NeSLAM: Neural Implicit Mapping and Self-Supervised Feature Tracking With Depth Completion and Denoising

Tianchen Deng, Yanbo Wang, Hongle Xie, Hesheng Wang, *Senior Member, IEEE*, Jingchuan Wang, Danwei Wang, *Fellow, IEEE*, Weidong Chen, *Member, IEEE*

Abstract—In recent years, there have been significant advancements in 3D reconstruction and dense RGB-D SLAM systems. One notable development is the application of Neural Radiance Fields (NeRF) in these systems, which utilizes implicit neural representation to encode 3D scenes. This extension of NeRF to SLAM has shown promising results. However, the depth images obtained from consumer-grade RGB-D sensors are often sparse and noisy, which poses significant challenges for 3D reconstruction and affects the accuracy of the representation of the scene geometry. Moreover, the original hierarchical feature grid with occupancy value is inaccurate for scene geometry representation. Furthermore, the existing methods select random pixels for camera tracking, which leads to inaccurate localization and is not robust in real-world indoor environments. To this end, we present NeSLAM, an advanced framework that achieves accurate and dense depth estimation, robust camera tracking, and realistic synthesis of novel views. First, a depth completion and denoising network is designed to provide dense geometry prior and guide the neural implicit representation optimization. Second, the occupancy scene representation is replaced with Signed Distance Field (SDF) hierarchical scene representation for high-quality reconstruction and view synthesis. Furthermore, we also propose a NeRF-based self-supervised feature tracking algorithm for robust real-time tracking. Experiments on various indoor datasets demonstrate the effectiveness and accuracy of the system in reconstruction, tracking quality, and novel view synthesis.

Index Terms—Neural Radiance Fields, Dense RGB-D SLAM, 3D Reconstruction, Novel View Synthesis.

I. INTRODUCTION

Visual Simultaneous Localization and Mapping (SLAM) has made significant progress and has various applications in different fields, such as autonomous driving, indoor robotics, and virtual reality (VR). For real-world deployment, a system must possess several essential properties. Firstly, it should be capable of incrementally constructing an accurate geometric representation of the scene and estimating camera pose in real-time. Secondly, the system should demonstrate robustness in handling noisy and incomplete observations, while also
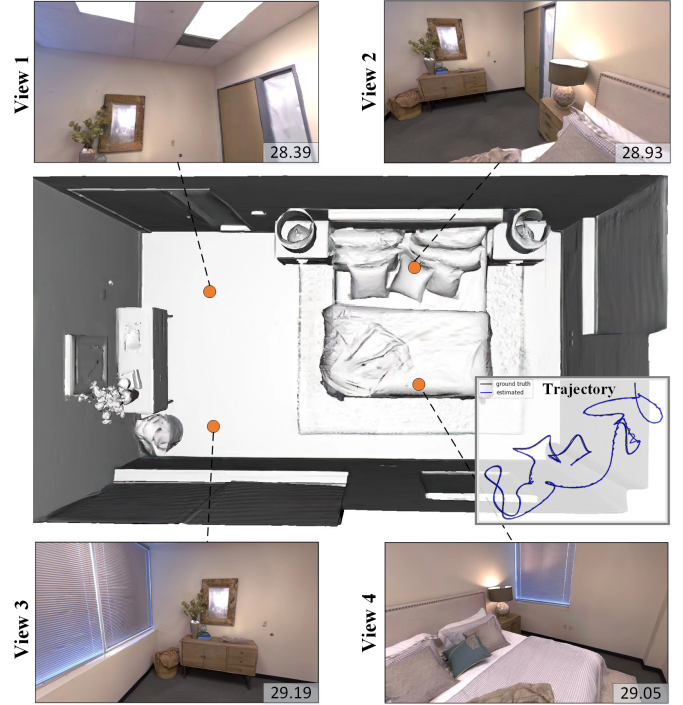
Fig. 1. 3D reconstruction and novel view synthesis results using NeSLAM. The final reconstruction mesh and images of novel view synthesis at different locations showcase the powerful scene reconstruction capability of our algorithm. We provide the PSNR value in the bottom right corner.

being scalable to handle large-scale scenarios. Additionally, the ability to synthesize novel views can provide valuable benefits for applications such as virtual reality roaming.

As for existing visual SLAM systems, there are several categories of them, such as sparse map points SLAM systems [1]–[4], and dense SLAM systems [5]–[7]. Those systems are able to perform real-time pose estimation and can be employed in large-scale scenes with loop closing [8]. However, they fall short in terms of their scene representation capabilities. They tend to inadequately capture and incorporate essential information, resulting in incomplete and limited scene representations. Sparse map representation methods [9] are not suitable for subsequent tasks in robotics, such as navigation and obstacle avoidance [10], [11]. With the rapid advances in deep learning, some learning-based SLAM systems are successively proposed to improve the ability of scene representation, such as Codeslam and Scenecode [12]–[14].

arXiv:2403.20034v1 [cs.CV] 29 Mar 2024

Compared with other representation methods, Neural radiance fields (NeRF) [15] is a promising recent advance technology with various application in robotics and autonomous driving [16]. NeRF utilizes differentiable rendering techniques and multi-layer perceptrons (MLP) to estimate the density and color of each point along a ray. The MLP has the ability to encode scene geometry in fine detail. Adopting these implicit representation methods in SLAM, there are several recently proposed systems such as iMAP [17] and NICE-SLAM [18], and so on [19]–[22] . Both of them successfully combine NeRF with SLAM and achieve real-time pose estimation and dense mapping.

However, there are two key challenges for dense visual SLAM. The first challenge arises from the inherent limitations of consumer-grade RGB-D sensors, which result in sparse and noisy depth images. These characteristics pose a considerable obstacle to neural implicit mapping, as they heavily rely on accurate geometry information. The second challenge lies in the limitations of existing methods when it comes to tracking in real-world indoor scenes. These methods use random random pixel selection strategy, which often exhibit low tracking accuracy and are prone to failure.

To this end, we propose NeSLAM, a dense RGB-D SLAM system that can represent the scene implicitly, camera tracking, and have the ability of novel view synthesis. For the first challenge, a depth completion and denoising network is proposed. This network aims to generate dense and precise depth images with depth uncertainty images. This geometry prior information plays a crucial role in guiding neural point sampling and optimizing the neural implicit representation. This network is used to improve the geometry representation capability and refine the performance of the entire system.

For the second challenge, we propose a NeRF-based self-supervised feature tracking network specifically designed for accurate and real-time camera tracking in indoor scenes. This network leverages the strengths of NeRF with feature tracking to enable self-supervised optimization during the system operation, which can enhance the generalization capability. The keypoint network can better adapt to different complex scenes and make the system more stable, accurate, and scalable. We evaluate the effectiveness of the method on different indoor RGB-D datasets and do exhaustive evaluations and ablation experiments on these datasets. Our system demonstrates superior performance compared to recent and concurrent methods [17], [18] that employ implicit mapping approaches. **In summary, our contributions are shown as follows:**

- A novel dense visual SLAM system is proposed with hierarchical implicit scene representation. This system is scalable, predictive, and robust to complex indoor scenes. It is an end-to-end, incrementally optimizable method for tracking and mapping. It offers the capability of generating photo-realistic novel views and producing accurate 3D meshes.
- A depth completion and denoising network is designed to provide dense and accurate depth images associated with depth uncertainty images. This geometry prior information is used to guide the point sampling process and improve geometric consistency. In addition, we replace

the occupancy value with Signed Distance Field (SDF) value to better represent scene geometry.
- We propose a NeRF-based self-supervised feature tracking method for accurate and robust camera tracking in large and complex indoor environments, which is proven effectiveness and robust in our experiments.

II. RELATED WORK

Visual SLAM System   Traditional real-time visual SLAM systems depend on the constructed maps. PTAM [23], a breaking SLAM work with parallel tracking and mapping, provides an effective method for keyframe selection, feature matching, and camera localization for every frame. Some sparse mapping methods [1], [24], [25], which are then proposed that use manipulated keypoint for tracking, mapping, relocalization, and loop closing. These systems are robust to severe motion clutter and large indoor environments.

For learning-based SLAM systems, DTAM [5] is one of the pioneer works that use the dense map and view-centric scene representation. Some recent dense SLAM systems, such as [26], adopt the framework of DTAM to estimate pose and depth. Kinectfusion [27] explicitly represents the surface of the environments with a fixed resolution of volume, but it is costly in memory. Bundle-Fusion and Ba-net [28], [29] are dense SLAM systems that successfully use bundle adjustment for pose estimation. Other methods, such as CodeSLAM [12] propose a new compact but dense representation of scene geometry with a latent code. And [30] use the probabilistic field on the Lie group Sim(3) manifold for SLAM in a dynamic environment. In contrast to these methods, we use implicit scene mapping, which allows us to achieve more accurate geometry representation and novel view synthesis along the trajectory.

Implicit Scene Representation   Scene reconstruction has made significant progress recently [31], [32]. With the proposal of Neural radiance fields (NeRF) [15], many researchers explore to combine this implicit method into 3D reconstruction. NeRF is a ground-breaking method for novel view synthesis. It represents the scene with an MLP and renders images with the predicted volume densities along the rays. However, the representation of volume densities can not commit the geometric consistency, leading to poor surface prediction for reconstruction tasks. In order to deal with it, some methods are proposed that combine world-centric 3D geometry representation with neural radiance fields, such as UNISURF [33] and NeuS [34]. UNISURF uses a unified way to formulate the implicit surface model with radiance fields. It enables more efficient points sampling and reconstructs accurate surfaces without input masks. NeuS [34] replaces the volume density with Signed Distance Field (SDF) values. It proposes a new rendering formulation and incorporates additional depth measurements. Other methods [35]–[39] use various scene geometry representation methods, such as truncated signed distance function, voxel grid, or occupancy grid with latent codes. However, they all need ground-truth camera poses.

NeRF with SLAM   Some works focus on pose estimation of NeRF, iNeRF [40], NeRF– [41] are concurrent work to
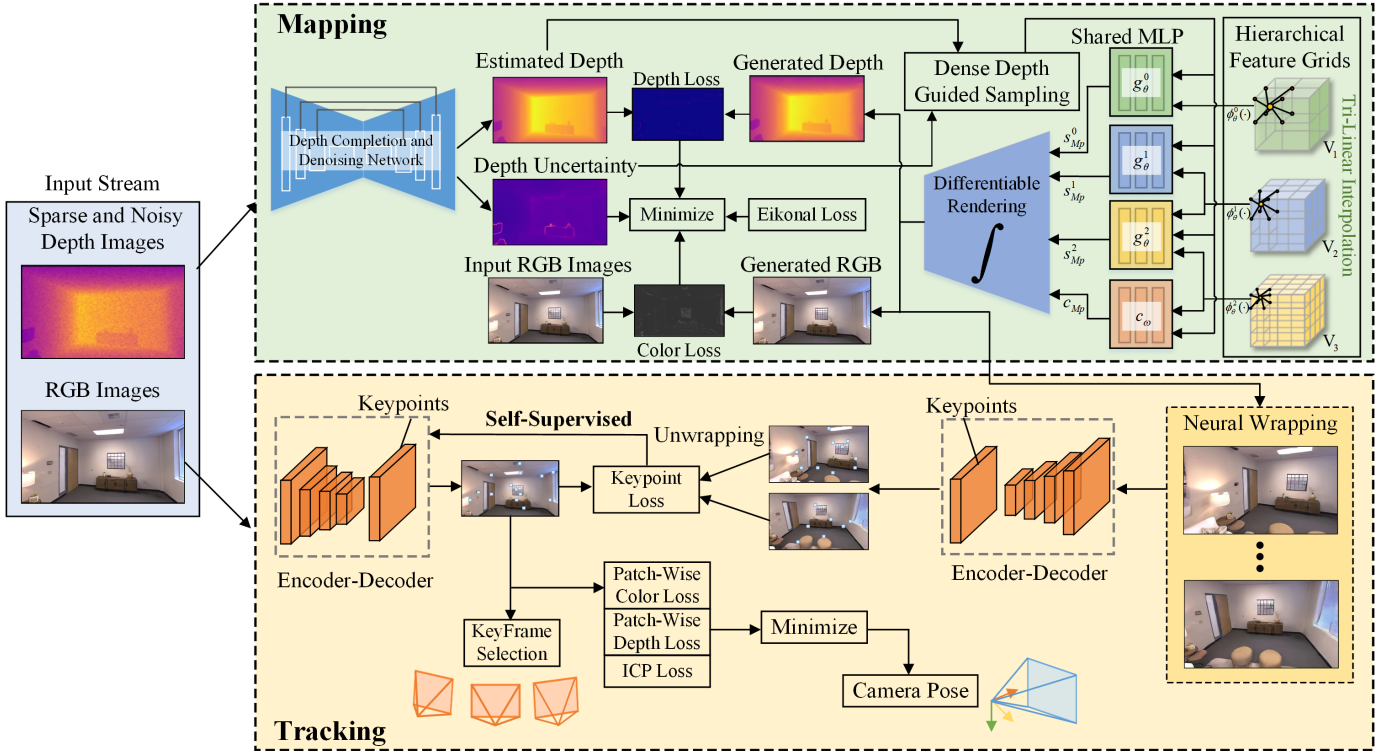
Fig. 2. The pipeline of our system. The input stream of our system is RGB and depth images, and the output is the implicit scene representation, generated RGB, depth images, depth uncertainty images, and the camera pose. Our system has two parallel threads: the mapping thread and the tracking thread. In the mapping thread, we estimate the dense and accurate depth image along with depth uncertainty. Then we use them to guide the neural point sampling and implicit representation optimization. The hierarchical feature grids are online updated by minimizing our carefully designed loss through differentiable rendering with the system operating. As for the tracking thread, we propose a NeRF-based self-supervised feature tracking network for accurate and robust pose estimation. This network is online self-supervised optimized via backpropagating keypoint loss. Those two threads are running with an alternating optimization.

estimate the camera pose with inverse NeRF optimization when the neural implicit network is fully trained. Without the pre-trained neural implicit network, BARF [42] is proposed to train a neural network with inaccurate poses images or unknown poses images through bundle adjustment. However, their methods can not optimize poses and neural implicit network simultaneously. Pushing this to the limits, iMAP [17], and NICE-SLAM [18] are successively proposed to combine neural implicit mapping with SLAM. iMAP uses a single multi-layer perceptron (MLP) to represent the scene, and NICE-SLAM uses a learnable hierarchical feature grid. These are the works most relevant to our approach, but our method differs from them in the following ways. With the designed depth completion and denoising network, we can get more accurate reconstruction and novel view synthesis. We also propose a self-supervised feature tracking method for robust pose estimation in complex environments.

III. METHOD

A. System Overview

The pipeline of our system is shown in Fig. 2. Following the prior works [17], [18], we use three-level hierarchical feature grids and their corresponding decoders to represent the scene geometry. We also use another feature grid and corresponding decoder for color representation. For the implicit mapping thread, a depth completion and denoising network (Sec. III-B) is designed to estimate dense depth images along with depth uncertainty images to strengthen geometry representation ability and improve sampling efficiency. Then the dense depth images and depth uncertainty are used to guide the neural point sampling and NeRF optimization. We also incorporate hierarchical neural scene representation with SDF into our system (Sec. III-C). For the camera tracking thread, a self-supervised feature tracking method (Sec. III-D) is designed for robust and accurate pose estimation. Several carefully designed loss functions are proposed to jointly optimize the scene implicit representation and camera pose estimation (Sec. III-E). The network is incrementally online and updated with the system operation.

B. Depth Completion and Denoising Network

With the limitations of consumer-grade RGB-D cameras, the input depth images have two downsides. Firstly, the input depth images are relatively sparse because depth cameras often fail to sense depth for shiny, bright, transparent, and distant surfaces. Secondly, the input depth images are often noisy and have outliers, which is harmful for implicit geometry representation. In order to address those two downsides, we propose our depth completion and denoising network D_θ inspired by [43], [44]. The architecture of our depth network is shown in Fig. 3. The input of our network is RGB images I_i, and sparse depth images D_i. The output of our network is initial depth prediction D_{ini}, non-local neighbor affinities ω_i, confidence map γ_i, and standard deviations S_i

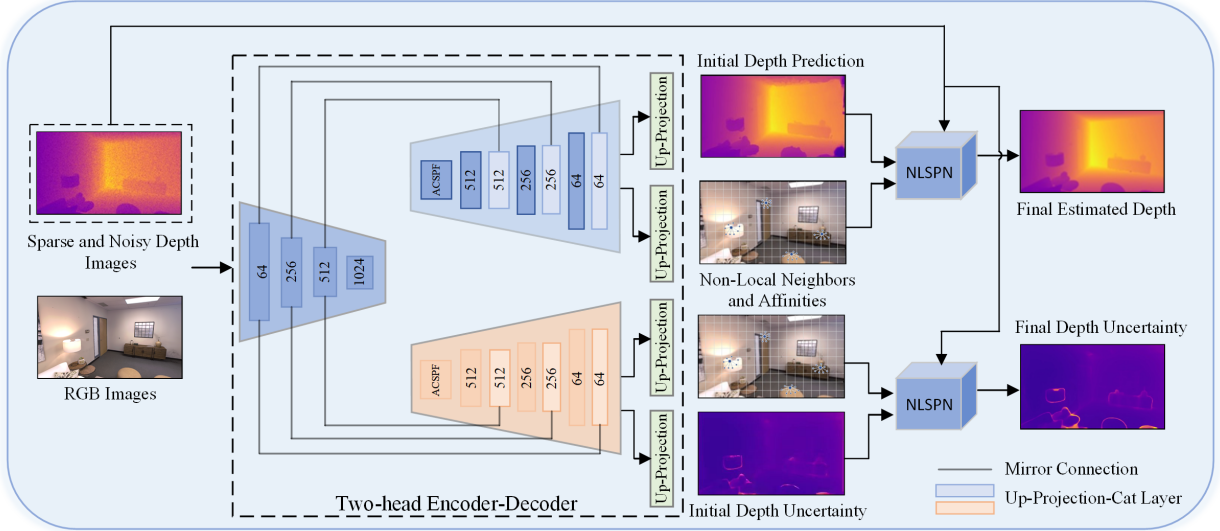$$D_\theta(I_i, D_i) = (D_{ini}, \omega_i, \gamma_i, S_i) \qquad (1)$$

Fig. 3. The architecture of our depth completion and denoising network. We use sparse and noisy depth images and corresponding RGB images as our input. We design a two-head encoder-decoder architecture to estimate dense depth along with depth uncertainty. We use mirror connections to add feature information from the encoder to the Up-Projection-Cat layer. The sparse depth map is embedded into the NLSPN module to guide the depth refinement.

We use the residual network [45] as the backbone of our two-head encoder-decoder architecture, which is a UNet-like [46] neural network. We design a two branches architecture network with mirror connections to predict dense depth D_i jointly with standard deviation S_i. The detailed parameters of the network are annotated in Fig. 3. In order to avoid the spatial information weaken with the down-sampling operation of the network, we add mirror connections by directly concatenating the feature from the encoder to decoder layers, which is the "Up-Projection-Cat" layer in Fig. 3. The feature dimensions of each layer of our encoder are 64, 256, 512, 1024. And the output feature dimensions of each layer of the decoder are 512, 512, 256, 256, 64, and 64, respectively. We utilize an ACSPF module [44], which combines Convolutional Spatial Pyramid Pooling (CSPP), Atrous Spatial Pyramid Pooling (ASPP), and Convolutional Feature Fusion (CFF) modules. The Up-Projection layer in Figure 3 is composed of the convolutional (conv) layer, batch normalization (bn) layer, and upsampling layer (bilinear interpolation). For better spatial information propagation, we employ Non-Local Spatial Propagation Network (NLSPN) [47] to refine depth and depth uncertainty. This network uses non-local spatial propagation to estimate missing values and refine less confident values by propagating neighbor values with corresponding affinities. This refinement procedure makes the blurry depth images become more detailed. We also incorporate the confidence map γ_i of the depth prediction to avoid negative influence from unreliable depth values during non-local propagation. This helps us get better results in depth completion and denoising.

For network training, we train our model on Replica [48] dataset, Scannet [49] dataset, and TUM RGB-D [50] dataset. Under the assumption that the depth and standard deviation are normally distributed, we use the negative log-likelihood of a Gaussian loss:

$$L_\theta = \frac{1}{n} \sum_{i=1}^{n} (log(S_i^2) + \frac{(\hat{D}_i - D_{gt})^2}{S_i^2}) \qquad (2)$$

where D_i, S_i are the estimated depth and uncertainty of pixel i. n is the number of valid pixels in depth images.

C. Neural Scene Representation

Scene Representation Following NeRF [15] and NICE-SLAM [18], we incorporate hierarchical scene representation into our system. We use multi-level grid features with corresponding pre-trained MLPs for scene geometry representation. Inspired by VolSDF [51], we change the occupancy with the Signed Distance Field (SDF) value which greatly improves the ability of geometry representation. For geometry representation, the feature grid is encoded into three levels: coarse $g_\theta^0(\cdot)$, middle $g_\theta^1(\cdot)$, fine $g_\theta^2(\cdot)$. With the corresponding geometry MLP decoder $g_\theta(\cdot)$, we can get the SDF value s_{Mp} and geometry feature z_{Mp} by querying the decoder. For any map point $Mp \in \mathbb{R}^3$:

$$coarse: s_{Mp}^0, z_{Mp}^0 = g_\theta^0(Mp, \phi_\theta^0(Mp))$$
$$middle: s_{Mp}^1, z_{Mp}^1 = g_\theta^1(Mp, \phi_\theta^1(Mp))$$
$$fine: s_{Mp}^2, z_{Mp}^2 = g_\theta^2(Mp, \phi_\theta^1(Mp), \phi_\theta^2(Mp)) \qquad (3)$$

where θ is an optimizable parameter for feature grids. The optimization for geometry is a coarse-to-fine process. We first use the mid-level grid to represent the coarse-level scene geometry and use the fine-level grid for refinement. For the coarse and mid-level grid, the features are directly decoded into SDF values and features with corresponding MLPs. For the fine-level grid, it is a residual value of the mid-level grid. We concatenate the mid-level feature $\phi_\theta^1(Mp)$ with the fine-level feature $\phi_\theta^2(Mp)$ as the input of the fine-level decoder. The output of the fine-level decoder is an offset from mid-level SDF value. The final SDF value $\hat{s}$ is defined as:

$$\hat{s} = s_{Mp}^1 + s_{Mp}^2 \qquad (4)$$

In our framework, these three pre-trained decoders are fixed for optimization stabilization and geometric consistency. We only optimize the feature grids $g_\theta(\cdot)$ during the optimization
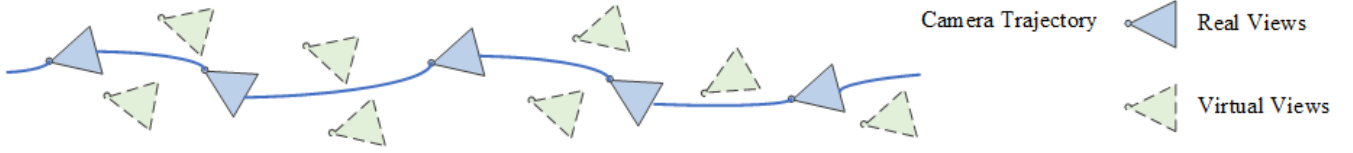
Fig. 4. The generation of virtual images. We synthesize novel view images with different poses $\{R, T\}$ for every coming keyframe. we generate its RGB and depth images and filter out inaccurate depths through a geometric consistency check.

process. The coarse-level feature grid is primarily used to extract low-frequency information (such as contours), while the fine-level feature grid is used to extract high-frequency information (such as detailed texture features) from the environment.

For color representation, we use another feature grid φ_ω and decoder c_ω:

$$color : c_{Mp} = c_\omega(Mp, z_{Mp}^0, z_{Mp}^1, z_{Mp}^2, \varphi_\omega(Mp)) \quad (5)$$

where ω is the learnable parameter of the color feature grid. During the optimization process, we jointly optimize the feature grids $\varphi_\omega(Mp)$ and decoder c_ω for global color consistency and incrementally learning. With the prior of [18], the feature dimension is 64 and 5 layers for the geometry and two layers for color decoders. We also incorporate the Gaussian positional encoding [18], [52] to Mp for better learning of high-frequency details of both color and geometry.
Differentiable Rendering   Following NeRF [15], we use the predicted SDF value and colors from decoders and integrate them for scene representation. We can determine a ray $r(t) = o + td$ whose origin is at the camera center of projection o. We sample points along this ray. The sample bound is within the near and far planes $t_k \in [t_n, t_f]$, $k \in 1, \ldots, K$. For every sample point Mp_k, we can get three level SDF values and color of them. We follow VolSDF [51] to transform the SDF value into density value:

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

where $\beta \in \mathbb{R}$ is a learnable parameter that controls the sharpness of the surface boundary. Then we define the termination probability as:

$$coarse : \omega_k^c = \prod_{j=1}^{k-1} \exp\left(-\sigma_j^c \delta_j\right)\left(1 - \exp\left(-\sigma_k^c\right)\right)$$

$$fine : \omega_k^f = \prod_{j=1}^{k-1} \exp(-\sigma_j^f \delta_j)(1 - \exp(-\sigma_k^f)) \quad (7)$$

where δ_j represents the distance between neighboring sample points. Then the color, depth, and standard deviation D_s of the ray are computed from the rendering weights ω_k:

$$\hat{C} = \sum_{k=1}^K \omega_k^f c_k \quad \hat{D}^f = \sum_{k=1}^K \omega_k^f t_k \quad \hat{D}^c = \sum_{k=1}^k \omega_k^c t_k$$

$$\hat{S}f^2 = \sum_{k=1}^K \omega_k^f(D_f - t_k)^2 \quad \hat{S}c^2 = \sum_{k=1}^K \omega_k^c(D_c - t_k)^2 \quad (8)$$

Depth Guided Sampling   The estimated depth images and depth uncertainty provide valuable geometry information which can guide neural point sampling along a ray within the bounds of depth uncertainty. For a room-scale scene in the Replica dataset, we get N_{strat} points for stratified sampling between the near and far planes. Then, $N_{surface}$ points are drawn from the Gaussian distribution determined by the depth prior $\mathcal{N}(D, S^2)$. When the depth is not known or invalid, we use the estimated depth prior from differentiable rendering and sample points according to $\mathcal{N}(\hat{D^f}, \hat{Sf}^2)$. Compared to the original methods, such as NeRF or NICE-SLAM, our approach allows for more efficient point sampling and enhances the scene representation capability of the network.

D. NeRF-Based Self-Supervised Feature Tracking

We parallel run this thread for pose estimation in real-time e.g.the rotation and translation $\{R, T\}$. In prior work [18], they random sample P_t pixels in the current frame to optimize the camera pose. However, random sampling is not fit for large scenes and noising observations, which are really common in real-world environments. The accuracy of their method is low, the robustness is poor, and the efficiency is also low. They fail in many situations, such as quick camera movement and large scenes. We consider that the keypoint is more suitable due to its inherent properties of rotation and translation invariance. To this end, we propose a Nerf-based self-supervised feature tracking network and incorporate it into our camera tracking thread. It can self-supervised optimize during the system operation compared with a superpoint network and achieve high localization accuracy in different scenarios.
Network Architecture   With the prior work [53], we use a fully-convolutional neural network architecture. The input is full-sized images, and the output is keypoints detections. We use a VGG-style encoder to reduce the dimensionality of the image. The encoder maps the input image $I \in \mathbb{R}^{H \times W}$ to a feature map $\mathcal{F} \in \mathbb{R}^{H_f \times W_f \times \mathcal{B}}$, where $H_f = H/8$ and $W_f = H/8$.

For the keypoint decoder, it uses $\mathcal{X} \in \mathbb{R}^{H_f \times H_f \times 65}$ tensor as input. We use 65 channels which contain 8×8 grid regions of pixels and an extra dustbin for no interest point area. Then we use a channel-wise softmax and remove the dustbin channel after that. The output is reshaped as $\mathbb{R}^{H \times W}$.
NeRF-Based Self-Supervised Refinement   The original superpoint model [53] is trained on the MS-COCO image dataset [54] with homographic adaptation for domain adaptation. However, the pre-trained superpoint model is unsuitable for different real-world datasets. We want to incrementally optimize the superpoint model with the operating of our system. So we propose a NeRF-based self-supervised refinement
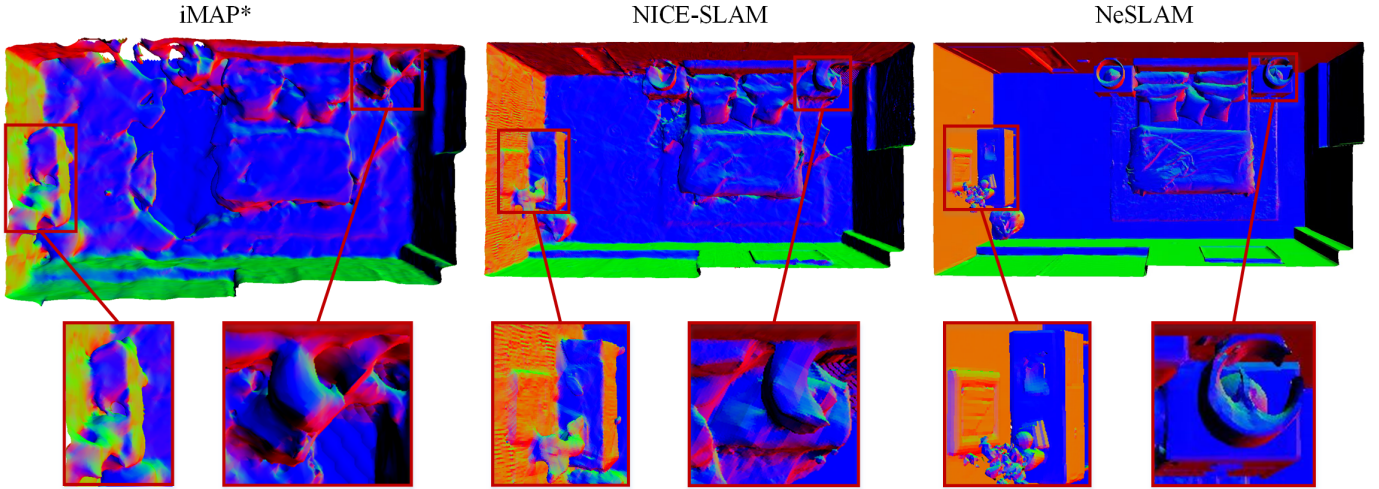
Fig. 5. Qualitative reconstruction results on the Replica dataset [48] of room 1. From left to right, we show the construction meshes of iMAP* [17], NICE-SLAM [18], and our method. The red box highlights the improvements of our algorithm compared to other algorithms.

TABLE I
RECONSTRUCTION RESULTS ON THE REPLICA DATASET [48]. WE USE THREE DIFFERENT METRICS ACC.↓, COMP.↓, COMP. RATIO↑. IMAP* REFER TO
THE RE-IMPLEMENTATION OF IMAP PROVIDED IN [18].

Methods	Metrics	room-0	room-1	room-2	office-0	office-1	office-2	office-3	office-4	Avg.
iMAP* [17]	Acc.[cm] ↓	3.28	3.49	4.48	5.57	3.41	4.72	4.09	4.61	4.21
	Comp.[cm] ↓	4.96	4.74	5.31	6.01	5.13	5.51	5.29	6.47	5.43
	Comp. Ratio[<5cm %] ↑	82.73	82.16	74.43	76.53	78.84	75.03	76.14	75.83	77.72
NICE-SLAM [18]	Acc.[cm] ↓	2.93	2.97	3.03	4.86	2.95	3.71	3.04	2.65	3.27
	Comp.[cm] ↓	2.95	2.92	2.87	3.95	3.63	3.24	3.51	3.65	3.34
	Comp. Ratio[<5cm %] ↑	91.55	87.25	94.03	86.04	87.83	87.35	87.05	89.58	88.83
NeSLAM	Acc.[cm] ↓	**2.55**	**2.11**	**2.14**	**2.13**	**3.02**	**3.23**	**2.91**	**2.45**	**2.57**
	Comp.[cm] ↓	**2.32**	**2.31**	**2.27**	**1.64**	**1.67**	**2.93**	**3.03**	**3.55**	**2.46**
	Comp. Ratio[<5cm %] ↑	**91.78**	**94.67**	**91.97**	**95.55**	**94.56**	**90.91**	**90.49**	**91.32**	**92.66**

method to achieve this. In our self-supervised approach, we use the pre-trained model for the base interest point detector. Then, we propose a novel neural wrapping procedure to get some different views of images for data augmentation. The generation of virtual images is shown in Fig. 4. In this procedure, we synthesize novel view images with different poses $\{R, T\}$ for every coming keyframe. We input these images into the network to get the interest points detections. Then we unwrap these images into the initial pose and calculate the keypoint L_p loss. The unwrapping procedure can be formulated as follows:

$$\mathcal{X}' = \mathcal{X}_{\{R,t\}} \langle \Pi \left(D, \{R, T\}, K \right) \rangle \quad (9)$$

where $\mathcal{X}, \mathcal{X}'$ are the current keyframe and the corresponding unwrapped synthesis images. D is the depth image and K is the camera intrinsics. Operator $\Pi()$ is the resulting 2D coordinates of projection. $\langle \rangle$ is the sampling operator. Then we can calculate keypoint loss:

$$L_p = \frac{1}{H_c W_c} \sum_{h=1, w=1}^{H_c, W_c} l_p(x_{hw}; x'_{hw}) \quad (10)$$

where l_p:

$$l_p\left(\mathbf{x}_{hw}; \mathbf{x}'_{hw}\right) = -\log \left(\frac{\exp\left(\mathbf{x}_{hwx'}\right)}{\sum_{k=1}^{65} \exp\left(\mathbf{x}_{hwk}\right)} \right) \quad (11)$$

l_p is cross-entropy loss over the cells $x_{hw} \in \mathcal{X}$, $x'_{hw} \in \mathcal{X}'$ from the current keyframe and the corresponding unwrapped synthesis images.

E. Optimization in Mapping and Tracking

In this section, we provide more details of the optimization of scene geometry θ, color ω, and camera poses $\{R, T\}$.

To optimize the scene feature grid in Section III-C, we uniformly sample P_t pixels from the current frame and the selected keyframes. Then, we iteratively optimize the feature grid to minimize the depth and color loss. The depth loss is defined as:

$$L_D^l = \frac{1}{P_t} \sum_{i=1}^{P_t} (log(\hat{S}_i^{l\,2}) + \frac{\hat{D}_i^l - D_i^l}{\hat{S}_i^{l\,2}}) \quad (12)$$

Here D_i^l and S_i^l are the target depth and standard deviation, $l \in c, f$. $\hat{D}_i^l$ and $\hat{S}_i^l$ are the estimated depth and standard deviation. We apply this loss to the pixel where one of the
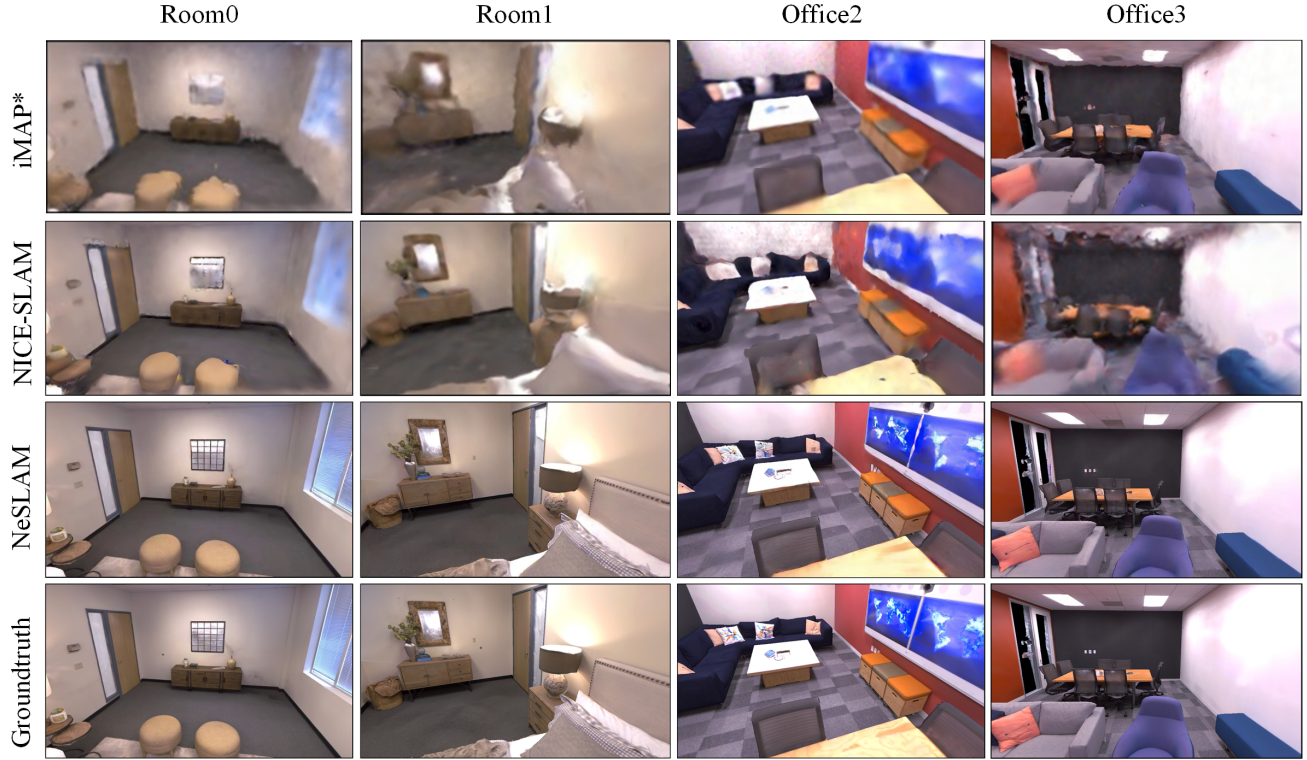
Fig. 6. Qualitative results on the Replica dataset [48]. We show the view synthesis results of iMAP* [17], NICE-SLAM [18], and our method. Our method performs better than other methods with higher-quality view synthesis results.
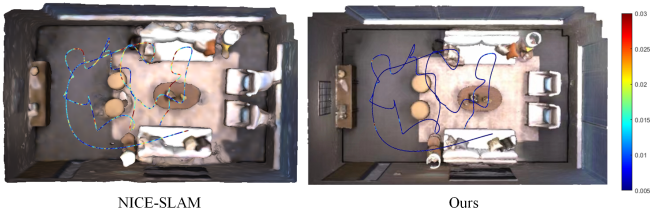

Fig. 7. We depict the final mesh and camera tracking trajectory error (Absolute Trajectory Error) of different methods in replica dataset [48]. The color bar on the right shows the ATE value.
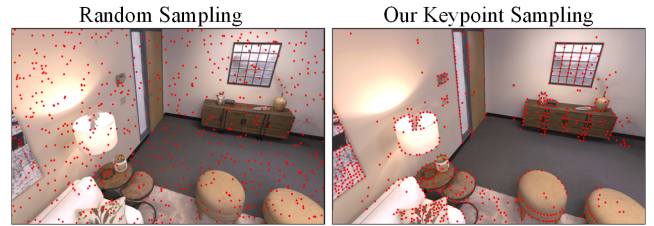

Fig. 8. Qualitative results of our self-supervised feature tracking network on the Replica dataset. We show the sampling points of different methods.

following conditions is true: 1) $|\hat{D}_i^l - D_i^l| > S_i^l$, the distance between generated depth and input depth is greater than the standard deviation value. 2) $\hat{S}_i^l > S_i^l$, the generated depth standard deviation. We also use this loss to optimize the pre-trained depth denoising network incrementally. The color loss is defined as:

$$L_c = \frac{1}{P} \sum_{i=1}^{P} \left\| \hat{C}_i - C_i \right\|_1 \tag{13}$$

where $\hat{C}_i$ and C_i are the estimated color and target color. Inspired by NICE-SLAM, we use geometry loss to optimize mid-level feature at the first stage. Then we also use L_D^l to jointly optimize mid and fine level feature. In addition, we add the Eikonal loss [55] to regularize the output SDF values:

$$\mathcal{L}_{\text{eikonal}} = \sum_{\mathbf{x} \in \mathcal{X}} \left(\|\nabla \hat{s}(\mathbf{x})\|_2 - 1 \right)^2 \tag{14}$$

where $\mathcal{X}$ are a set of uniformly sampled near-surface points. Finally, we jointly optimize all level feature grids and the color decoder with the loss:

$$\min_{\theta, \omega} (L_D^f + L_D^c + \lambda_c L_c + \lambda_e L_{eikonal}) \tag{15}$$

This multi-stage optimization can lead to better geometry consistency and convergency.

Camera Tracking We use the extracted keypoints in the current frame to optimize the camera pose. We apply color loss in Eq.(13), and modified depth loss:

$$L_{D\_v} = \frac{1}{P_t} \sum_{i=1}^{P_t} \frac{\left\| D_i - \hat{D}_i^c \right\|_1}{\hat{S}^c} + \frac{\left\| D_i - \hat{D}_i^f \right\|_1}{\hat{S}^f} \tag{16}$$

This depth modified loss avoid less certain regions make influence on the reconstructed geometry.

Patch-Wise Loss Furthermore, we replace original depth and color loss with patch-wise depth variance loss $L_{p\_D\_v}$, patch-wise color loss L_{p_c}, and patch-wise depth loss L_{p_D}. We use 3×3 patch for every interest point to obtain better

TABLE II
CAMERA TRACKING RESULTS ON THE REPLICA DATASET [48] OF TRADITIONAL METHODS AND LEARNING-BASED METHODS.

Methods	Metrics	room-0	room-1	room-2	office-0	office-1	office-2	office-3	office-4	Avg.
iMAP	RMSE[m] ↓	0.0553	0.0459	0.0239	0.0247	0.0177	0.0495	0.0697	0.0267	0.0391
	Mean[m] ↓	0.0345	0.0407	0.0206	0.0178	0.0165	0.0327	0.0591	0.0229	0.0306
NICE-SLAM	RMSE[m] ↓	0.0225	0.0238	0.0199	0.0148	0.0128	0.0198	0.0223	0.0235	0.0199
	Mean[m] ↓	0.0191	0.0207	0.0156	0.0113	0.0107	0.0157	0.0185	0.0188	0.0163
NeSLAM	RMSE[m] ↓	0.0060	0.0093	**0.0052**	**0.0041**	0.0043	**0.0057**	0.0096	**0.0083**	0.0066
	Mean[m] ↓	0.0053	0.0082	**0.0045**	**0.0037**	0.0038	**0.0045**	0.0076	**0.0065**	0.0056
ORB-SLAM2(RGB)	RMSE[m] ↓	0.0050	0.0043	0.0225	0.0049	0.0048	0.1225	0.0077	0.1137	0.0356
	Mean[m] ↓	0.0044	0.0038	0.0199	0.0037	0.0041	0.1102	0.0065	0.0938	0.0308
ORB-SLAM2(RGB-D)	RMSE[m] ↓	**0.0034**	**0.0027**	0.0057	0.0048	**0.0039**	0.0058	**0.0087**	0.0098	**0.0055**
	Mean[m] ↓	**0.0030**	**0.0021**	0.0051	0.0039	**0.0032**	0.0048	**0.0071**	0.0085	**0.0047**

Fig. 9. This is our wheelchair prototype, serving as our data collection platform. We also present a panoramic image of the indoor scenario. We use Realsense D435i to collect color and depth images and use VICON as groundtruth.
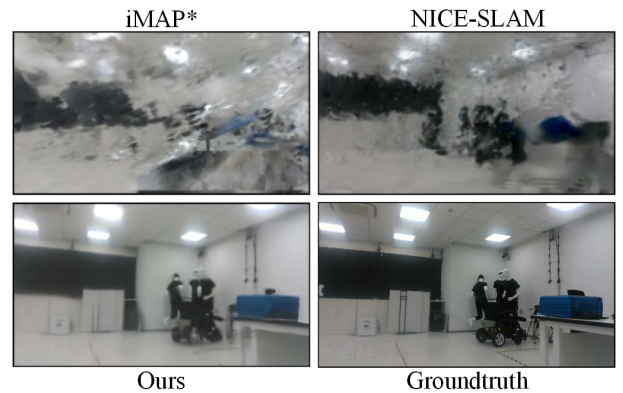

Fig. 10. Qualitative results on our own real-world datasets. We show the view synthesis results of iMAP* [17], NICE-SLAM [18], and our method in scenes with fast camera movements and noisy input.

Fig. 11. Qualitative results on our own real-world datasets (hospital). We show the view synthesis results of iMAP* [17], NICE-SLAM [18], and our method in real-world robot operational scenario.

gradient descent and convergency. Finally, we incorporate ICP loss into our system to explicitly express the camera pose in loss function.

$$L_{ICP} = \sum_{i=1}^{P_t} \left\| \mathcal{C}(X_a^i - \Pi_{R,T}(X_b)) \right\|_1 \qquad (17)$$

where X_a^i is ith keypoint of frame a, X_b are the keypoints from frame b. Then, we project the keypoint from frame b into frame a and find the closest matching $\mathcal{C}$ of those keypoints. The ICP loss is defined as the pixel coordinate loss of the matching keypoints. The final tracking loss is defined as:

$$\min_{R,T}(L_{p\_d\_v} + \lambda_{p\_D} L_{p\_D} + \lambda_1 L_{p\_c} + \lambda_2 L_{ICP}) \qquad (18)$$

we formulate this minimization problem to optimize camera poses.

IV. EXPERIMENTS

A. Implementation Details

We evaluate our method on various datasets and conduct a comprehensive ablation study to verify the effectiveness of our design. All training and evaluation experiments are conducted on a single NVIDIA RTX 3090 GPU. In all our experiments, we use N_{strat} = 32 and $N_{surface}$ = 16 sampling points on a ray. The color loss weighting is λ_c = 0.3 and λ_e = 0.1 for mapping and λ_1 = 0.15 for tracking. The

TABLE III
GEOMETRIC (DEPTH L1) AND PHOTOMETRIC (PSNR) RESULTS ON THE REPLICA [48] DATASETS. IMAP* REFERS TO THE RE-IMPLEMENTATION OF IMAP PROVIDED IN [18].

Methods	Metrics	room-0	room-1	room-2	office-0	office-1	office-2	office-3	office-4	Avg.
iMAP* [17]	Depth L1 [cm] ↓	5.80	5.27	5.67	7.49	11.87	8.22	7.74	6.12	7.27
	PSNR [db] ↑	20.17	20.37	19.98	24.37	23.01	18.07	24.03	21.55	21.44
NICE-SLAM [18]	Depth L1 [cm] ↓	1.81	1.44	2.04	1.39	1.76	8.33	4.99	2.01	2.97
	PSNR [db] ↑	24.31	22.52	21.07	26.93	28.79	20.45	25.07	22.37	23.93
NeSLAM	Depth L1 [cm] ↓	**1.25**	**2.01**	**1.67**	**1.02**	**0.91**	**4.02**	**2.81**	**1.53**	**1.90**
	PSNR [db] ↑	**27.72**	**25.37**	**24.56**	**27.19**	**30.37**	**27.28**	**27.22**	**26.56**	**27.03**

TABLE IV
COMPARISION OF RUNTIME IN REPLICA DATASET [48].

Methods	Tracking [s]	Mapping [s]
iMAP [17]	101.45	448.85
NICE-SLAM [18]	47.88	140.74
Ours	**44.58**	**130.58**

ICP loss weighting is $\lambda_2 = 0.2$ and patch-wise depth loss is $\lambda_{p\_D} = 0.35$. For small-scale datasets, such as Replica, we select five active frames for mapping. For large-scale real-world datasets, we select ten active frames for mapping. we select Adam optimizer [56] ($\beta = (0.9, 0.999)$) for scene representation and camera tracking optimization. The learning rate for tracking on Replica, ScanNet, and TUM RGB-D dataset is 1×10^{-3}, 5×10^{-4}, 1×10^{-2}.

B. Evaluation Datasets and Metrics

We operate our system in different datasets ranging from small room scenes to large indoor scenes. We also collect our own dataset to evaluate the performance of our system in real-world scenarios and its deployment on mobile robots. To evaluate scene reconstruction results, we choose the Replica dataset [48], which is a synthetic 3D indoor dataset from a room to an entire apartment scale. In order to create a more realistic depth input, we randomly remove the depth of some pixels and perturb the depth with Gaussian noise $\mathbb{N}(0, s^2)$, where the standard deviation increases with the depth value. For camera tracking, we use TUM RGB-D dataset [50] to evaluate pose estimation with the given groundtruth trajectory. Moreover, we consider ScanNet [49] to evaluate the scalability of our system. Following [17], [18], we evaluate Accuracy, Completion, Completion Ratio [$< 5cm\%$], and Depth L1 metrics for scene geometry. As for the evaluation of camera tracking results, we use Absolute Trajectory Error (ATE) Root Mean Squared Error (RMSE), Mean, and Median. We also use Peak Signal-to-noise Ratio (PSNR) to evaluate novel view synthesis results. Please note that iMAP* is the re-implementation of iMAP provided in [18].

For Replica datasets [48], it is a synthetic dataset. So, we use the processed RGB-D sequence with noisy depth input to better simulate real-world environments. The Gaussian noise is set to $\mathbb{N}(0, 0.8)$. The quantitative and qualitative reconstruction results are shown in Table I and Fig. 5. With the depth denoising and completion network and improved hierarchical scene representation method, our method can

TABLE V
CAMERA TRACKING RESULTS ON OUR OWN DATASET AND TUM RGB-D DATASETS [50]. WE USE ATE RMSE [CM] AS OUR EVALUATION METRIC.

Methods	fr1/desk	fr2/xyz	fr3/office	ROOM
iMAP [17]	4.93	2.04	5.84	6.34
NICE-SLAM [18]	2.75	1.83	3.02	4.73
DI-Fusion [57]	4.45	2.39	15.73	6.39
Ours	**1.83**	**1.09**	**2.14**	**2.95**
BAD-SLAM [6]	1.89	1.21	1.83	2.88
ElasticFusion [58]	2.04	1.27	1.71	3.21
ORB-SLAM2 [1]	**1.63**	**0.62**	**1.36**	2.97

TABLE VI
CAMERA TRACKING RESULTS ON THE SCANNET DATASETS [49]. WE USE ATE RMSE [CM] AS OUR EVALUATION METRIC.

Scene ID	0000	0059	0106	0169	0181	0207	Avg.
iMAP* [17]	55.95	32.06	17.50	70.51	32.10	11.91	36.67
NICE-SLAM [18]	8.64	12.25	8.09	10.28	12.93	5.59	9.63
Ours	**6.87**	**7.37**	**5.23**	**9.07**	**9.27**	**4.08**	**6.98**

reconstruct the scene more precisely. In Fig. 5, we can see that our algorithm significantly outperforms other algorithms in reconstruction accuracy, smoothness, and completeness. To better showcase the reconstructed results, we have zoomed in on a specific region of the images. The left and right red boxes show the zoomed-in reconstruction results of the desk and the lamp, respectively. In Table I, we can see that the accuracy metric is 21.4% higher than NICE-SLAM. The improved hierarchical scene representation method with SDF value greatly enhances the capability of scene representation of our method. Our proposed depth denoising and completion algorithm also improves the reconstruction accuracy in the presence of noisy inputs, while other algorithms exhibit low accuracy when dealing with noisy inputs.

The camera tracking results are shown in Table II. Our method outperforms all NeRF-based SLAM systems in all metrics. Compared with NICE-SLAM, our RMSE metric is 65.3% higher on average, thanks to the NeRF-based self-supervised feature tracking method. Compared with the traditional SLAM system [1], We can achieve competitive camera tracking performance, while providing dense and high-fidelity scene reconstruction performance. The dense mapping of the scene is really important in robots navigation and human interaction. The qualitative results of sampling points are shown
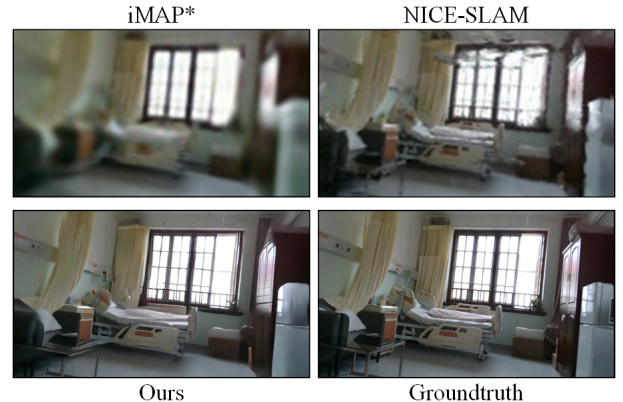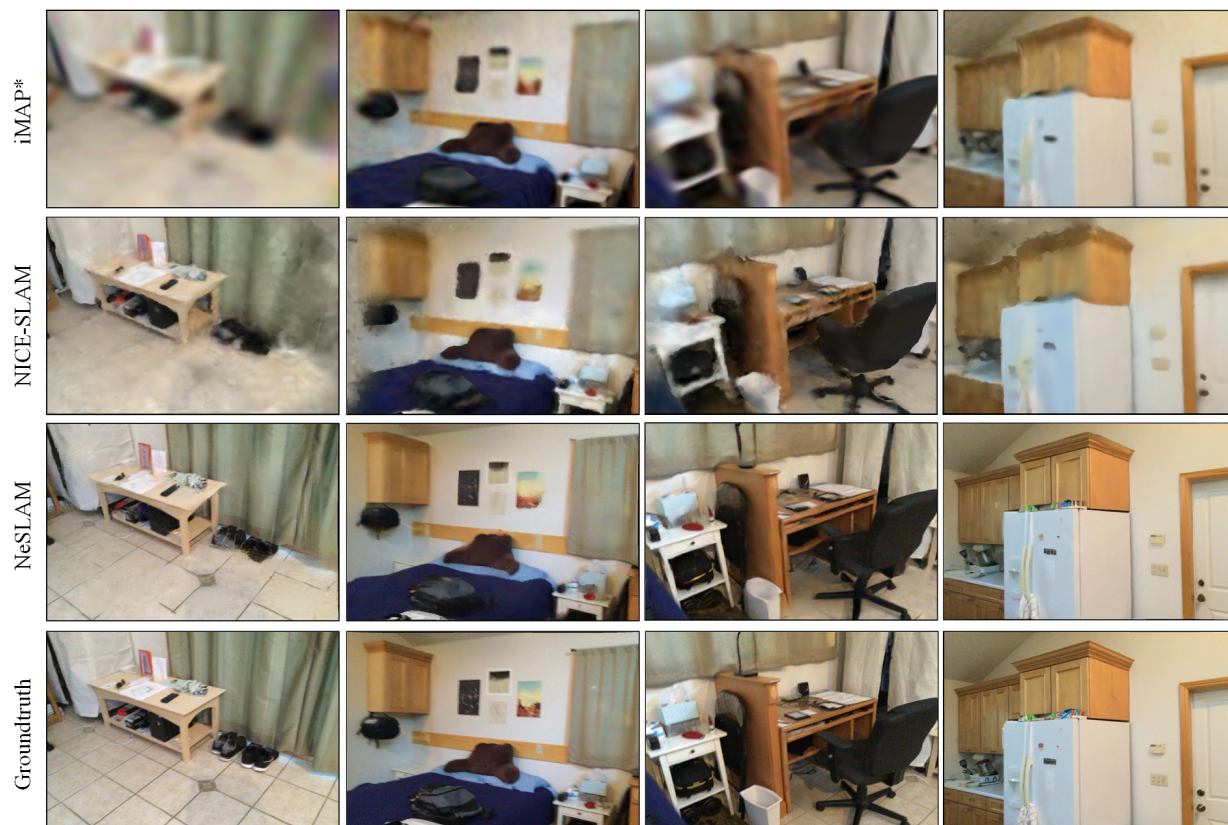
Fig. 12. Qualitative results on the Scannet dataset [49]. We show the view synthesis results of iMAP* [17], NICE-SLAM [18], and our method. In large real-world indoor scenes, our method outperforms other algorithms in scene representation and view synthesis.

TABLE VII
ABLATION STUDY ON DIFFERENT DATASETS OF DIFFERENT MODULE.

Methods		Replica	ScanNet	TUM RGB-D	ROOM
(a)w/o D_θ	Acc.[cm] ↓	3.17	-	-	-
	Depth[cm] ↓	2.37	23.98	7.89	8.92
	RMSE[cm] ↓	0.73	7.93	1.82	3.29
	PSNR[db] ↑	25.56	22.80	21.84	21.76
(b)w/o SDF	Acc.[cm] ↓	3.84	-	-	-
	Depth[cm] ↓	2.31	23.28	7.13	8.79
	RMSE[cm] ↓	0.83	8.87	2.08	3.47
	PSNR[db] ↑	25.43	22.43	21.23	21.18
(c)w/o FT-Ref	Acc.[cm] ↓	3.01	-	-	-
	Depth[cm] ↓	2.03	23.19	6.99	8.68
	RMSE[cm] ↓	0.81	8.213	2.02	3.62
	PSNR[db]	25.75	22.51	22.01	21.57
(d)w/o PW Loss	Acc.[cm] ↓	2.87	-	-	-
	Depth[cm] ↓	1.92	21.97	6.82	8.53
	RMSE[cm] ↓	0.74	7.82	1.83	3.24
	PSNR[db]	26.86	23.01	22.73	22.72
(e)w/o ICP Loss	Acc.[cm] ↓	2.97	-	-	-
	Depth[cm]	1.98	21.82	6.93	8.75
	RMSE[cm]	0.73	8.03	1.98	3.44
	PSNR[db]	26.02	22.97	23.26	22.31
NeSLAM (Full)	Acc.[cm]	**2.57**	-	-	-
	Depth[cm]	**1.90**	**20.37**	**6.75**	**8.41**
	RMSE[cm]	**0.69**	**6.98**	**1.68**	**3.01**
	PSNR[db]	**27.03**	**23.88**	**23.87**	**23.59**

in Fig. 8. Our methods can effectively leverage environmental information for localization. The view synthesis results and depth estimation results are shown in Table III and Fig. 6. Compared with iMAP [17] and NICE-SLAM [18], Our depth L1 metric is **33%** better than NICE-SLAM [18]. Our PSNR metric is **48%** better than NICE-SLAM [18]. Fig. 6 provides the qualitative comparison of view synthesis between different methods. Our method achieves the most high-fidelity novel views results.

For our own real-world datasets, we collect data from two different scenes: a laboratory environment and a hospital ward scene. We use them to evaluate camera tracking performance and view synthesis in small room scenes with rapid camera movement, limited perspective, and relatively sparse view. As shown in Fig. 9, we present our mobile robot platform equipped with a camera (Realsense D435i) and LiDAR (RS Lidar-16). We also present the scenario of our datasets. The room is equipped with the VICON motion capture system V2.2, which we use as ground truth for our dataset. We also use the lidar sensor to provide the groundtruth of our dataset. In Fig. 10, we show the qualitative results of view synthesis in a real-world scenario (hospital). Our algorithm achieves better image synthesis results compared with other methods. In Table V, we compare our method with other methods in real-world datasets. Our method performs better than iMAP [17]and NICE-SLAM [18] (with implicit representation) and reduces the gap between implicit SLAM with traditional SLAM (ORB-SLAM [1], BAD-SLAM [6], ElasticFusion [58]). Due to the self-supervised keypoint detection,
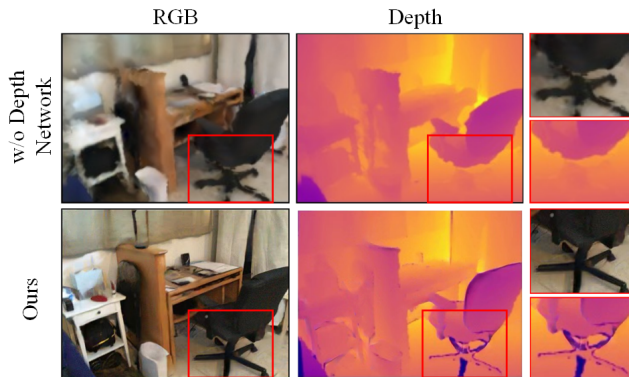
Fig. 13. Qualitative results of our depth network ablation study on the Scannet dataset. We show the impact of the depth completion and denoising network on the final results.

our system is more accurate and robust for different scenes. With the limitations in obtaining ground truth, we are unable to compare the localization accuracy in the hospital ward scene. So we only show the qualitative results in the hospital scene in Fig. 11. We also present our camera tracking results on the TUM RGB-D dataset [50].

For ScanNet datasets [49], we employ this dataset to evaluate the performance of our algorithm in large real-world indoor scenarios. We select different scenes to evaluate the scalability, camera tracking accuracy, and view synthesis results. As shown in Table VI, compared with iMAP and NICE-SLAM, our method performs better in tracking accuracy. Our feature tracking algorithm provides more accurate and robust results in larger-scale scenes. In Fig. 12, we show the qualitative results of view synthesis. Our mapping algorithm effectively addresses the issue of noise input in real-world environments. Our algorithm achieves the best image synthesis results with high clarity and completeness. We also compare the runtime for tracking and mapping. We modify our code to achieve better performance in time consumption. Thanks to the keypoint detection model, we can achieve better tracking performance and time consumption with fewer sampling pixels. We use 44 milliseconds for tracking and 147 milliseconds for mapping, compared with NICE-SLAM (50 milliseconds for tracking and 145 milliseconds for mapping). Our method is also robust to rapid camera movement and sudden frame loss. We provide extensive experiments in the supplementary material **https://github.com/dtc111111/NeSLAM**.

C. Ablation Study

In this section, we conduct sufficient ablation studies to verify the effectiveness of our designed network. We show our ablation results in Table VII. (a) is NeSLAM without depth denoising and completion network. (b) is NeSLAM without SDF scene representation. (c) is NeSLAM without self-supervised feature tracking refinement (d) is NeSLAM without patch-wise loss (e) is NeSLAM without ICP loss.

Depth Denoising and Completion Network   In Table VII (a), we remove our designed depth network. It is obvious that this network has a great influence on depth L1 and PSNR metrics. This network significantly improves the capacity of scene geometry representation and enhances geometric

consistency and robustness for noisy input. The qualitative results of our ablation study on depth network are shown in Fig. 10. The depth network aids in recovering the geometric representation, ensuring geometric consistency across multi-view, and improving the results of depth estimation and view synthesis.

Hierarchical Scene Representation with SDF   In Table VII (b), we replace the SDF hierarchical scene representation with original occupancy value. Our reconstruction and view synthesis metrics show a significant decrease. It indicates that the SDF transformation is really helpful in scene reconstruction.

Feature Tracking Network   In Table VII (c), we cancel the refinement of our self-supervised feature tracking network. We can see that the refinement network plays an important role in accurate camera tracking. It also makes our system more robust to rapid camera movement and sudden frame loss.

Loss Function Design   As displayed in Table VII (d), we use the original color and depth loss (without patch-wise loss). The reconstruction and tracking accuracy decreases, which verifies the effectiveness of this design. Table VII (e) shows that the RMSE metric decreases greatly without the ICP loss. It is obvious that explicitly expressing the pose into loss function is effective for tracking.

V. Conclusion

This paper proposes a dense SLAM system NeSLAM, which combines neural implicit scene representation with the SLAM system. A depth denoising and completion network and a self-supervised feature tracking network are proposed. Our depth network provides dense depth images with depth uncertainty which can guide the neural point sampling and enhance scene geometry consistency. In addition, we incorporate the Signed Distance Field (SDF) value into the hierarchical feature grid, which can better represent scene geometry. Furthermore, the proposed NeRF-based self-supervised feature tracking network enables accurate camera tracking and enhances the robustness of our system. Our extensive experiments demonstrate the effectiveness and accuracy of our system in both scene reconstruction, tracking, and view synthesis in complex indoor scenes. In our future work, we will focus on dynamic scenes, aiming to achieve high reconstruction and localization accuracy.

References

[1] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[2] T. Deng, H. Xie, J. Wang, and W. Chen, "Long-term visual simultaneous localization and mapping: Using a bayesian persistence filter-based global map prediction," *IEEE Robotics & Automation Magazine*, vol. 30, no. 1, pp. 36–49, 2023.

[3] H. Xie, T. Deng, J. Wang, and W. Chen, "Robust incremental long-term visual topological localization in changing environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–14, 2022.

[4] ——, "Angular tracking consistency guided fast feature association for visual-inertial slam," *IEEE Transactions on Instrumentation and Measurement*, 2024.

[5] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2011, pp. 2320–2327.

[6] T. Schops, T. Sattler, and M. Pollefeys, "Bad slam: Bundle adjusted direct rgb-d slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 134–144.

[7] H. Matsuki, R. Scona, J. Czarnowski, and A. J. Davison, "Codemapping: Real-time dense mapping for sparse slam using compact scene representations," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7105–7112, 2021.

[8] M. U. M. Bhutta, M. Kuse, R. Fan, Y. Liu, and M. Liu, "Loop-box: Multiagent direct slam triggered by single loop closure for large-scale mapping," *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 5088–5097, 2022.

[9] J. Liu, R. Yu, Y. Wang, Y. Zheng, T. Deng, W. Ye, and H. Wang, "Point mamba: A novel point cloud backbone based on state space model with octree-based ordering strategy," *arXiv preprint arXiv:2403.06467*, 2024.

[10] X. Liu, Z. Lin, Y. Niu, Z. Lyu, Q. Xu, B. Cui, and T. Deng, "A multi-uav cooperative search system design based on man-in-the-loop," in *2020 3rd International Conference on Unmanned Systems (ICUS)*. IEEE, 2020, pp. 757–762.

[11] T. Deng, "Research on aerial robot based on visual servo," in *Journal of Physics: Conference Series*, vol. 1678, no. 1. IOP Publishing, 2020, p. 012007.

[12] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison, "Codeslam—learning a compact, optimisable representation for dense visual slam," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2560–2568.

[13] S. Zhi, M. Bloesch, S. Leutenegger, and A. J. Davison, "Scenecode: Monocular dense semantic reconstruction using learned encoded scene representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 776–11 785.

[14] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in neural information processing systems*, vol. 34, pp. 16 558–16 569, 2021.

[15] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European Conference on Computer Vision*, 2020.

[16] T. Deng, S. Liu, X. Wang, Y. Liu, D. Wang, and W. Chen, "Prosgnerf: Progressive dynamic neural scene graph with frequency modulated autoencoder in urban scenes," *arXiv preprint arXiv:2312.09076*, 2023.

[17] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October 2021, pp. 6229–6238.

[18] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2022, pp. 12 786–12 796.

[19] T. Deng, G. Shen, T. Qin, J. Wang, W. Zhao, J. Wang, D. Wang, and W. Chen, "Plgslam: Progressive neural scene represenation with local to global bundle adjustment," *arXiv preprint arXiv:2312.09866*, 2023.

[20] M. Li, S. Liu, and H. Zhou, "Sgs-slam: Semantic gaussian splatting for neural dense slam," *arXiv preprint arXiv:2402.03246*, 2024.

[21] M. Li, J. He, G. Jiang, and H. Wang, "Ddn-slam: Real-time dense dynamic neural implicit slam with joint semantic encoding," *arXiv preprint arXiv:2401.01545*, 2024.

[22] T. Deng, Y. Chen, L. Zhang, J. Yang, S. Yuan, D. Wang, and W. Chen, "Compact 3d gaussian splatting for dense visual slam," *arXiv preprint arXiv:2403.11247*, 2024.

[23] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Proceedings of the IEEE/ACM International Conference on Symposium on Mixed and Augmented Reality*, 2007, pp. 225–234.

[24] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[25] T. Deng, H. Xie, J. Wang, and W. Chen, "Long-term visual simultaneous localization and mapping: Using a bayesian persistence filter-based global map prediction," *IEEE Robotics & Automation Magazine*, vol. 30, no. 1, pp. 36–49, 2023.

[26] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "Demon: Depth and motion network for learning monocular stereo," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, July 2017.

[27] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. 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*, 2011, pp. 559–568.

[28] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, 2017.

[29] C. Tang and P. Tan, "Ba-net: Dense bundle adjustment network," in *Proceedings of the International Conference on Learning Representations*, September 2018.

[30] X. Gao, X. Liu, Z. Cao, M. Tan, and J. Yu, "Dynamic rigid bodies mining and motion estimation based on monocular camera," *IEEE Transactions on Cybernetics*, pp. 1–12, 2022.

[31] R. Fan, U. Ozgunalp, Y. Wang, M. Liu, and I. Pitas, "Rethinking road surface 3-d reconstruction and pothole detection: From perspective transformation to disparity map segmentation," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 5799–5808, 2022.

[32] S. Zhao, X. Wang, D. Zhang, G. Zhang, Z. Wang, and H. Liu, "Fm-3dfr: Facial manipulation-based 3-d face reconstruction," *IEEE Transactions on Cybernetics*, pp. 1–10, 2023.

[33] M. Oechsle, S. Peng, and A. Geiger, "Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October 2021, pp. 5589–5599.

[34] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 27 171–27 183.

[35] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural rgb-d surface reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2022, pp. 6290–6301.

[36] A. Bozic, P. Palafox, J. Thies, A. Dai, and M. Niessner, "Transformerfusion: Monocular rgb scene reconstruction using transformers," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 1403–1414.

[37] J. Choe, S. Im, F. Rameau, M. Kang, and I. S. Kweon, "Volumefusion: Deep depth fusion for 3d scene reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October 2021, pp. 16 086–16 095.

[38] J. Sun, Y. Xie, L. Chen, X. Zhou, and H. Bao, "Neuralrecon: Real-time coherent 3d reconstruction from monocular video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2021, pp. 15 598–15 607.

[39] C. Xia, Y. Shen, Y. Yang, X. Deng, S. Chen, J. Xin, and N. Zheng, "Onboard sensors-based self-localization for autonomous vehicle with hierarchical map," *IEEE Transactions on Cybernetics*, vol. 53, no. 7, pp. 4218–4231, 2023.

[40] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "inerf: Inverting neural radiance fields for pose estimation," in *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 1323–1330.

[41] 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.

[42] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October 2021, pp. 5741–5751.

[43] B. Roessle, J. T. Barron, B. Mildenhall, P. P. Srinivasan, and M. Nießner, "Dense depth priors for neural radiance fields from sparse input views," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2022, pp. 12 892–12 901.

[44] X. Cheng, P. Wang, and R. Yang, "Learning depth with convolutional spatial propagation network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2361–2379, 2020.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

[46] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceeding of the International Conference in Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

[47] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. So Kweon, "Non-local spatial propagation network for depth completion," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 120–136.

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

[49] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.

[50] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 573–580.

[51] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 4805–4815.

[52] M. Tancik, P. Srinivasan, B. Mildenhall, and Fridovich-Keil, "Fourier features let networks learn high frequency functions in low dimensional domains," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 7537–7547.

[53] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2018.

[54] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 740–755.

[55] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, "Implicit geometric regularization for learning shapes," in *Proceeding of the International Conference on Machine Learning*, 2020, pp. 3789–3799.

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, 2015.

[57] J. Huang, S.-S. Huang, H. Song, and S.-M. Hu, "Di-fusion: Online implicit 3d reconstruction with deep priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8932–8941.

[58] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph." Robotics: Science and Systems, 2015.