Multi-Modal Neural Radiance Field for Monocular Dense SLAM with a Light-Weight ToF Sensor

Xinyang Liu[1], Yijin Li[1], Yanbin Teng[1], Hujun Bao[1], Guofeng Zhang[1], Yinda Zhang[2], Zhaopeng Cui[1]*
[1]State Key Lab of CAD&CG, Zhejiang University [2]Google

Abstract

Light-weight time-of-flight (ToF) depth sensors are compact and cost-efficient, and thus widely used on mobile devices for tasks such as autofocus and obstacle detection. However, due to the sparse and noisy depth measurements, these sensors have rarely been considered for dense geometry reconstruction. In this work, we present the first dense SLAM system with a monocular camera and a light-weight ToF sensor. Specifically, we propose a multi-modal implicit scene representation that supports rendering both the signals from the RGB camera and light-weight ToF sensor which drives the optimization by comparing with the raw sensor inputs. Moreover, in order to guarantee successful pose tracking and reconstruction, we exploit a predicted depth as an intermediate supervision and develop a coarse-to-fine optimization strategy for efficient learning of the implicit representation. At last, the temporal information is explicitly exploited to deal with the noisy signals from light-weight ToF sensors to improve the accuracy and robustness of the system. Experiments demonstrate that our system well exploits the signals of light-weight ToF sensors and achieves competitive results both on camera tracking and dense scene reconstruction. Project page: https://zju3dv.github.io/tof_slam/.

1. Introduction

Dense simultaneous localization and mapping (dense SLAM) [38, 8, 9, 44] has extensive applications in augmented reality [13, 17], indoor robotics, etc. It usually relies on high-precision and high-resolution depth sensors, such as time-of-flight (ToF) sensors or structured light sensors. Due to the size, weight, and price issues, these depth sensors are only used in a few high-end mobile devices until recent years. In contrast, light-weight ToF sensors, which are cost-effective, compact, and energy-efficient, were integrated into hundreds of smartphone models[1]. As a result, it would be valuable if we could fully utilize these light-weight sensors for dense SLAM, which further facilitates other applications like AR/VR and micro-Robot.

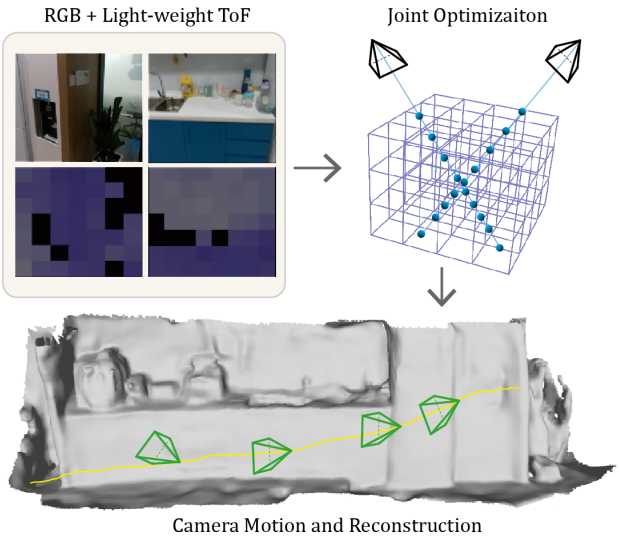

Figure 1. **Monocular Dense SLAM with Our Multi-Modal Implicit Representation.** We present a novel SLAM system based on implicit scene representation. The system does not require high-precision and high-resolution depth sensors and only takes RGB images and the signals of light-weight ToF sensors as input.

Unfortunately, limited by the compact electronic design, the light-weight ToF sensor can only provide coarse measurement in the form of depth distribution in an extremely low resolution as illustrated in Fig. 2. Existing RGB-D dense SLAM systems [38, 46, 31] are designed for accurate and pixel-wise depth inputs, thus cannot work with the light-weight ToF signals directly. They will also fail if we simply consider the light-weight ToF signals as a low-resolution depth (*i.e.*, mean depth values in each zone).

In this paper, we aim to design a novel learning-based dense SLAM system that provides accurate pose tracking and dense reconstruction taking the RGB sequences from a color camera and the sparse signals of light-weight ToF as input (Fig. 1). However, it is non-trivial to design such

*Corresponding author.
[1]https://www.st.com/content/st_com/en/about/media-center/press-item.html/t4210.html

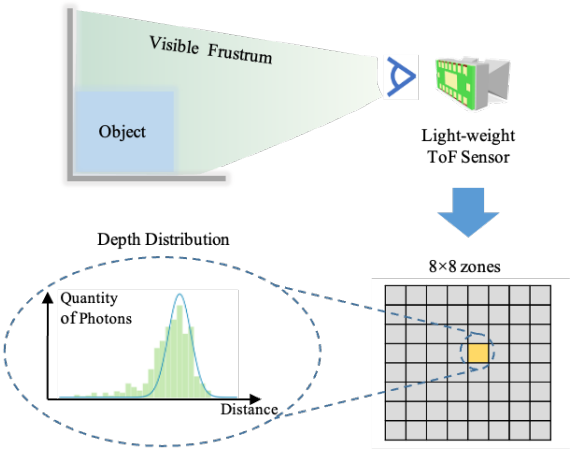

Figure 2. **L5 Sensing Principle.** Instead of pixel-wise depth measurement, L5 measures the depth distribution of a large zone and returns its depth mean and variance. Specifically, L5 measures 8×8 zones in total. The figure is adapted from [14].

a system. At first, motivated by the recent achievements in the field of neural rendering [20, 2, 40, 39, 41] and grid-based feature encoding [46, 36, 18], we propose to design our system based on a novel implicit representation from which we can render both the RGB image and original ToF signal. In this way, we can define the losses directly against the multi-domain input, which can be utilized to optimize camera poses and 3D scenes. However, we find that this cannot guarantee plausible tracking and reconstruction results because of low-quality raw depth signals from the sensor. Inspired by recent works [31, 46] that high-resolution accurate depth maps play an important role in the implicit SLAM systems, we further exploit the depth estimation model [14] for the light-weight ToF sensor to predict an intermediate high-resolution depth as additional supervision. Based on all the above insights, we develop a multi-modal implicit scene representation with multi-level feature grids, which is able to generate both the zone-level signals of light-weight ToF sensors and pixel-wise RGB/depth images via the differentiable rendering for the camera tracking and reconstruction. For efficient network convergence, we also design a coarse-to-fine optimization process for this novel implicit scene representation, *i.e.*, firstly using zone-level ToF signals to optimize the scene at the coarse level, then adding pixel-wise RGB/depth supervisions to recover geometry details.

Moreover, although the predicted per-pixel depth is generally smooth and provides reasonable supervision as intermediate signals, it may also produce severe artifacts when there is a large portion of missing L5 signals since it is hard for the network to handle such cases due to the inherent depth ambiguity in the missing regions. As a result, we further develop a temporal filtering technique to enhance depth prediction. Specifically, when a new signal is captured, we render a zone-level light-weight ToF signal from our multi-modal scene representation with an initialized pose and fuse it with that new observation signal, which serves as the input of the depth prediction network. Such an explicit filtering technique improves the depth estimation performance significantly, particularly in extreme cases where the raw L5 signals are very noisy or contain large amounts of missing data, and therefore further benefits the whole SLAM system.

Our contributions can be summarized as follows. At first, to our best knowledge, we present the first dense SLAM system by only taking the monocular images and the signals from a light-weight ToF sensor as input. Moreover, we propose a multi-modal implicit scene representation which supports rendering both the zone-level signals of light-weight ToF sensors and pixel-wise RGB/depth images. By minimizing the re-rendering loss of these signals in a coarse-to-fine strategy, we can recover the camera pose and the scene geometry via differentiable neural rendering. Furthermore, we propose a temporal filtering technique to enhance the signals of light-weight ToF sensors and corresponding depth prediction which significantly improve the proposed SLAM system in extreme cases. Experiments on the real datasets demonstrate that the proposed system well exploits the signals of light-weight ToF sensors and achieves competitive results both on camera tracking and dense scene reconstruction compared to existing methods.

2. Related Work

Visual SLAM. Sparse visual SLAM systems [13, 23, 16] focus on solving accurate camera poses based on the tracking of sparse keypoint [28, 25]. These types of methods usually struggle in texture-less environments and cannot provide a complete reconstruction result of the scene. In comparison, dense visual SLAM systems [10, 38, 11, 24] perform much more robust but usually require depth images as input. More recently, many methods emerged to estimate the dense depth map and camera pose simultaneously from RGB sequences only using deep neural networks [3, 6, 34, 35] and reconstruct the scene through fusing the estimated depth maps. Unlike these previous methods optimizing depth maps per frame, we follow iMAP [31] and NICE-SLAM [46] and use an implicit scene representation. However, unlike iMAP and NICE-SLAM, we do not require the depth camera and take the light-weight ToF signals as input instead.

Neural Implicit Representation. Neural implicit representations are widely used in various kinds of tasks, including novel view synthesis [20, 45, 2, 19] and scene reconstruction [1, 36, 26]. While showing promising results, these methods require precise camera poses as input which greatly limits their application scenarios. Some re-

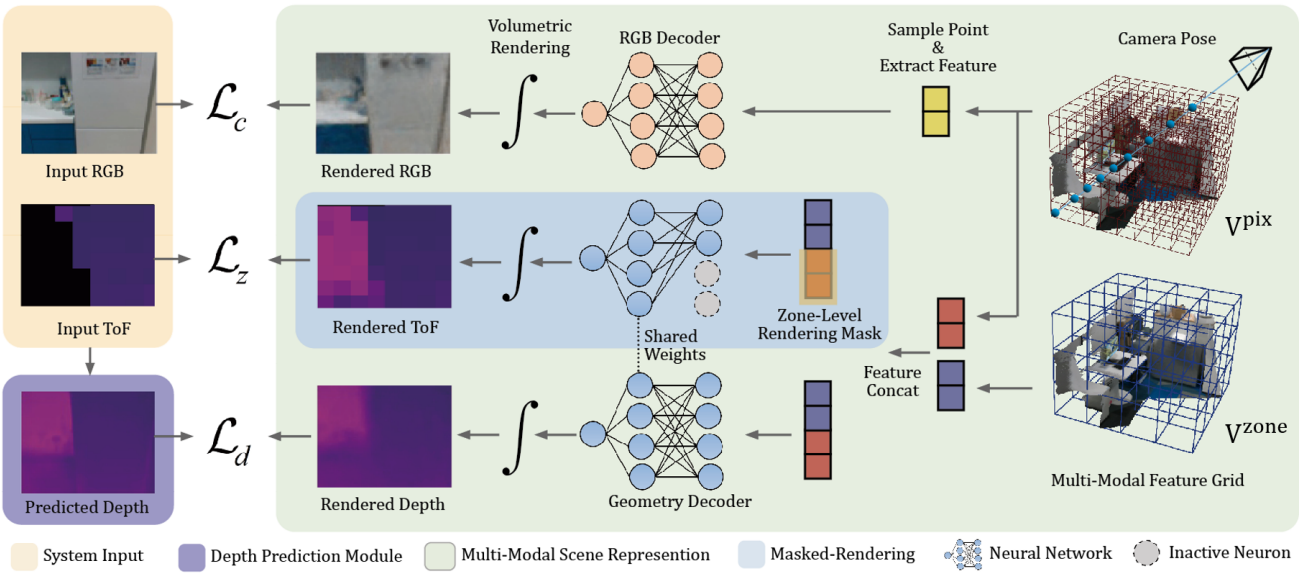

Figure 3. **System Overview.** Our method uses a monocular camera and a light-weight ToF sensor as input and recovers both camera motion and the scene structures. Through differentiable rendering, our method can render multi-modal signals, including color images, depth images and zone-level L5 signals. Both the scene structure and the camera poses are optimized by minimizing the re-rendering loss.

cent works [15, 42] propose to optimize camera pose with the scene representation simultaneously. Lately, iMAP [31] and NICE-SLAM [46] bring neural implicit representations into visual SLAM systems. Nevertheless, requirements of RGB-D sequences as input significantly constrain their applications, especially on mobile devices with tight power budgets. In contrast, we propose a visual SLAM method using only an RGB camera and a light-weight ToF sensor, which commonly co-exists on mobile devices.

Neural Grid Optimization. Typically, a single multi-layer perceptron (MLP) is employed to represent the entire scene in earlier neural implicit methods [20, 2, 31]. However, these methods perform poorly when applied to large scenes and need an extremely long time for training. To address this issue, recent works [33, 22, 5] try to optimize neural features on an explicit volumetric grid. These methods, however, are only capable of pixel-level rendering and do not accommodate L5 signals as it has extremely low resolution and measures the depth distribution of a large area. To this end, we introduce a multi-modal feature grid representation that enables rendering multi-modal information at different resolutions and can deal with zone-level depth distribution data measured by light-weight ToF sensors.

3. Method

In this section, we provide an overview of the sensing principle of light-weight ToF sensors (*e.g.*, L5) in Sec. 3.1. Based on the characteristics of L5's signals, we design the first dense SLAM system with a monocular camera and a light-weight ToF sensor as shown in Fig. 3. Specifically, we first propose a multi-modal implicit scene representation which enables rendering L5 signals together with the common RGB image and depth maps (Sec. 3.2). To guarantee successful tracking and mapping, we exploit a depth prediction model [14] to predict intermediate per-pixel depth maps as additional supervision. By minimizing the difference between the rendered signals and input/predicted ones, we can simultaneously optimize the camera pose and scene structure in a coarse-to-fine way. Furthermore, since the depth prediction may contain severe artifacts when there is a large portion of missing L5 signals, we propose to refine the L5 signals with temporal filtering techniques to enhance the depth prediction module (Sec. 3.3). Finally, we describe the system implementation detail in Sec. 3.4.

3.1. Preliminaries: L5 Sensing Principle

Light-weight ToF sensors are designed to be low-cost, small, and low-energy and have been massively deployed on mobile devices. Compared with conventional ToF sensors, which provide high-resolution depth measurement and measure the per-pixel distance to the scene, light-weight ToF sensors usually have extremely low resolution (e.g., 8×8 zones) and measure the depth distribution for each zone. Here we take ST VL53L5CX [29] (denoted as L5), a representative product of light-weight ToF sensors, as an example to declare the sensing principle of these sensors. As shown in Fig. 2, L5 measures the depth distribution by counting the received photon number within specific time

intervals. The result is then fitted with a Gaussian distribution, and L5 only transmits the mean and variance to decrease both the energy consumption and broadband load. None of the previous studies have explored the usage of L5 for downstream applications like SLAM due to its low resolution and high uncertainty.

3.2. Multi-Modal Implicit Scene Representation

The combination of neural rendering and grid-based feature encoding has found broad utilization in applications like SLAM and surface reconstruction. [46, 36, 43]. By minimizing the loss between the rendered image/depth and the input image/depth, they achieve accurate 6-DoF camera pose tracking and the reconstruction of scene geometry. In this paper, we propose a multi-modal implicit scene representation, which supports the rendering of zone-level L5 signals apart from the common RGB and depth images. Specifically, we encode the geometry and color separately and propose the masked rendering technique in the geometry encoding for the rendering of both zone-level L5 signals and pixel-level depth images.

Geometry Encoding with Masked Rendering. The idea of the proposed masked rendering is inspired by the integrated positional encoding(IPE) theory proposed in Mip-NeRF [2], but we promote it to the grid-based scene representation. The core idea of IPE is that the input features are passed through a low pass filter, *i.e.*, if the frequency of a particular feature has a larger period than the ray, then the feature is unaffected; otherwise, the feature is scaled down towards zero. The original method represents the scene using a single MLP and achieves the low pass filtering by calculating an integral of the positional encoding as input. In our grid-based case, we concatenate features from different-level feature grids and use a rendering mask to mask out features extracted from overly high spatial frequency grids based on the current rendering scale.

To be more specific, we encode the scene geometry into a multi-level feature grid containing four layers $\mathcal{V}_\theta = \{V^0, V^1, V^2, V^3\}$. The zone-level feature grid V^{zone} contains the coarser feature grids$\{V^0, V^1\}$ while the pixel-level feature grid V^{pix} contains the finer feature grids$\{V^2, V^3\}$. For a given point $\mathbf{x} \in \mathbb{R}^3$, its whole geometry feature $\mathcal{F}$ is extracted by tri-linearly interpolating features at each grid level and concatenating these features together. The feature can be decoded into SDF values in both pixel-level $\phi_{pix}(\mathbf{x})$ and zone-level $\phi_{zone}(\mathbf{x})$ via the same geometry decoder $f_\omega(\cdot)$ switched by the rendering mask. We use all the features for rendering pixel-level results ϕ_{pix}, and mask out the features extracted from the finer grids for zone-level rendering results ϕ_{zone}:

$$\begin{aligned} \phi_{pix}(\mathbf{x}) &= f_\omega\left([V^{zone}(\mathbf{x}), V^{pix}(\mathbf{x})]\right), \\ \phi_{zone}(\mathbf{x}) &= f_\omega\left([V^{zone}(\mathbf{x}), \mathbf{0}]\right). \end{aligned} \tag{1}$$

This mask operation also makes the corresponding neurons in the geometry decoder inactive. The SDF values from both branches are used in the tracking and mapping process corresponding to the pixel-level and zone-level supervision, and only the pixel-level SDF values are used for the final mesh extraction.

Color Encoding. For color information, we encode it only at the finest level using a separate set of feature grid $\mathcal{W}_\beta$ and decoder $g_\gamma(\cdot)$ as [46, 36]. When decoding color, we additionally use the ray direction $\mathbf{r}$, so the color value for a 3D point is given by:

$$\mathbf{c} = g_\gamma\left(\mathcal{W}_\beta(\mathbf{x}), \mathbf{r}\right). \tag{2}$$

Rendering of L5 Signals, Color and Depth Images. We render color and depth values with the volumetric rendering technique [37]. Specifically, to render a color pixel, we sample N points along the corresponding emitted ray, denoted as $\mathbf{x}_i = \mathbf{o} + d_i\mathbf{r}, i \in \{1, 2, ..., \mathrm{N}\}$ where $\mathbf{o}$ is the camera center, $\mathbf{r}$ is the direction of this ray and d_i is the distance of the sampled point $\mathbf{x_i}$ along the ray. We then accumulate the color value along the ray through:

$$\hat{\mathbf{c}} = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}_i, \tag{3}$$

where $T_i = \prod_{j=1}^{i-1}(1 - \alpha_j)$ is the accumulated transmittance, and α_i is the opacity value converted from the SDF prediction ϕ_{pix}. The conversion follows the original definition in NeuS [37].

The process of rendering the L5 signals is similar to rendering the color. The differences are that for rendering the mean depth value $\hat{d}_z$ of an L5 zone, we emit the ray from the center of that zone and accumulate the distance along the ray through:

$$\hat{d}_z = \sum_{i=1}^{N} T_i \alpha_i d_i. \tag{4}$$

The zone-level opacity α_i is derived in the same manner as color rendering but using ϕ_{zone} instead of ϕ_{pix}. Intuitively, we can optimize the camera pose and scene structure by only supervising the rendered color images and L5 signals. However, in our experiment (Sec. 4.3), we show that the results are far from satisfactory. As a result, we also render the pixel-wise depth maps using Eq. 4 with ϕ_{pix} and supervise it with the depth prediction from [14].

3.3. Temporal Filtering of L5 Signals

As mentioned before, we use DELTAR [14] to predict a pixel-wise depth map as additional supervision. DELTAR is a pre-trained neural network that takes L5 signals and RGB

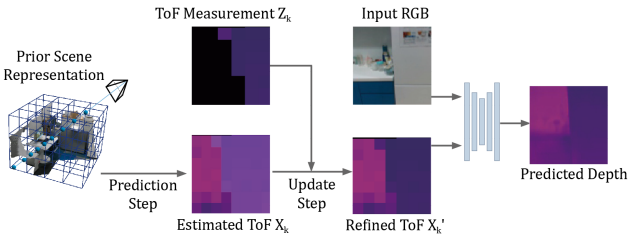

Figure 4. **Temporal Filtering of L5 Signals.** By fusing the latest ToF measurement with a rendered one, we can significantly improve the signal quality and the corresponding predicted depth.

images as input and predicts corresponding depth maps. We observe that when there are a large number of missing or noisy L5 signals, the depth map predicted by DELTAR may contain severe artifacts due to the inherent depth ambiguities of missing or noisy regions, thus further contaminating the learning of implicit features and degrading the SLAM system's performance.

This motivates us to develop an explicit temporal filtering technique (Fig. 4) to enhance the L5 signals before feeding into DELTAR. Since the past observations are stored implicitly in our scene representation, we can leverage this information to refine the current observation. Specifically, the proposed filtering algorithm contains two steps: the prediction step and the update step. In the prediction step, we predict the per-zone ToF measurement $X_k = \{\mu_1, \sigma_1\}$ in the timestamp k using neural rendering (Eq. 4) with an initialized pose. Then we update X_k to X_k' with the current L5 measurement $Z_k = \{\mu_2, \sigma_2\}$:

$$\mu_{X_k'} = \frac{\mu_1\sigma_2^2 + \mu_2\sigma_1^2}{\sigma_1^2 + \sigma_2^2}, \quad \sigma^2_{X_k'} = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}. \tag{5}$$

As for the zones with no valid raw L5 signals, we simply use the predicted L5 signals. Finally, the enhanced signals X_k' are fed into the depth prediction network with corresponding RGB image [14] to obtain a high-resolution depth estimation.

3.4. SLAM System Implementation

Initialization. Since we do not have reliable depth maps as input, we need to perform initialization in order to build a local map for our system to bootstrap. The first N_i frames are added to the initialization process with a fixed interval I_{skip} sequentially. The pose of each newly added frame is initialized by the previous frame. During this process, the feature grids, decoders and camera poses are jointly optimized using the same loss function as in the mapping process. Finally, when all the frames are added, we optimize the whole frame set for N_e iterations.

Coarse-to-Fine Mapping. To optimize the scene representation mentioned in Sec. 3.2, we uniformly sample total M pixels and Z zones and perform optimization to minimize the photometric loss, geometric loss and SDF regularization. The zones and pixels are sampled from the current optimization window, which consists of two types of frames: neighbor frames (the nearest N_n frames with an interval of 4) and global frames (chosen from all the past co-visible mapping frames). The photometric loss is the average color difference of all the sampled pixels:

$$\mathcal{L}_c = \frac{1}{M}\sum_{m=1}^{M} |I[u_m, v_m] - \hat{\mathbf{c}_m}|, \tag{6}$$

where u_m, v_m represents for the image coordinate of pixel m. The geometric loss is calculated on both pixel-level and zone-level. For zone-level loss, only zones with valid L5 signals Z_d are considered. We also add SDF supervision on pixel-level as in [36, 26] to add robustness to the optimization. The whole geometric loss is defined as:

$$\begin{aligned} \mathcal{L}_g &= \mathcal{L}_d + \mathcal{L}_z + \lambda_{sdf}\mathcal{L}_{sdf}, \\ \mathcal{L}_d &= \frac{1}{M}\sum_{m=1}^{M} |\widetilde{D}[u_m, v_m] - \hat{d}_m|, \\ \mathcal{L}_z &= \frac{1}{|Z_d|}\sum_{z\in Z_d} |\bar{D}_z - \hat{d}_z|, \end{aligned} \tag{7}$$

where $\overline{D} \in \mathbb{R}^{8\times 8}$ represents the mean depth value measured by L5 and $\widetilde{D}$ represents the pixel-wise depth prediction. Since we do not have high-quality depth as input and there are many texture-less areas in typical indoor scenes, directly optimizing the feature volume and camera poses is an ill-posed problem. Therefore, we employ two regularizations on the SDF values during pixel-level rendering as [36], namely the eikonal term $\mathcal{L}_{eik}$ and the smoothness term $\mathcal{L}_s$:

$$\mathcal{L}_r = \lambda_{eik}\mathcal{L}_{eik} + \lambda_s\mathcal{L}_s, \tag{8}$$

where the eikonal regularization is employed to ensure the network produces valid SDF values [37, 36], the smoothness regularization is used to encourage neighbor points to have similar normal directions, and λ_{eik} and λ_s are weights to balance the two regularization terms.

During the mapping process, we perform a coarse-to-fine optimization process, bringing better convergence since pixel-wise optimization can rely on the already initialized coarse grids. We first optimize the scene at the coarse level using zone-level ToF signals. Then, pixel-wise RGB/depth supervisions are added to jointly optimize the decoders, feature grids and camera poses by minimizing the mentioned losses for N_m iterations with a local BA to cover geometry details.

$$\min_{\theta,\omega,\beta,\gamma,\{\mathbf{R}_i,\mathbf{t}_i\}} \lambda_c\mathcal{L}_c + \lambda_g\mathcal{L}_g + \lambda_r\mathcal{L}_r. \tag{9}$$

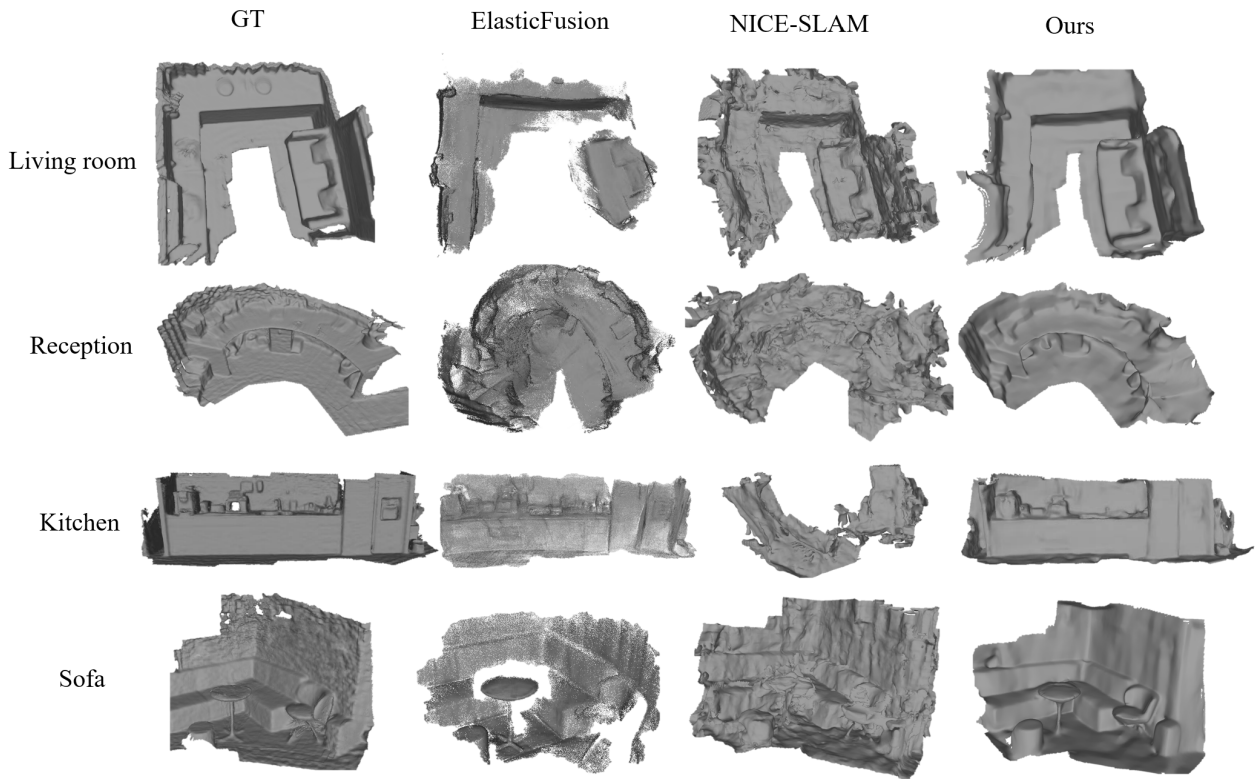

Figure 5. **Qualitative Reconstruction Results.** Compared to other methods, our approach produces high-quality reconstructions with smooth surfaces and fewer artifacts.

We also conduct a global BA every N_g frame by adding all the past mapping frames to the optimization window and jointly optimize them to globally optimize the camera poses and scene geometry. We provide the details of losses in the supplementary material.

Tracking. During tracking, the feature grid and the decoders remain fixed, while only the 6-DoF pose of the current frame is optimized. Similar to previous methods [46, 31], M_t pixels are sampled in the current frame to calculate both the pixel-level photometric loss and geometric loss as previously described. The camera pose is then updated iteratively via back-propagation. Tracking with only pixel-level supervision is susceptible to a suboptimal solution, so we leverage our multi-modal scene representation and apply zone-level supervision in the early stage. Specifically, we additionally sample Z zones in the current frame and use the L5 signal to minimize zone depth loss as well, which leads to a more robust tracking result.

4. Experiments

4.1. Experimental Setup

Datasets. Since no public SLAM benchmarks take the L5 signals as input, we build a data capture device as in [14] and use it to create an indoor SLAM dataset, which contains 7 sequences captured in 5 typical indoor scenes. We record color images (640×480), L5 signals and depth maps for each sequence. Note that we only use the color images and L5 signals as the input to our SLAM system. The depth maps are used to obtain the ground truth 6-DoF camera poses and 3D surface mesh. We follow the automated capture pipeline proposed in ScanNet [7] to compute the ground truth data.

Baselines. We compare our method with two categories of baselines: (a) learning-based SLAM methods including iMAP [31] (re-implemented by [46]) and NICE-SLAM [46]; (b) traditional SLAM methods including ORB-SLAM3 [4], KinectFusion [10], ElasticFusion [38] and BundleFusion [8]. For the methods that only support RGB-D input, including iMAP, NICE-SLAM, KinectFusion, ElasticFusion and BundleFusion, we use the RGB image and the predicted depth map by [14] as the system input. For ORB-SLAM3, we evaluate it in both the RGB version (given only RGB image as input) and the RGB-D version (using the predicted depth, marked as $\widetilde{D}$). Note that none of these methods can work well with only raw L5 zone-level depths or take it as additional inputs. We evaluate both the scene reconstruction and the camera tracking results. Since

Scene Name		Kitchen	Sofa	Office	Reception	Living room	Office2	Sofa2	Avg.
KinectFusion[10]	Acc.↓	-	0.190	0.211	0.261	-	0.267	0.135	0.213
	Comp.↓	-	0.048	0.046	0.064	-	**0.078**	0.064	**0.060**
	F-score	-	0.278	0.288	0.285	-	0.274	0.381	0.301
ElasticFusion[38]	Acc.↓	0.092	0.135	0.084	0.297	0.151	**0.096**	0.122	0.140
	Comp.↓	**0.065**	0.048	0.082	0.305	0.216	0.147	0.047	0.130
	F-score	0.553	0.420	0.529	0.274	0.382	0.416	0.481	0.436
BundleFusion[8]	Acc.↓	0.170	0.100	0.103	0.122	-	0.121	0.123	0.123
	Comp.↓	0.088	**0.030**	**0.038**	**0.057**	-	0.214	0.034	0.077
	F-score	0.373	0.571	0.474	0.470	-	0.442	0.527	0.476
iMAP[31]	Acc.↓	-	0.135	0.229	0.365	0.225	0.233	0.139	0.221
	Comp.↓	-	0.054	0.103	0.245	0.291	0.139	0.069	0.150
	F-score	-	0.445	0.315	0.238	0.170	0.255	0.416	0.307
NICE-SLAM[46]	Acc.↓	0.303	0.119	0.116	0.216	0.103	0.156	0.464	0.211
	Comp.↓	0.456	0.042	0.070	0.199	**0.089**	0.163	0.045	0.152
	F-score	0.221	0.554	0.411	0.402	0.400	0.273	0.401	0.380
Ours	Acc.↓	**0.081**	**0.068**	**0.067**	**0.079**	**0.078**	0.113	**0.121**	**0.087**
	Comp.↓	0.071	0.041	0.045	0.062	0.122	0.085	**0.033**	0.066
	F-score	**0.559**	**0.661**	**0.646**	**0.643**	**0.496**	**0.557**	**0.656**	**0.604**

Table 1. **Quantitative Comparison on Reconstruction.** We perform the mapping evaluation on 7 indoor sequences and report results of three metrics including accuracy (Acc.), completion (Comp.) and F-score. The failure cases are marked as "-".

Scene Name	Kitchen	Sofa	Office	Reception	Living room	Office2	Sofa2
ORB-SLAM3 [4]	**0.054**	-	**0.017**	**0.025**	-	**0.022**	-
ORB-SLAM3(with $\widetilde{D}$) [4]	0.082	**0.035**	0.019	0.049	-	0.058	-
KinectFusion [10]	-	0.146	0.209	0.157	-	0.321	0.125
ElasticFusion [38]	0.253	0.110	0.070	0.193	0.530	0.121	0.146
BundleFusion [8]	0.176	0.102	0.135	0.101	-	0.163	0.120
iMAP [31]	-	1.658	0.338	0.648	0.679	0.344	0.214
NICE-SLAM [46]	0.745	0.144	0.155	0.251	0.289	0.228	0.421
Ours	0.113	0.081	0.056	0.114	**0.200**	0.101	**0.085**

Table 2. **Camera Tracking Results.** ATE RMSE [m] (↓) is used as the evaluation metric. The failure cases are marked as "-". For the variations of ORB-SLAM3, we only mark the best one in each sequence. Our approach outperforms all the other methods except for ORB-SLAM3 [4]. However, ORB-SLAM3 fails on 3 of the 7 sequences due to the textureless indoor environment while our approach tracks successfully on all of the sequences.

ORB-SLAM3 cannot output dense models, we exclude it from the mapping evaluation.

Implementation Details. Our SLAM system is executed on a desktop PC equipped with an Intel i7-9700K CPU and an NVIDIA RTX 3090 GPU. In all our experiments, we set the grid sizes to [3, 6, 24, 96] cm and the number of sampling pixels M to 5000. Our method is implemented using PyTorch [27] with ADAM [12] optimizer. The decoders, multi-modal feature grids, and camera poses are trained with learning rates of 0.001, 0.01, and 0.0005, respectively. The loss weights are set to 10 for λ_c and λ_{sdf} and 1 for the others. Inspired by instant-ngp [22], we use the tiny-cuda-nn [21] library to implement the proposed multi-modal grid encoding, which significantly accelerates the optimization process. We follow DELTAR [14] to pre-train the multi-modal depth prediction network on NYU-V2 with simulated L5 signals. For more details, please refer to the supplementary material.

4.2. Evaluation of Mapping and Tracking

Mapping. We use three metrics to evaluate the reconstruction result including accuracy (Acc.), completion (Comp.) and F-score following the previous work [32]. For detailed descriptions of these metrics, please refer to the supplementary material. As shown in Table 1, our method outperforms all the baseline methods by a large margin (ours 0.604 vs. the second-best 0.476 in terms of F-score). We also show qualitative results in Fig. 5. Our method is able to produce high-quality 3D models with smooth surfaces and high accuracy. It is easy to notice that our reconstruction result has much fewer artifacts and noisy points. Since NICE-SLAM [46] relies on high-quality depth input, its performance is poor given the noisy and unreliable depth input.

Tracking. We use ATE RMSE [30] to evaluate the camera tracking. As shown in Table 2, our method outperforms all other methods except for ORB-SLAM3 [4]. However, ORB-SLAM3 is much less robust and tracks lost on 3 of

E	M	Kitchen	Sofa	Office	Reception	Living room	Office2	Sofa2	Avg.
		0.203	0.088	0.063	0.143	0.216	0.120	0.094	0.132
✓		0.135	0.085	0.060	0.144	0.211	0.108	0.089	0.119
✓	✓	**0.113**	**0.081**	**0.056**	**0.114**	**0.200**	**0.101**	**0.085**	**0.107**

Table 3. **Ablation Study.** We explore the efficiency of multi-modal scene representation (marked as "M") and L5 signal temporal filtering (marked as "E"). We use ATE RMSE [m] (↓) as the evaluation metric.

		Tracking	Mapping
Learning-based Methods	iMAP [31]	101 ms	448 ms
	NICE-SLAM [46]	470 ms	1300 ms
	Ours	116 ms	380 ms
Classical Methods	ORB-SLAM3 [4]	31 ms	159 ms
	ElasticFusion [38]	31 ms	-

Table 4. **Runtime Comparison.** Different from NICE-SLAM [46] that measures one iteration of optimization, we report the averaging total runtime for tracking and mapping respectively for better comparison with classic methods. ElasticFusion does not have an explicit mapping process thus it is labeled as "-".

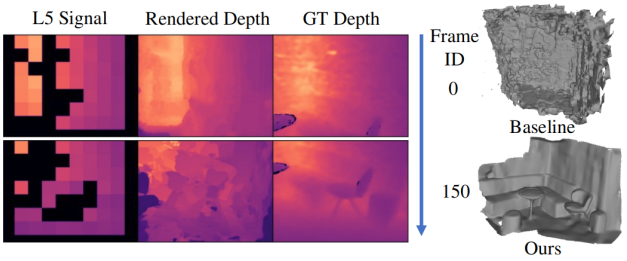

Figure 6. **Results without Pixel-Wise Depth Supervision.** It cannot guarantee plausible reconstruction results without the pixel-wise depth maps as additional supervision.

the 7 sequences, while our method successfully tracks all the sequences. This is because that ORB-SLAM3, as a keypoint-based method, struggles in texture-less regions, which are commonly seen in indoor environments. Given the predicted depth, ORB-SLAM3 can reduce the lost ratio from 42.8% to 28.5% but leads to worse camera tracking results since the predicted depth is not accurate enough.

Runtime Analysis. We compare the runtime of tracking and mapping in Table 4 for both classical methods and learning-based methods. The runtime of our method is similar to iMAP, but it is faster than NICE-SLAM due to our implementation optimization inspired by instant-ngp [22].

4.3. Ablation Study

In this section, we first study the effectiveness of using pixel-wise depth prediction as additional supervision, then investigate the importance of the proposed multi-modal feature grid representation and the temporal filtering technique.

Impact of the Pixel-Wise Depth Supervision. We try using only the L5 raw signals and RGB images for supervision. Removing the pixel-wise depth supervision makes

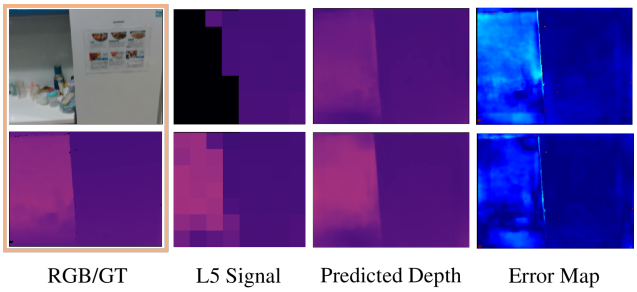

Figure 7. **Qualitative Study of the Effect of Temporal Filtering on Depth Prediction.** We compare the depth prediction result obtained through two methods: solely employing raw L5 signals (top row) and incorporating our temporal filtering technique (bottom row). Our temporal filtering technique enhances the raw L5 signals and fills up the missing regions, leading to a better depth prediction result.

the optimization problem harder, which is more obvious when the problem size grows. In Fig. 6 we show the rendered depth when optimizing the first frame and the first 150 frames. The latter contains serious errors. As a result, it leads to a distorted reconstruction, *i.e.*, the "Baseline" result in the upper right.

Impact of the Multi-Modal Feature Grid. We verify the effectiveness of our multi-modal feature grid representation (denoted as "M") in terms of camera tracking and show the result in Table 3. With the multi-modal feature grid representation, we are able to use raw L5 signal as supervision. As a result, we can optimize the camera poses on both pixel-level and zone-level, leading to better trajectory accuracy.

Impact of the Temporal Filtering. In Fig. 7 we show the qualitative comparison with and without the proposed temporal filtering. It can be seen that the large portion of missing L5 signals leads to severe errors on the corresponding regions of the predicted depth map. The error reduces significantly with the proposed temporal filtering technique.

We also study the quantitative impact of temporal filtering (marked as "E") on both the depth prediction (Table 5) and the camera tracking (Table 3). We divide the testing sequences into normal and hard cases according to the quality of L5 signals. As shown in Table 5, the temporal filtering technique improves the predicted depth in general and significantly in hard cases. Such an improvement benefits the final SLAM system and leads to more accurate camera

		$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	REL$\downarrow$	RMSE$\downarrow$
H	w/o E	0.773	0.910	0.969	0.159	0.513
	w E	**0.925**	**0.972**	**0.983**	**0.093**	**0.366**
N	w/o E	0.954	0.989	0.997	0.065	0.151
	w E	**0.956**	**0.992**	**0.998**	**0.065**	**0.145**

Table 5. **Ablation Study on Temporal Filtering for Depth Prediction.** "H"/"N" stands for the hard/normal cases and "E" represents the temporal filtering. We report the quantitative result of depth maps predicted from L5 raw signals and our refined signals.

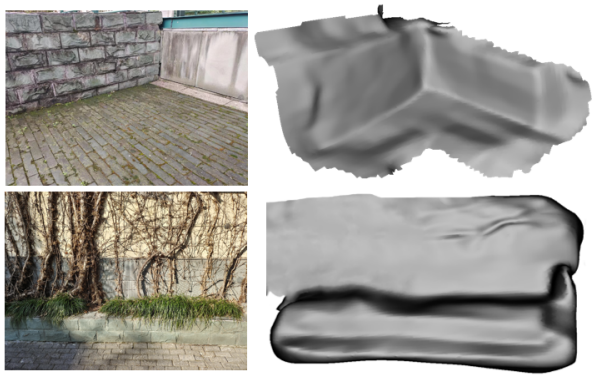

Figure 8. **Qualitative Results of Outdoor Reconstruction.** The proposed method can recover a complete scene model in outdoor scenes but missing more details than that in indoor due to the poor L5 signals.

tracking, as shown in Table 3 (marked as "E"). Please refer to the supplementary material for the definition of the normal/hard cases and the depth evaluation metrics.

4.4. Performance in Outdoor Scenes

Currently, we focus on indoor scenes as other NeRF-based SLAM systems [31, 46] due to the optical interference limitation of depth sensors. Actually, the sunlight has a stronger interference on the L5 sensor for its low-power emitter. In Fig. 8, we show two examples of outdoor scene reconstruction using our proposed methods. The recovered mesh is complete but missing more details than that in indoor scenes due to the poor L5 signals.

5. Conclusion

We introduce a novel dense visual SLAM framework working with RGB cameras and light-weight ToF sensors using neural implicit scene representation. To accommodate this new input modality, we propose a novel multi-modal feature grid that enables both zone-level rendering for the ToF sensors and pixel-level rendering for other high-resolution signals. To guarantee robust tracking and mapping, we exploit a per-pixel depth prediction as additional supervision, which is further improved by a novel temporal filtering strategy. Our experiments demonstrate that the proposed method can provide accurate camera tracking and high-quality reconstruction result on indoor scenes. Similar to other NeRF-based RGB-D SLAM systems, as future work we plan to further improve the system to overcome the limitation of ToF sensors in outdoor scenarios and make it efficient enough to run on mobile robots.

Acknowledgements: This work was partially supported by NSF of China (No. 62102356).

References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural RGB-D surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6290–6301, 2022.

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[3] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. CodeSLAM—learning a compact, optimisable representation for dense visual SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2560–2568, 2018.

[4] Carlos Campos, Richard Elvira, Juan J. Gómez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[5] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial radiance fields. In *Proceedings of the European Conference on Computer Vision*, pages 333–350. Springer, 2022.

[6] 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.

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017.

[8] 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 Transactions on Graphics (ToG)*, 36(4):1, 2017.

[9] Zilong Dong, Guofeng Zhang, Jiaya Jia, and Hujun Bao. Keyframe-based real-time camera tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1538–1545. IEEE, 2009.

[10] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. KinectFusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User Interface Software and Technology*, pages 559–568, 2011.

[11] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual SLAM for RGB-D cameras. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2100–2106. IEEE, 2013.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234. IEEE, 2007.

[14] Yijin Li, Xinyang Liu, Wenqi Dong, Han Zhou, Hujun Bao, Guofeng Zhang, Yinda Zhang, and Zhaopeng Cui. DELTAR: Depth estimation from a light-weight tof sensor and rgb image. In *Proceedings of the European Conference on Computer Vision*, pages 619–636. Springer, 2022.

[15] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5741–5751, 2021.

[16] Haomin Liu, Mingyu Chen, Guofeng Zhang, Hujun Bao, and Yingze Bao. ICE-BA: Incremental, consistent and efficient bundle adjustment for visual-inertial SLAM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1974–1982, 2018.

[17] Haomin Liu, Guofeng Zhang, and Hujun Bao. Robust keyframe-based monocular SLAM for augmented reality. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pages 1–10. IEEE, 2016.

[18] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020.

[19] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021.

[20] 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 *Proceedings of the European Conference on Computer Vision*, pages 405–421. Springer, 2020.

[21] Thomas Müller. tiny-cuda-nn. https://github.com/NVlabs/tiny-cuda-nn, 4 2021.

[22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022.

[23] Raúl 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.

[24] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. DTAM: Dense tracking and mapping in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2320–2327. IEEE, 2011.

[25] Junjie Ni, Yijin Li, Zhaoyang Huang, Hongsheng Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. PATS: Patch area transportation with subdivision for local feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17776–17786, 2023.

[26] 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 *Proceedings of the Robotics: Science and Systems*, 2022.

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

[28] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to sift or surf. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2564–2571. Ieee, 2011.

[29] STMicroelectronics. Time-of-Flight 8x8 multizone ranging sensor with wide field of view. https://www.st.com/en/imaging-and-photonics-solutions/vl53l5cx.html. Accessed 19-Jul-2022.

[30] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580. IEEE, 2012.

[31] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6229–6238, 2021.

[32] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. NeuralRecon: Real-Time Coherent 3D Reconstruction from Monocular Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15593–15602, Nashville, TN, USA, 2021. IEEE.

[33] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021.

[34] Zachary Teed and Jia Deng. DeepV2D: Video to depth with differentiable structure from motion. In *Proceedings of the International Conference on Learning Representations*, 2019.

[35] Zachary Teed and Jia Deng. Droid-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cam-

eras. *Advances in Neural Information Processing Systems*, 34:16558–16569, 2021.

[36] Jingwen Wang, Tymoteusz Bleja, and Lourdes Agapito. GO-Surf: Neural feature grid optimization for fast, high-fidelity RGB-D surface reconstruction. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 433–442. IEEE, 2022.

[37] 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. *Advances in Neural Information Processing Systems*, 34:27171–27183, 2021.

[38] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. ElasticFusion: Real-time dense SLAM and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016.

[39] Bangbang Yang, Chong Bao, Junyi Zeng, Hujun Bao, Yinda Zhang, Zhaopeng Cui, and Guofeng Zhang. NeuMesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In *Proceedings of the European Conference on Computer Vision*, pages 597–614. Springer, 2022.

[40] Bangbang Yang, Yinda Zhang, Yijin Li, Zhaopeng Cui, Sean Fanello, Hujun Bao, and Guofeng Zhang. Neural rendering in a room: amodal 3d understanding and free-viewpoint rendering for the closed scene composed of pre-captured objects. *ACM Transactions on Graphics (TOG)*, 41(4):1–10, 2022.

[41] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13779–13788, 2021.

[42] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. iNeRF: Inverting neural radiance fields for pose estimation. in 2021 ieee. In *Proceedings of the RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1323–1330, 2021.

[43] 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*, 35:25018–25032, 2022.

[44] Guofeng Zhang, Jiaya Jia, Tien-Tsin Wong, and Hujun Bao. Recovering consistent video depth maps via bundle optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[45] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. NeRF++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

[46] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022.