.69
	Comp.[cm]↓	4.78	23.90	17.42	12.98	12.35	4.96	16.17	4.41	12.12
	Comp.Rat.[<5cm %]↑	83.08	22.89	64.47	72.59	69.52	81.12	64.38	82.92	67.62
	Normal Cons.[%]↑	72.49	60.10	69.42	69.91	74.04	71.84	71.49	71.75	70.13
TANDEM	Acc.[cm]↓	6.76	7.81	5.60	5.00	4.66	10.68	7.34	6.97	6.85
	Comp.[cm]↓	9.00	7.99	12.27	15.30	14.46	12.63	10.50	10.38	11.57
	Comp.Rat.[<5cm %]↑	52.81	56.58	55.71	57.88	54.18	49.19	44.82	47.60	52.35
	Normal Cons.[%]↑	78.26	77.73	82.06	82.14	79.48	79.73	79.68	82.80	80.24
NeRF-SLAM	Acc.[cm]↓	11.84	10.62	11.86	9.32	14.40	11.54	16.31	11.11	12.13
	Comp.[cm]↓	5.63	5.88	9.22	13.29	10.17	6.95	7.81	5.26	8.03
	Comp.Rat.[<5cm %]↑	61.13	68.19	47.85	37.64	56.17	66.20	55.67	61.86	56.84
	Normal Cons.[%]↑	63.39	53.31	57.52	64.09	57.13	57.06	59.73	58.59	58.85
DIM-SLAM*	Acc.[cm]↓	14.19	9.56	8.41	10.16	7.86	16.50	13.01	13.08	11.60
	Comp.[cm]↓	6.24	6.45	12.17	5.95	8.33	8.28	6.77	8.62	7.85
	Comp.Rat.[<5cm %]↑	69.77	66.30	51.21	74.16	62.10	54.92	63.88	55.43	62.22
	Normal Cons.[%]↑	77.69	82.16	78.89	81.44	79.41	73.68	77.09	78.05	78.55
DROID-SLAM	Acc.[cm]↓	12.18	8.35	3.26	3.01	2.39	5.66	4.49	4.65	5.50
	Comp.[cm]↓	8.96	6.07	16.01	16.19	16.20	15.56	9.73	9.63	12.29
	Comp.Rat.[<5cm %]↑	60.07	76.20	61.62	64.19	60.63	56.78	61.95	67.51	63.62
	Normal Cons.[%]↑	72.81	74.71	79.21	77.53	78.57	75.79	77.69	76.38	76.59
NICER-SLAM	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.[<5cm %]↑	88.75	76.61	86.10	65.19	77.84	74.51	82.01	83.98	79.37
	Normal Cons.[%]↑	93.00	91.52	**92.38**	87.11	86.79	90.19	90.10	90.96	90.27

Table 1. **Reconstruction Results on the Replica dataset.** Best results are highlighted as **first**, second, and third. NICER-SLAM performs the best among RGB SLAM methods, and is on par with RGB-D methods. DIM-SLAM* indicates our re-implementation. Note that we do not report the numbers from DIM-SLAM paper because they cull meshes differently. Please refer to the supp. mat. for discussion.

			rm-0	rm-1	rm-2	off-0	off-1	off-2	off-3	off-4	Avg.
RGB-D input											
NICE-SLAM	Extra-polate	PSNR ↑	23.83	22.61	21.97	25.78	**25.30**	18.50	22.82	**25.26**	23.26
		SSIM ↑	0.788	0.813	0.858	**0.887**	**0.842**	0.826	0.862	0.875	0.844
		LPIPS ↓	0.284	0.249	0.218	0.209	**0.145**	0.242	0.190	**0.191**	0.216
	Inter-polate	PSNR ↑	22.12	22.47	24.52	29.07	30.34	19.66	22.23	24.94	24.42
		SSIM ↑	0.689	0.757	0.814	0.874	**0.886**	0.797	0.801	0.856	0.809
		LPIPS ↓	0.330	0.271	0.208	0.229	0.181	0.235	0.209	0.198	0.233
Vox-Fusion	Extra-polate	PSNR ↑	23.45	20.83	18.38	23.28	24.48	17.50	23.06	24.84	21.98
		SSIM ↑	0.765	0.773	0.747	0.751	0.762	0.727	0.824	0.851	0.775
		LPIPS ↓	0.280	0.272	0.282	0.235	0.169	0.292	0.232	0.212	0.247
	Inter-polate	PSNR ↑	22.39	22.36	23.92	27.79	29.83	20.33	23.47	25.21	24.41
		SSIM ↑	0.683	0.751	0.798	0.857	0.876	0.794	0.803	0.847	0.801
		LPIPS ↓	0.303	0.269	0.234	0.241	0.184	0.243	0.213	0.199	0.236
RGB input											
NeRF-SLAM	Extra-polate	PSNR ↑	17.34	19.00	15.18	17.50	19.59	12.79	13.97	17.66	16.63
		SSIM ↑	0.699	0.738	0.642	0.704	0.672	0.639	0.718	0.787	0.700
		LPIPS ↓	0.301	**0.228**	0.242	0.289	0.187	0.295	0.298	0.254	0.262
	Inter-polate	PSNR ↑	16.45	19.62	21.17	21.44	20.86	15.49	15.11	18.96	18.64
		SSIM ↑	0.576	0.700	0.754	0.773	0.747	0.731	0.688	0.790	0.720
		LPIPS ↓	0.330	**0.177**	**0.170**	0.335	0.229	0.251	0.282	0.241	0.252
DIM-SLAM*	Extra-polate	PSNR ↑	21.03	21.38	17.38	24.82	24.96	17.34	19.54	21.43	20.99
		SSIM ↑	0.702	0.768	0.698	0.822	0.803	0.680	0.752	0.783	0.751
		LPIPS ↓	0.372	0.303	0.344	0.251	0.182	0.362	0.311	0.295	0.303
	Inter-polate	PSNR ↑	18.48	**26.19**	24.95	**30.16**	**31.75**	21.36	21.22	23.65	24.72
		SSIM ↑	0.622	0.765	0.788	0.856	0.882	0.744	0.751	0.797	0.776
		LPIPS ↓	0.422	0.283	0.291	0.234	0.185	0.304	0.293	0.256	0.284
DROID-SLAM	Extra-polate	PSNR ↑	18.25	18.65	13.49	16.13	10.31	14.78	15.53	15.71	15.36
		SSIM ↑	0.737	0.793	0.786	0.760	0.650	0.800	0.797	0.800	0.765
		LPIPS ↓	0.352	0.283	0.299	0.298	0.286	0.300	0.302	0.311	0.304
	Inter-polate	PSNR ↑	21.41	24.04	22.08	23.59	21.29	20.64	20.22	20.22	21.69
		SSIM ↑	0.693	**0.786**	0.826	**0.868**	0.863	**0.828**	0.808	0.819	0.812
		LPIPS ↓	0.329	0.270	0.228	0.232	0.207	0.231	0.234	0.237	0.246
NICER-SLAM	Extra-polate	PSNR ↑	**25.64**	**23.69**	**22.62**	**25.88**	22.56	**21.46**	**24.42**	25.15	**23.93**
		SSIM ↑	**0.810**	**0.820**	**0.871**	0.885	0.828	**0.863**	**0.888**	**0.887**	**0.857**
		LPIPS ↓	**0.254**	0.233	**0.200**	**0.193**	0.160	**0.203**	**0.175**	0.192	**0.201**
	Inter-polate	PSNR ↑	**25.33**	23.92	**26.12**	28.54	25.86	**21.95**	**26.13**	**25.47**	**25.41**
		SSIM ↑	**0.751**	0.771	**0.831**	0.866	0.852	0.820	**0.856**	**0.865**	**0.827**
		LPIPS ↓	**0.250**	0.215	0.176	**0.172**	**0.178**	**0.195**	**0.162**	**0.177**	**0.191**

Table 2. **Novel View Synthesis Evaluation on the Replica dataset.** Best results are shown as **first**, second, and third. We outperform all baselines, including those utilizing depth inputs. Note that while NeRF-SLAM reported excellent rendering performance on training views, its large tracking errors lead to its less impressive performance in test views.

NICE-SLAM [75] and Vox-Fusion [67], (b) concurrent neural implicit RGB SLAM system DIM-SLAM [24]/DIM-SLAM*[1], NeRF-SLAM [46] and Orbeez-SLAM [7], (c) classic SLAM methods COLMAP [47] and DSO [14], and (d) SOTA dense monocular SLAM systems DROID-SLAM [58] and TANDEM [22]. For camera tracking evaluation, we also compare with DROID-SLAM*, which does not perform the final global bundle adjustment and loop closure (identical to our NICER-SLAM setting). For DROID-SLAM's 3D reconstruction, we run TSDF fusion with their predicted depths of keyframes.

Metrics. For camera tracking, we follow the conventional monocular SLAM evaluation pipeline where the estimated trajectory is aligned to the GT using `evo` [17], and then evaluate (*ATE RMSE*) [51]. To evaluate scene geometry, we use *Accuracy*, *Completion*, *Completion Ratio*, and *Normal Consistency*. The reconstructed meshes from monocular SLAM systems are aligned to the GT mesh using the ICP tool from [15]. We also use PSNR, SSIM and LPIPS for novel view synthesis evaluation.

[1]DIM-SLAM code was only partially released upon the submission deadline. We faithfully re-implemented the entire pipeline with extensive discussion with the authors.

4.1. Mapping, Tracking and Rendering Evaluations

Evaluation on Replica [50]. In evaluating scene geometry on Replica (see Table 1), our method surpasses all RGB-only methods by a noticeable margin, and even shows competitive results against RGB-D SLAM approaches like NICE-SLAM and Vox-Fusion. Note that the recent works NeRF-SLAM/Orbeez-SLAM/TANDEM use DROID-SLAM/ORB-SLAM2/DSO as tracking frontends to obtain more accurate camera poses, but our system still yields better reconstructions even with our simple end-to-

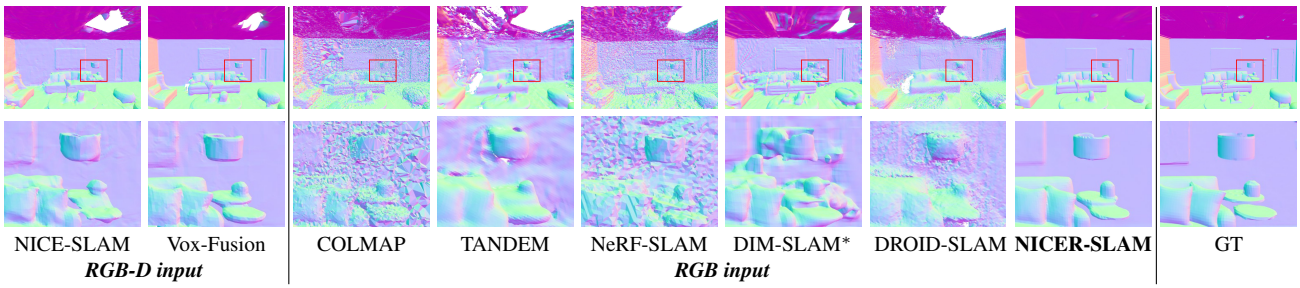

Figure 3. **3D Reconstruction Results on the Replica Dataset [50].** The second row shows zoom-in views for better comparison.

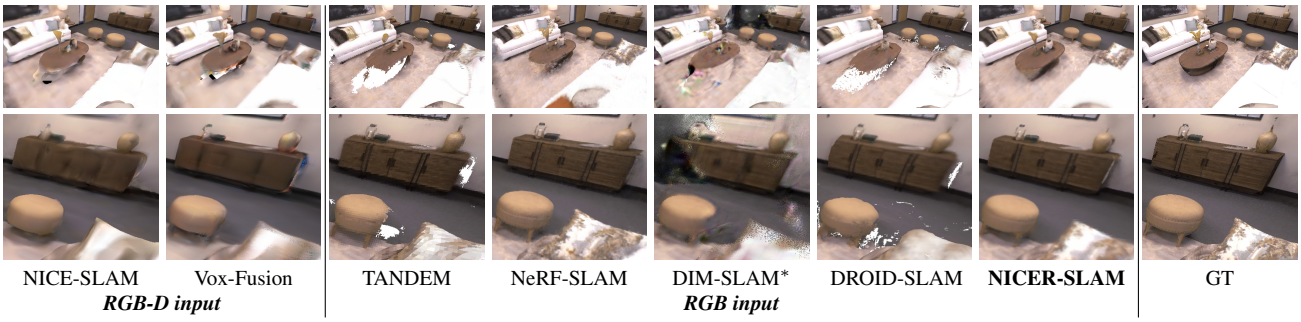

Figure 4. **Novel View Synthesis Results on the Replica the Dataset [50].** The second row shows zoom-in renderings for better comparison. Note that we selected novel viewpoints far from the training views (extrapolation).

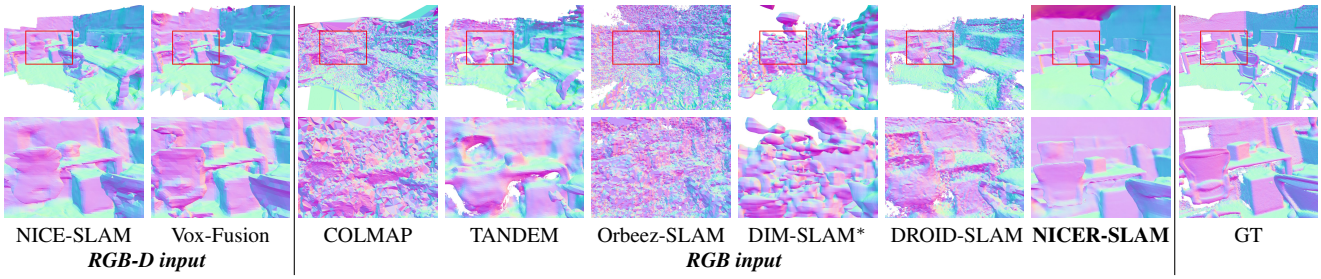

Figure 5. **3D Reconstruction Results on the 7-Scenes Dataset [49].** The second row shows zoomed-in normal maps. It is apparent that the quality of the scene representation is substantially worse for the other RGB-based methods despite their better tracking accuracy.

end tracking. Fig. 3 shows that NICER-SLAM can produce the most visually appealing reconstructions. Note that for DIM-SLAM, due to their use of simple warping loss, they reconstruct those textureless regions wrongly. Moreover, their use of occupancy also creates floaters, requiring post-processing before evaluation (see supp. mat.).

For camera tracking, we can see from Table 3 that SLAM systems designed for tracking (e.g. DROID-SLAM and DSO) outperform all other methods. Nevertheless, even though tracking is not the focus of our method, we are still on par with NICE-SLAM (1.88 vs 1.95 cm on average), while no depth information is used as additional input.

Even with the less accurate camera poses from our simple tracking pipeline, NICER-SLAM still produces visually more compelling and complete novel-view rendering results than baseline methods, even those using depth inputs, see Fig. 4, Fig. 1 and Table 2. Classic methods like COLMAP and DROID-SLAM cannot render missing regions. Neural-implicit approaches like NICE-SLAM, Vox-Fusion and DIM-SLAM fill in missing areas but their ren-

	rm-0	rm-1	rm-2	off-0	off-1	off-2	off-3	off-4	Avg.
RGB-D input									
NICE-SLAM	1.69	2.04	1.55	0.99	0.90	1.39	3.97	3.08	1.95
Vox-Fusion	0.27	1.33	0.47	0.70	1.11	0.46	0.26	0.58	0.65
RGB input									
COLMAP	0.62	23.7	0.39	0.33	0.24	0.79	**0.14**	1.73	3.49
TANDEM	0.54	0.43	0.47	0.61	0.33	5.42	0.68	0.75	1.15
DSO	**0.26**	0.25	**0.19**	0.38	0.20	2.53	0.22	**0.38**	0.55
Orbeez-SLAM	0.34	0.41	0.27	0.36	F	F	0.294	2.89	0.76
NeRF-SLAM	17.26	11.94	15.76	12.75	10.34	14.52	20.32	14.96	14.73
DIM-SLAM	0.48	0.78	0.35	0.67	0.37	0.36	0.33	0.36	0.46
DIM-SLAM*	1.06	0.49	0.32	0.43	0.26	0.65	0.55	3.69	0.93
DROID-SLAM	0.34	**0.13**	0.27	**0.25**	0.42	**0.32**	0.52	0.40	**0.33**
DROID-SLAM*	0.58	0.58	0.38	1.06	0.40	0.70	0.53	1.33	0.70
NICER-SLAM	1.36	1.60	1.14	2.12	3.23	2.12	1.42	2.01	1.88

Table 3. **Camera Tracking Results on the Replica Dataset.** ATE RMSE [cm] (↓) is the evaluation metric. DIM-SLAM* is our re-implementation, and DROID-SLAM* has no global BA and loop closure. "F" denotes program failure or final trajectory unable to align with GT (SVD decomposition error).

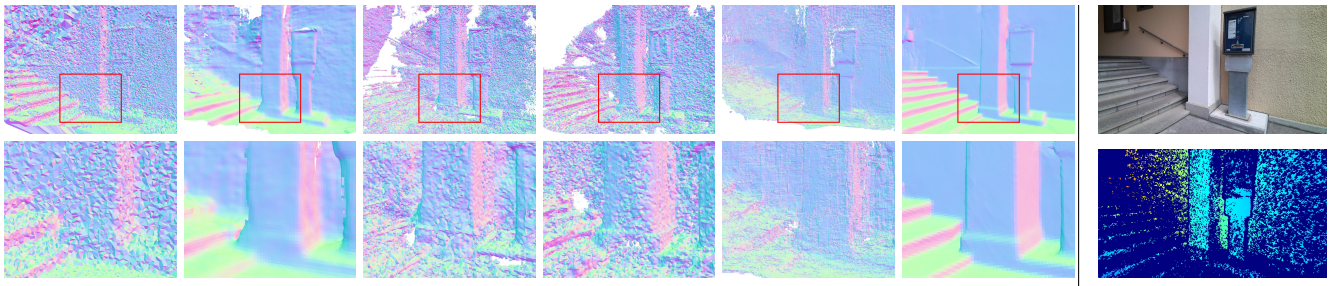

Figure 6. **3D Reconstruction Results on the Self-Captured Outdoor (SCO) Dataset.** The second row shows zoomed-in normal maps for better comparison. Note that only RGB images are used as input, while the depth image is only for visualization, showing the captured depth is unable to provide reliable readings in outdoor environments.

Figure 7. **Novel View Synthesis Results on our Self-captured Outdoor Dataset.** We can obtain visually compelling and more complete results than other SLAM methods, and perform similarly to NeRF-SLAM, which primarily focuses on novel view synthesis and relies on DROID-SLAM for camera tracking.

derings are normally over-smooth. We can faithfully render high-fidelity novel views even when those views are far from the training views. It is worth noting that our renderings are visually on par with NeRF-SLAM, a system primarily dedicated to novel view synthesis. NICER-SLAM also achieves higher metrics due to lower tracking errors.

Evaluation on 7-Scenes [49]. We also evaluate the challenging real-world dataset 7-Scenes to benchmark the robustness of different methods when the input images are of low resolutions and have severe motion blurs. For geometry illustrated in Fig. 5, NICER-SLAM produces sharper and more detailed geometry over all baselines. For tracking, please check the supplementary material.

Evaluation on Self-Captured Outdoor Dataset. We further extend the evaluation to outdoor scenarios with a self-captured outdoor dataset, where an Azure Kinect camera is used to capture diverse outdoor scenes. Note that the captured depths are unable to provide reliable readings in the outdoor environment (see Fig. 6), so we only compare among monocular SLAM approaches. DIM-SLAM struggles even at the initialization stage, due to its simple warping loss's inability to effectively handle large textureless regions. As can be seen Fig. 6, except for TANDEM, all other baseline methods cannot reconstruct detailed geometry due to the lack of textures in the scene. NICER-SLAM can not only handle textureless regions and reconstruct flat surfaces like walls, but also captures small details, e.g. the handrail. Fig. 7 shows novel view synthesis results. NICER-SLAM produces compelling and more complete renderings than other methods. We also perform similarly to the dedicated view synthesis system, NeRF-SLAM, but without the need of leveraging DROID-SLAM for tracking.

5. Conclusions

We present NICER-SLAM, a novel dense RGB SLAM system that enables highly accurate 3D reconstruction with realistic appearances. In contrast to vanilla NeRF, it doesn't require camera poses as input and instead jointly optimizes poses and neural implicit maps in an end-to-end manner. Extensive experiments in both indoor and outdoor scenes demonstrate the effectiveness of NICER-SLAM, especially in surface reconstruction and novel view synthesis.

Limitations. Although we show benefits over SLAM methods using traditional scene representations in terms of mapping and novel view synthesis, our pipeline is currently not yet optimized for real-time operations. Implementing Instant-NGP [32]-like CUDA solutions and performing on-the-fly empty space skipping are straightforward strategies for potential optimization. Moreover, we currently do not perform loop closure, which will yield further improvements in tracking performance.

Acknowledgements. This project is partially supported by the SONY Research Award Program and a research grant by FIFA. The authors thank the Max Planck ETH Center for Learning Systems (CLS) for supporting Songyou Peng and the strategic research project ELLIIT for supporting Viktor Larsson. We thank Zehao Yu, Yiming Zhao, Weicai Ye, Boyang Sun, Jianhao Zheng, and Heng Li for helpful discussions.

References

[1] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. *arXiv preprint arXiv:2212.07388*, 2022. 2

[2] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J. Davison. Codeslam - learning a compact, optimisable representation for dense visual SLAM. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2560–2568, Salt Lake City, UT, USA, 2018. Computer Vision Foundation / IEEE Computer Society. 1, 2

[3] Erik Bylow, Jürgen Sturm, Christian Kerl, Fredrik Kahl, and Daniel Cremers. Real-time camera tracking and 3d reconstruction using signed distance functions. In *Robotics: Science and Systems (RSS)*, page 2, 2013. 2

[4] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5939–5948, 2019. 2

[5] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6970–6981, 2020. 3

[6] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 264–280. Springer, 2022. 2

[7] Chi-Ming Chung, Yang-Che Tseng, Ya-Ching Hsu, Xiang-Qian Shi, Yun-Hung Hua, Jia-Fong Yeh, Wen-Chin Chen, Yi-Ting Chen, and Winston H Hsu. Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9400–9406. IEEE, 2023. 2, 6

[8] Ronald Clark. Volumetric bundle adjustment for online photorealistic scene capture. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6124–6132, 2022. 2

[9] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J Davison. Deepfactors: Real-time probabilistic dense monocular slam. *IEEE Robotics and Automation Letters*, 5 (2):721–728, 2020. 1, 2

[10] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4, 5

[11] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Trans. on Graphics*, 36(4):1, 2017. 2

[12] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6260–6269, 2022. 5

[13] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 10786–10796, 2021. 5

[14] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 40(3):611–625, 2017. 2, 6, 4

[15] Daniel Girardeau-Montaut. Cloudcompare. *France: EDF R&D Telecom ParisTech*, 11, 2016. 6

[16] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 5, 1

[17] Michael Grupp. evo: Python package for the evaluation of odometry and slam. https://github.com/MichaelGrupp/evo, 2017. 6

[18] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5511–5520, 2022. 2

[19] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6001–6010, 2020. 2

[20] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. *arXiv preprint arXiv:2211.11704*, 2022. 2, 3

[21] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 225–234. IEEE, 2007. 2

[22] Lukas Koestler, Nan Yang, Niclas Zeller, and Daniel Cremers. Tandem: Tracking and dense mapping in real-time using deep multi-view stereo. In *Proc. Conf. on Robot Learning (CoRL)*, pages 34–45. PMLR, 2022. 2, 6

[23] Evgenii Kruzhkov, Alena Savinykh, Pavel Karpyshev, Mikhail Kurenkov, Evgeny Yudin, Andrei Potapov, and Dzmitry Tsetserukou. Meslam: Memory efficient slam based on neural fields. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 430–435. IEEE, 2022. 2, 3

[24] Heng Li, Xiaodong Gu, Weihao Yuan, Luwei Yang, Zilong Dong, and Ping Tan. Dense rgb slam with neural implicit maps. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2023. 3, 6, 1, 4

[25] C. Lin, W. Ma, A. Torralba, and S. Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 2

[26] Stefan Lionar, Daniil Emtsev, Dusan Svilarkovic, and Songyou Peng. Dynamic plane convolutional occupancy networks. In *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1829–1838, 2021. 2

[27] Daniil Lisus and Connor Holmes. Towards open world nerf-based slam. *arXiv preprint arXiv:2301.03102*, 2023. 2, 3

[28] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2019–2028, 2020. 2

[29] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4460–4470, 2019. 2

[30] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 2, 3, 4

[31] Yuhang Ming, Weicai Ye, and Andrew Calway. idf-slam: End-to-end rgb-d slam with neural implicit mapping and deep feature tracking. *arXiv preprint arXiv:2209.07919*, 2022. 2, 3

[32] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. on Graphics*, 41(4), 2022. 2, 3, 4, 8

[33] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017. 2, 4

[34] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 2

[35] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011. 1, 2

[36] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 2320–2327. IEEE, 2011. 1, 2

[37] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[38] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Trans. on Graphics*, 32(6):1–11, 2013. 2

[39] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 5589–5599, 2021. 4

[40] Joseph Ortiz, Alexander Clegg, Jing Dong, Edgar Sucar, David Novotny, Michael Zollhoefer, and Mustafa Mukadam. isdf: Real-time neural signed distance fields for robot perception. In *Robotics: Science and Systems (RSS)*, 2022. 2, 3

[41] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 2

[42] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 523–540. Springer, 2020. 2

[43] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:13032–13044, 2021. 2

[44] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2020. 5

[45] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 14335–14345, 2021. 2

[46] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. *arXiv preprint arXiv:2210.13641*, 2022. 2, 6

[47] J. L. Schönberger and J. M. Frahm. Structure-from-motion revisited. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 6

[48] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. BAD SLAM: bundle adjusted direct RGB-D SLAM. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 134–144, 2019. 1, 2

[49] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013. 5, 7, 8, 1

[50] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. R., S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 1, 5, 6, 7, 2

[51] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2012. 6, 4, 5

[52] Edgar Sucar, Kentaro Wada, and Andrew Davison. Nodeslam: Neural object descriptors for multi-view shape

reconstruction. In *Proc. of the International Conf. on 3D Vision (3DV)*, pages 949–958. IEEE, 2020. 2

[53] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 6229–6238, 2021. 1, 2, 4

[54] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 11358–11367, 2021. 3

[55] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

[56] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2019. 2

[57] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.

[58] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *Advances in Neural Information Processing Systems*, pages 16558–16569, 2021. 1, 2, 6

[59] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5038–5047, 2017. 2

[60] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 4

[61] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2

[62] Thomas Whelan, Michael Kaess, Maurice Fallon, Hordur Johannsson, John Leonard, and John McDonald. Kintinuous: Spatially extended kinectfusion. In *RSS '12 Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, 2012. 1

[63] Thomas Whelan, Stefan Leutenegger, Renato Salas-Moreno, Ben Glocker, and Andrew Davison. Elasticfusion: Dense slam without a pose graph. In *Robotics: Science and Systems (RSS)*, 2015. 1, 2

[64] Xiuchao Wu, Jiamin Xu, Zihan Zhu, Hujun Bao, Qixing Huang, James Tompkin, and Weiwei Xu. Scalable neural indoor scene rendering. *ACM Transactions on Graphics (TOG)*, 41(4):1–16, 2022. 2

[65] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, pages 641–676. Wiley Online Library, 2022. 1, 2

[66] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8121–8130, 2022. 5

[67] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 499–507. IEEE, 2022. 2, 3, 6, 4

[68] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2492–2502, 2020. 2, 3

[69] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4805–4815, 2021. 3, 4

[70] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T. Lin. iNeRF: Inverting neural radiance fields for pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021. 2

[71] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3, 4, 5, 1

[72] K. Zhang, G. Riegler, N. Snavely, and V. Koltun. NERF++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[73] Shuaifeng Zhi, Michael Bloesch, Stefan Leutenegger, and Andrew J Davison. Scenecode: Monocular dense semantic reconstruction using learned encoded scene representations. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 11776–11785, 2019. 1, 2

[74] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 822–838, 2018. 2

[75] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 12786–12796, 2022. 1, 2, 3, 4, 5, 6

NICER-SLAM: Neural Implicit Scene Encoding for RGB SLAM

Supplementary Material

In this **supplementary document**, we first provide additional implementation details in Sec. 6, then provide additional results on different datasets in Sec. 7. After that, we show the fitted curve of our locally adaptive transformation in Sec. 8. In Sec. 9, we supplement with an ablation study on our design choices. Finally, we provide discussions on the ScanNet and TUM RGB-D datasets in Sec. 10. We also provide a **supplementary video** https://youtu.be/H4cOCa3oUno where we show additional visual comparisons.

6. Implementation Details

Use of COLMAP for intrinsics. In 7-Scenes [49], the RGB and depth camera are not calibrated. Therefore, we employ COLMAP to obtain the intrinsic parameters for each scene. Regarding the SCO dataset, a calibration board was not available at the time of capture. Consequently, we utilized COLMAP to determine the intrinsic parameters. However, it is worth noting that direct calibration could bypass the need for COLMAP to acquire intrinsic parameters.

Frame Selection for Mapping. During mapping, we need to select multiple frames from which we sample rays and pixels. We introduce a simple frame selection strategy. We maintain a global keyframe list, adding a frame every 10 frames. For mapping, we select a total of $K = 16$ frames: 5 randomly selected from the keyframe list, 10 selected from the latest 20 keyframes, and the current frame.

System Details. Mapping happens every 5 frames, while tracking is done every frame. Drifting is a known challenge with pure RGB input. To mitigate this, during the local BA stage in mapping, we jointly optimize the poses for half of the 16 selected frames (those closest to the current frame) with the scene representation, while freezing the rest. In Eq. (9), we set $c_0 = 1.208 \cdot 10^{-2}$, $c_1 = 6.26471 \cdot 10^{-6}$ and $c_2 = 2.3 \cdot 10^{-3}$. For mapping and tracking, we sample $M = 8096$ and $M_t = 1024$ pixels, respectively, and optimize for 100 iterations. During every mapping iteration, pixels are randomly sampled from all selected frames. With our unoptimized PyTorch implementation, each mapping and tracking iteration requires an average of 496ms and 147ms on a single A100 GPU. To optimize the scene representation during the first mapping step, we set the scale and shift as $w = 20$ and $q = 0$ in Eq. (13), aligning the scaled monocular depth with a sensible range (e.g. 1-5 meters).

Initialization for the Scene Representation. Our scene representation is composed of hierarchical feature grids and MLPs. We initialize all levels of feature grids to be zero. As for the coarse MLP for geometry, we adapt the geometric initialization from IGR [16] and MonoSDF [71] to approximate the SDF to a sphere. As for fine geometry MLP and the color MLP, we randomly initialize the network.

DIM-SLAM*. Only a small part of the DIM-SLAM [24] codes (initialization stage) have been released, so we faithfully implement the entire pipeline ourselves, denoted as DIM-SLAM*. We discuss extensively with the authors during our implementation. What we have implemented are the following:

- Keyframe selection and graph management mechanism
- Selection of overlapping frames for bundle adjustments
- Trajectory alignment for tracking evaluation
- Mesh extraction and evaluation
- Mesh culling
- Post optimization module
- Tuning loss weights and learning rates for scene representation and camera poses

Mesh Culling for DIM-SLAM*. DIM-SLAM originally requires rendering out multiple depth images within the scene and fusing them into a TSDF volume. A bounding box is then calculated on the extracted mesh to remove floating undesired geometries outside. In our implementation, the reconstructed mesh is first aligned to the dataset's coordinate system (e.g. Replica) based on the alignment matrix of the predicted camera trajectory. Geometries beyond the ground-truth mesh's bounding box are then removed. It's noteworthy that NICER-SLAM requires no post-processing after extracting the mesh thanks to the use of SDF for representing scene geometry. See Fig. 8 for comparison.

7. Additional Results

Additional Results on Replica Dataset. Additional results on reconstruction and rendering are shown in Fig. 9 and Fig. 10.

Additional Results on 7-Scenes Dataset. In terms of tracking, as can be observed in Table 4, baselines with RGB-D input outperform RGB-only methods overall, emphasizing depth inputs' importance for tracking, especially amidst imperfect RGB images. Among RGB-only methods, COLMAP, DSO, and DROID-SLAM* perform poorly in the `pumpkin` scene because of large textureless and reflective regions. The same behavior is observed in NeRF-SLAM and Orbeez-SLAM since they rely on off-the-shelf SLAM pipelines as tracking frontends. In contrast, NICER-SLAM is more robust to such issues thanks to the predicted

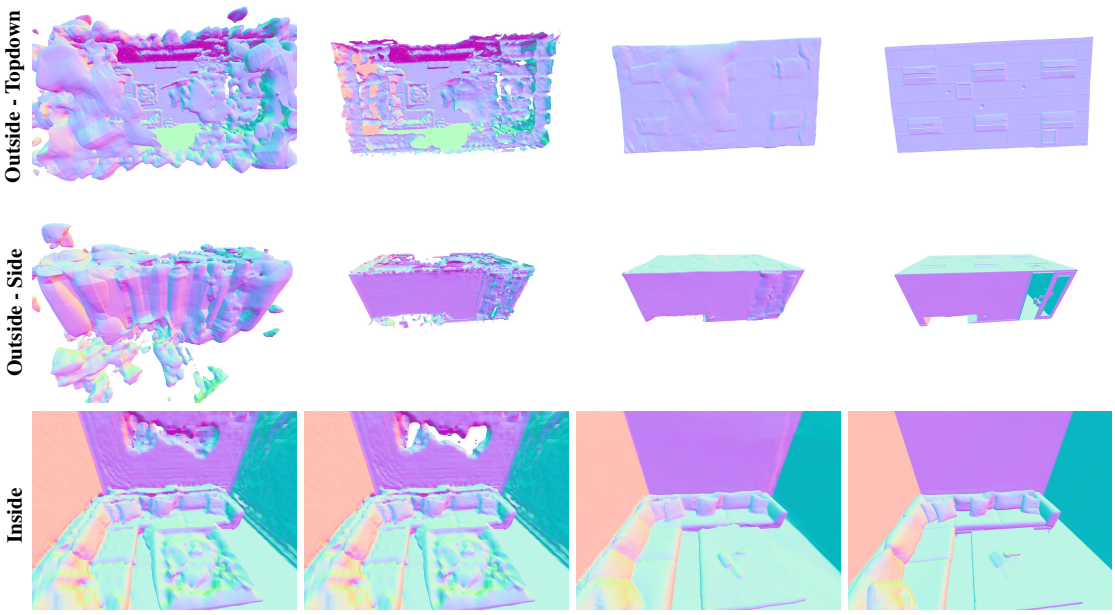

Figure 8. **DIM-SLAM* Mesh Culling.** For DIM-SLAM*, their reconstructed mesh is first aligned to Replica's coordinate system, and those floating geometries outside the bounding box of ground-truth mesh are removed. Note that for NICER-SLAM no post-processing is needed, thanks to our SDF representation.

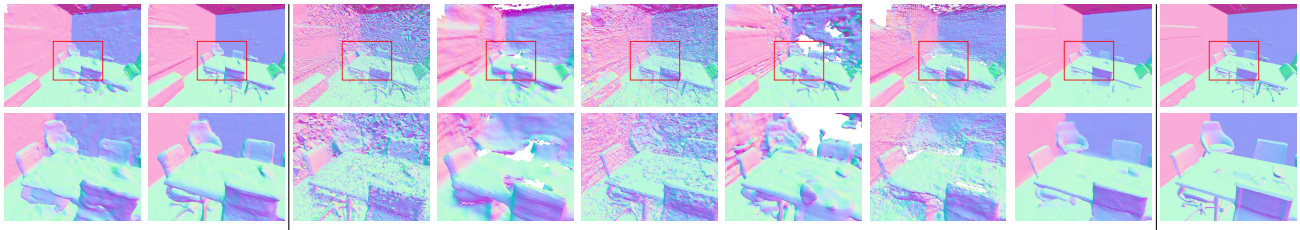

Figure 9. **3D Reconstruction Results on the Replica Dataset [50].** The second row shows zoom-in views for better comparison.

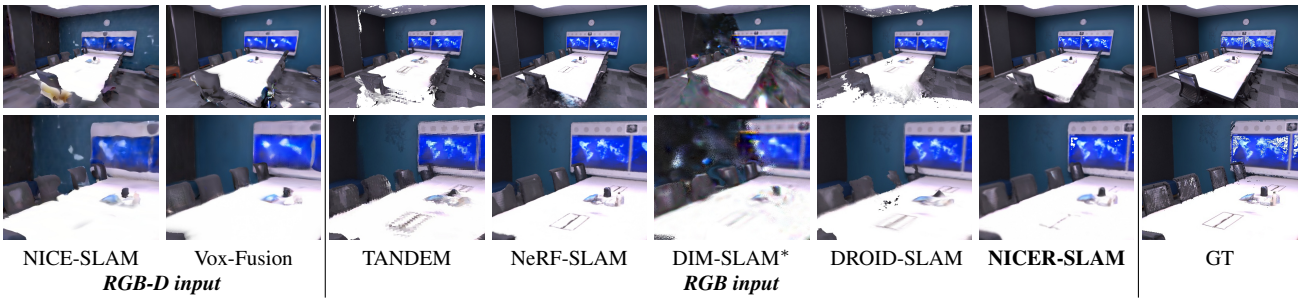

Figure 10. **Novel View Synthesis Results on the Replica Dataset [50].** The second row shows zoom-in renderings for better comparison. Note that we selected novel viewpoints far from the training views (extrapolation).

monocular geometric priors and its dense alignment.

Additional Results on Self-Captured Outdoor Dataset. Additional results on reconstruction and rendering are shown in Fig. 11 and Fig. 12.

8. Fitted Curve for Locally Adaptive Transformation

In the main paper, we propose the locally adaptive transformation from SDF to volume density. This strategy in-

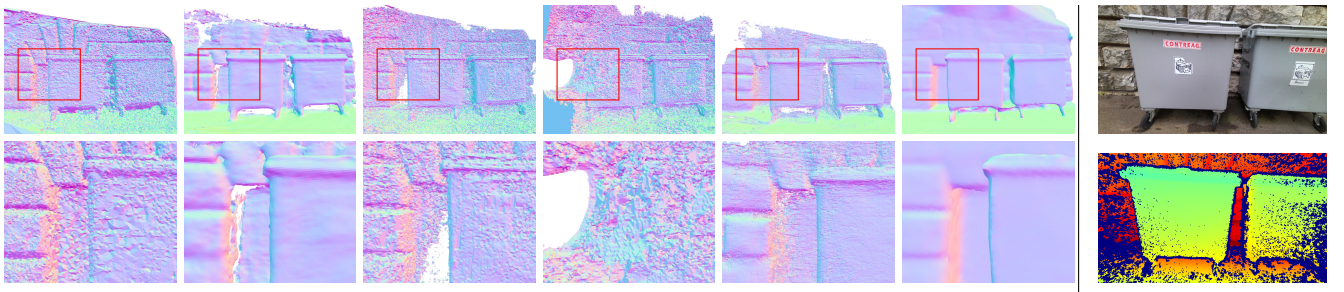

Figure 11. **3D Reconstruction Results on the Self-Captured Outdoor (SCO) Dataset.** The second row shows zoomed-in normal maps for better comparison. Note that only RGB images are used as input, while the depth image is only for visualization, showing the captured depth is unable to provide reliable readings in outdoor environments.

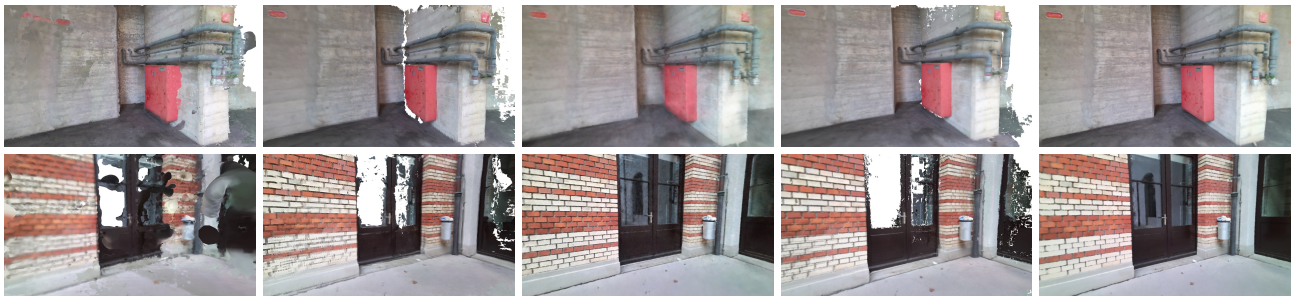

Figure 12. **Novel View Synthesis Results on our Self-captured Outdoor Dataset.** We can obtain visually compelling and more complete results than other SLAM methods, and perform similarly to NeRF-SLAM, which primarily focuses on novel view synthesis and relies on DROID-SLAM for camera tracking.

	chess	fire	heads	office	pumpkin	kitchen	stairs	Avg.
RGB-D input								
NICE-SLAM	**2.16**	**1.63**	7.80	5.73	19.34	3.31	4.31	6.33
Vox-Fusion	2.53	1.91	1.94	**5.26**	**15.33**	**2.79**	3.40	**4.74**
RGB input								
COLMAP	3.42	2.40	1.70	10.42	52.32	5.08	2.62	11.14
TANDEM	42.63	2.59	6.65	9.77	17.20	29.59	13.62	17.44
DSO	18.90	7.53	21.25	11.04	56.57	31.82	15.36	23.21
Orbeez-SLAM	3.36	2.07	**1.27**	10.50	67.78	4.25	**1.69**	12.99
NeRF-SLAM	9.34	8.57	4.44	16.67	43.96	9.02	5.41	13.92
DIM-SLAM*	60.13	2.41	39.42	24.15	13.07	39.11	10.45	26.96
DROID-SLAM	3.36	2.40	1.43	9.19	16.46	4.94	1.85	5.66
DROID-SLAM*	3.55	2.49	3.32	10.93	48.53	4.72	2.58	10.87
NICER-SLAM	3.28	6.85	4.16	10.84	20.00	3.94	10.81	8.55

Table 4. **Camera Tracking Results on the 7-Scenes Dataset.** ATE RMSE [cm] (↓) is used as the evaluation metric.

volves locally assigning the β value within the space. To achieve this, we design a function for translating the local point counter T_p to β. The function takes the form $\beta = c_0 \cdot \exp(-c_1 \cdot T_p) + c_2$, where $c_0 = 1.208 \cdot 10^{-2}$, $c_1 = 6.26471 \cdot 10^{-6}$ and $c_2 = 2.3 \cdot 10^{-3}$. The fitted exponential curve representing this function is depicted in Fig. 13.

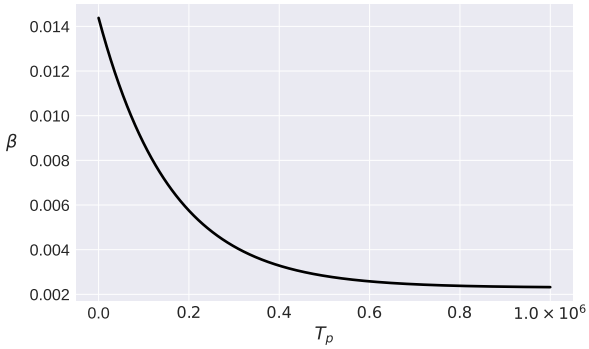

Figure 13. **Fitted Curve for Locally Adaptive Transformation**. The function is $\beta = c_0 \cdot \exp(-c_1 \cdot T_p) + c_2$, where $c_0 = 1.208 \cdot 10^{-2}$, $c_1 = 6.26471 \cdot 10^{-6}$ and $c_2 = 2.3 \cdot 10^{-3}$.

9. Ablation Study

Losses. In Table 5 (a), we assess the impact of different mapping losses from Sec. 3.3 on both reconstruction and tracking, because we conduct local BA on the third stage of mapping. As can be noticed, using all losses together yields the best overall performance. Without monocular depth or normal loss, both mapping and tracking accuracy drops significantly, indicating the crucial role of monocular geometric cues in disambiguating the optimization process.

Ablation on Hierarchical Architecture. In Table 5 (b) we show that removing the multi-resolution color feature grids $\{\Phi_l^{\text{color}}\}_1^L$ and representing scene colors solely on the MLP f^{color} results in a significant performance drop, em-

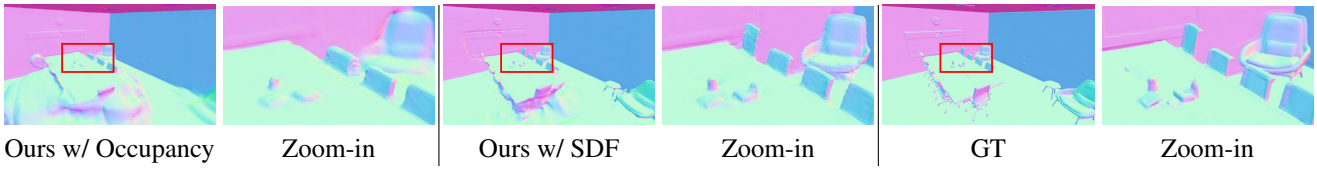

Figure 14. **Ablation Study on SDF vs. Occupancy.** We conduct the ablation on one random Replica scene (`office-4`). The figures on the right depict zoomed-in normal maps for better comparison.

	ATE RMSE↓	Acc. ↓	Comp. ↓	Comp. Ratio ↑	Normal Cons. ↑
w/o $\mathcal{L}_{\text{depth}}$	4.48	4.74	6.18	71.10	89.23
w/o $\mathcal{L}_{\text{normal}}$	3.22	7.25	6.98	51.07	86.64
w/o $\mathcal{L}_{\text{warp}}$	2.96	3.76	4.60	74.42	**91.32**
w/o $\mathcal{L}_{\text{flow}}$	2.30	3.31	4.31	81.10	91.00
Ours	**2.01**	**3.03**	**3.87**	**83.98**	90.96

(a) Ablation Study on Losses in Eq. (16).

	ATE RMSE↓	Acc. ↓	Comp. ↓	Comp. Ratio ↑	Normal Cons. ↑
w/o $\{\Phi_l^{\text{color}}\}_1^L$	9.92	8.32	8.49	50.13	87.84
w/o Φ^{coarse}	3.07	4.51	5.11	67.29	90.17
Ours	**2.01**	**3.03**	**3.87**	**83.98**	**90.96**

(b) Ablation Study on Hierarchical Architecture.

	ATE RMSE↓	Acc. ↓	Comp. ↓	Comp. Ratio ↑	Normal Cons. ↑
Fixed β=0.01	3.81	7.77	8.28	39.48	87.52
Fixed β=0.001	3.98	3.48	5.05	76.67	90.39
Global optimizable β	2.62	3.64	4.53	76.35	90.88
Voxel size of 32^3	3.00	3.19	4.35	81.90	90.65
Voxel size of 128^3	2.16	4.35	4.87	68.96	90.40
Ours	**2.01**	**3.03**	**3.87**	**83.98**	**90.96**

(c) Ablation Study on SDF-to-Density Transformation.

Table 5. **Ablation Study.** On a single random Replica scene (`office-4`), we evaluate both camera tracking and reconstruction.

phasizing the importance of multi-res color feature grids. Similarly, removing the coarse feature grid Φ^{coarse} and only using fine-level feature grids for SDFs also causes inferior performance, especially in the completeness/completeness ratio.

SDF-to-Density Transformation. We also compare different choices for the transformation from SDF to volume density (see Sec. 3.2): (a) Fixed β value, (b) globally optimizable β as in [69], and also (c) different voxel size for counting (our default setting uses 64^3). As can be seen in Table 5 (c), with the locally adaptive transformation and under the chosen voxel size, our method is able to obtain both better scene geometry and camera tracking.

SDF vs. Occupancy. Unlike recent implicit-based SLAM systems [24, 67, 75] that use occupancy for scene geometry, we employ SDFs. To verify this choice, we keep the pipeline identical but replace the output in Eq. (4) to confine the occupancy probability between 0 and 1, and eliminate the Eikonal loss $\mathcal{L}_{\text{eikonal}}$. As evident in Fig. 14 where we compare reconstruction with given GT poses, using SDFs leads to more accurate geometry.

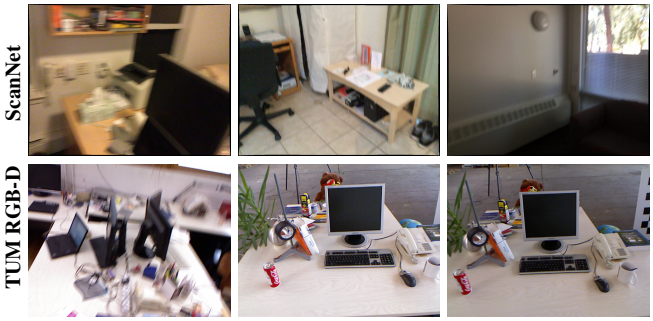

Figure 15. **Sampled Frames from ScanNet and TUM RGB-D Datasets.** Both datasets were captured using outdated cameras. ScanNet exhibits significant motion blur and sudden camera motion changes, while TUM RGB-D is characterized by fluctuating light conditions and uncalibrated white balance.

10. ScanNet & TUM RGB-D Datasets

In this section, we discuss why ScanNet [10] and TUM-RGBD [51] datasets might not be the preferred choices for evaluating monocular SLAM systems.

ScanNet. In particular, ScanNet is notable for its significant motion blur and a high level of noises in the RGB sequences, as clearly evident from the sampled frames in the first row of Fig. 15. The deterioration of image quality compounds the complexity of optimization, especially in the absence of robust geometry guidance provided by depth maps. We tested various RGB SLAM approaches on ScanNet including ORB-SLAM2 [33], DSO [14], and DIM-SLAM [24]. However, they often failed at the initialization stage already.

DROID-SLAM, the SOTA SLAM method for camera tracking, also struggles with ScanNet. As shown in Table 6, it breaks down in the middle of the sequence in 3 out of 6 scenes. Consequently, recent methods like NeRF-SLAM and Orbeez-SLAM, which heavily depend on these SOTA SLAM methods for camera pose estimation, also struggle or fail on ScanNet. As for NICER-SLAM, while camera tracking is not the primary objective, our system manages to deliver reasonable results without a complete breakdown.

TUM RGB-D Dataset. The TUM-RGBD dataset was introduced over a decade ago, using hardware that is now outdated. It shares similar motion blur issues with ScanNet, and presents additional challenges in its RGB sequences,

Scene ID	0000	0059	0106	0169	0181	0207
RGB-D input						
iMAP*	55.95	32.06	17.50	70.51	32.10	11.91
NICE-SLAM	8.64	**12.25**	**8.09**	**10.28**	**12.93**	**5.59**
RGB input						
DROID-SLAM	**5.63**	F	207.67	F	F	7.97
DROID-SLAM*	18.75	F	209.98	F	F	18.28
NICER-SLAM	96.03	63.24	117.05	77.88	67.97	70.76

Table 6. **Camera Tracking Results on ScanNet [10].** ATE RMSE (↓) is used as the evaluation metric. F denotes program failure or final trajectory unable to align with GT (SVD decomposition error). DROID-SLAM* is the DROID-SLAM without global BA and loop closure.

	fr1/desk	fr2/xyz	fr3/office
RGB-D input			
BAD-SLAM	1.7	1.1	1.7
Kintinuous	3.7	2.9	3.0
iMAP	4.9	2.0	5.8
iMAP*	7.2	2.1	9.0
DI-Fusion	4.4	2.3	15.6
NICE-SLAM	2.7	1.8	3.0
RGB input			
ORB-SLAM2	**1.6**	**0.4**	**1.0**
DROID-SLAM	1.8	0.5	2.8
NICER-SLAM	7.7	11.9	8.7

Table 7. **Camera Tracking Results on TUM RGB-D [51].** ATE RMSE (↓) is used as the evaluation metric.

such as fluctuating light conditions and uncalibrated white balance, as shown in the second row of Fig. 15.

As presented in Table 7, our method does not perform as well as baseline approaches on this dataset. This may be attributed to this kind of rapid fluctuations in light conditions, which is especially challenging for NeRF-based optimization. Specifically, NICER-SLAM optimizes camera poses and scene in an end-to-end manner, by directly back-propagating the loss between volume-rendered colors and input RGB colors. However, the varying lighting conditions introduce a significant challenge to the joint learning of 3D-consistent geometry and colors, as well as camera poses.

References

[1] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. *arXiv preprint arXiv:2212.07388*, 2022. 2

[2] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J. Davison. Codeslam - learning a compact, optimisable representation for dense visual SLAM. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2560–2568, Salt Lake City, UT, USA, 2018. Computer Vision Foundation / IEEE Computer Society. 1, 2

[3] Erik Bylow, Jürgen Sturm, Christian Kerl, Fredrik Kahl, and Daniel Cremers. Real-time camera tracking and 3d reconstruction using signed distance functions. In *Robotics: Science and Systems (RSS)*, page 2, 2013. 2

[4] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5939–5948, 2019. 2

[5] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6970–6981, 2020. 3

[6] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 264–280. Springer, 2022. 2

[7] Chi-Ming Chung, Yang-Che Tseng, Ya-Ching Hsu, Xiang-Qian Shi, Yun-Hung Hua, Jia-Fong Yeh, Wen-Chin Chen, Yi-Ting Chen, and Winston H Hsu. Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9400–9406. IEEE, 2023. 2, 6

[8] Ronald Clark. Volumetric bundle adjustment for online photorealistic scene capture. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6124–6132, 2022. 2

[9] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J Davison. Deepfactors: Real-time probabilistic dense monocular slam. *IEEE Robotics and Automation Letters*, 5 (2):721–728, 2020. 1, 2

[10] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4, 5

[11] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Trans. on Graphics*, 36(4):1, 2017. 2

[12] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6260–6269, 2022. 5

[13] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 10786–10796, 2021. 5

[14] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 40(3):611–625, 2017. 2, 6, 4

[15] Daniel Girardeau-Montaut. Cloudcompare. *France: EDF R&D Telecom ParisTech*, 11, 2016. 6

[16] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 5, 1

[17] Michael Grupp. evo: Python package for the evaluation of odometry and slam. https://github.com/MichaelGrupp/evo, 2017. 6

[18] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5511–5520, 2022. 2

[19] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6001–6010, 2020. 2

[20] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. *arXiv preprint arXiv:2211.11704*, 2022. 2, 3

[21] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 225–234. IEEE, 2007. 2

[22] Lukas Koestler, Nan Yang, Niclas Zeller, and Daniel Cremers. Tandem: Tracking and dense mapping in real-time using deep multi-view stereo. In *Proc. Conf. on Robot Learning (CoRL)*, pages 34–45. PMLR, 2022. 2, 6

[23] Evgenii Kruzhkov, Alena Savinykh, Pavel Karpyshev, Mikhail Kurenkov, Evgeny Yudin, Andrei Potapov, and Dzmitry Tsetserukou. Meslam: Memory efficient slam based on neural fields. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 430–435. IEEE, 2022. 2, 3

[24] Heng Li, Xiaodong Gu, Weihao Yuan, Luwei Yang, Zilong Dong, and Ping Tan. Dense rgb slam with neural implicit maps. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2023. 3, 6, 1, 4

[25] C. Lin, W. Ma, A. Torralba, and S. Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. 2

[26] Stefan Lionar, Daniil Emtsev, Dusan Svilarkovic, and Songyou Peng. Dynamic plane convolutional occupancy networks. In *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1829–1838, 2021. 2

[27] Daniil Lisus and Connor Holmes. Towards open world nerf-based slam. *arXiv preprint arXiv:2301.03102*, 2023. 2, 3

[28] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2019–2028, 2020. 2

[29] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4460–4470, 2019. 2

[30] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 2, 3, 4

[31] Yuhang Ming, Weicai Ye, and Andrew Calway. idf-slam: End-to-end rgb-d slam with neural implicit mapping and deep feature tracking. *arXiv preprint arXiv:2209.07919*, 2022. 2, 3

[32] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. on Graphics*, 41(4), 2022. 2, 3, 4, 8

[33] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017. 2, 4

[34] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 2

[35] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011. 1, 2

[36] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 2320–2327. IEEE, 2011. 1, 2

[37] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[38] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Trans. on Graphics*, 32(6):1–11, 2013. 2

[39] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 5589–5599, 2021. 4

[40] Joseph Ortiz, Alexander Clegg, Jing Dong, Edgar Sucar, David Novotny, Michael Zollhoefer, and Mustafa Mukadam. isdf: Real-time neural signed distance fields for robot perception. In *Robotics: Science and Systems (RSS)*, 2022. 2, 3

[41] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 2

[42] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 523–540. Springer, 2020. 2

[43] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:13032–13044, 2021. 2

[44] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2020. 5

[45] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 14335–14345, 2021. 2

[46] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. *arXiv preprint arXiv:2210.13641*, 2022. 2, 6

[47] J. L. Schönberger and J. M. Frahm. Structure-from-motion revisited. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 6

[48] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. BAD SLAM: bundle adjusted direct RGB-D SLAM. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 134–144, 2019. 1, 2

[49] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013. 5, 7, 8, 1

[50] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. R., S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 1, 5, 6, 7, 2

[51] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2012. 6, 4, 5

[52] Edgar Sucar, Kentaro Wada, and Andrew Davison. Nodeslam: Neural object descriptors for multi-view shape reconstruction. In *Proc. of the International Conf. on 3D Vision (3DV)*, pages 949–958. IEEE, 2020. 2

[53] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 6229–6238, 2021. 1, 2, 4

[54] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 11358–11367, 2021. 3

[55] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

[56] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2019. 2

[57] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.

[58] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *Advances in Neural Information Processing Systems*, pages 16558–16569, 2021. 1, 2, 6

[59] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5038–5047, 2017. 2

[60] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 4

[61] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2

[62] Thomas Whelan, Michael Kaess, Maurice Fallon, Hordur Johannsson, John Leonard, and John McDonald. Kintinuous: Spatially extended kinectfusion. In *RSS '12 Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, 2012. 1

[63] Thomas Whelan, Stefan Leutenegger, Renato Salas-Moreno, Ben Glocker, and Andrew Davison. Elasticfusion: Dense slam without a pose graph. In *Robotics: Science and Systems (RSS)*, 2015. 1, 2

[64] Xiuchao Wu, Jiamin Xu, Zihan Zhu, Hujun Bao, Qixing Huang, James Tompkin, and Weiwei Xu. Scalable neural indoor scene rendering. *ACM Transactions on Graphics (TOG)*, 41(4):1–16, 2022. 2

[65] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, pages 641–676. Wiley Online Library, 2022. 1, 2

[66] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8121–8130, 2022. 5

[67] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 499–507. IEEE, 2022. 2, 3, 6, 4

[68] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2492–2502, 2020. 2, 3

[69] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4805–4815, 2021. 3, 4

[70] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T. Lin. iNeRF: Inverting neural radiance fields for pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021. 2

[71] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3, 4, 5, 1

[72] K. Zhang, G. Riegler, N. Snavely, and V. Koltun. NERF++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[73] Shuaifeng Zhi, Michael Bloesch, Stefan Leutenegger, and Andrew J Davison. Scenecode: Monocular dense semantic reconstruction using learned encoded scene representations. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 11776–11785, 2019. 1, 2

[74] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 822–838, 2018. 2

[75] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 12786–12796, 2022. 1, 2, 3, 4, 5, 6