IR-MCL: Implicit Representation-Based Online Global Localization

Haofei Kuang Xieyuanli Chen Tiziano Guadagnino Nicky Zimmerman Jens Behley Cyrill Stachniss

***Abstract*—Determining the state of a mobile robot is an essential building block of robot navigation systems. In this paper, we address the problem of estimating the robot's pose in an indoor environment using 2D LiDAR data and investigate how modern environment models can improve gold standard Monte-Carlo localization (MCL) systems. We propose a neural occupancy field to implicitly represent the scene using a neural network. With the pretrained network, we can synthesize 2D LiDAR scans for an arbitrary robot pose through volume rendering. Based on the implicit representation, we can obtain the similarity between a synthesized and actual scan as an observation model and integrate it into an MCL system to perform accurate localization. We evaluate our approach on self-recorded datasets and three publicly available ones. We show that we can accurately and efficiently localize a robot using our approach surpassing the localization performance of state-of-the-art methods. The experiments suggest that the presented implicit representation is able to predict more accurate 2D LiDAR scans leading to an improved observation model for our particle filter-based localization. The code of our approach will be available at: https://github.com/PRBonn/ir-mcl.**

***Index Terms*—Localization, Deep Learning Methods**

I. Introduction

LOCALIZING a robot on a known map is a key capability often needed by mobile robots deployed in indoor environments. For such indoor localization, we often need a map representation of the scene to establish an observation model to correct the pose estimate of a probabilistic localization algorithm, such as Monte-Carlo localization (MCL) [5]. The map representation quality and the observation model's design are critical for localization accuracy.

Recently, learning-based methods are widely used in the computer vision domain for representing the surrounding [19], [26], [27]. Among these works, Mildenhall et al. [19] propose the seminal work of neural radiance fields (NeRF), which learns an implicit function to encode the environment that can be used to generate novel views at new poses using volumetric rendering. The generated views show a high fidelity including direction-dependent illumination effects and attracted increasing interest in the computer vision community. Moreover, the implicit representation encoded by a neural network has appealing properties also relevant for robotics applications: They offer a compact representation that only needs to store the parameters of the trained neural network, and they can generalize well to locations not seen during the training. Recently, multiple works [6], [16], [23], [33], [43] have been proposed to leverage depth information to impose stronger geometric constraints. In this work, we are interested in global localization using 2D LiDAR sensors commonly employed in indoor robotics. In indoor environments, occupancy grid maps [34] are widely used to explicitly represent the environment. However, the discrete nature of occupancy maps can cause loss of scene details, which potentially leads to an inaccurate observation model of probabilistic localizations algorithms [5] and consequently inaccurate localization results.

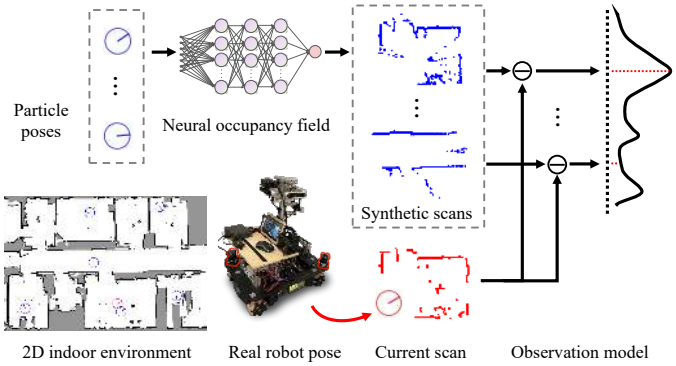

Fig. 1: Given a set of particles and a real scan from 2D LiDAR, we establish an observation model with a pretrained neural representation for accurate robot global localization.

The main contribution of this paper is the use of an implicit NeRF-based representation of the environment for MCL together with an observation model exploiting this implicit representation. It tackles the limitation of the discrete occupancy grid map and improves the localization accuracy. Our proposed method represents the 2D world using an implicit function through a neural occupancy field, named NOF. It exploits a multi-layer perceptron (MLP) to encode the 2D world. Given a location, the MLP outputs the corresponding occupancy probability. Based on that, our method then uses a ray casting-based rendering algorithm to synthesize a range scan for an input sensor pose, see Fig. 1 for an illustration. We train the NOF by comparing the rendered synthetic scan to the real sensor measurements. We use the NOF to build a novel observation model for MCL [5]. For each particle in MCL, we use our NOF to render a synthetic view and compare it to the current observation to update the particle weight. We call our global localization system implicit representation-based MCL (IR-MCL).

Manuscript received: Sep 7, 2022; Revised: Dec 2, 2022; Accepted: Jan 10, 2023. This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments.

This work has partially been funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017008 (Harmony). All authors are with the University of Bonn, Germany. Cyrill Stachniss is additionally with the Department of Engineering Science at the University of Oxford, UK, and with the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany.

Digital Object Identifier (DOI): see top of this page.

In summary, we make the following three key claims: (i) we are able to build an effective observation model based on the proposed implicit representation of the environment for 2D LiDAR-based (global) localization; (ii) we achieve state-of-the-art localization performance compared to approaches using occupancy grid maps; (iii) our approach converges fast to globally localize a robot and operates online. We support these claims by our experimental evaluation on multiple datasets.

II. Related Work

For global localization and pose tracking, Dellaert et al. [5] propose using particle filters to realize Monte-Carlo localization. MCL is still the gold standard for robot localization and often uses LiDARs [5], [8], [32], [37], cameras [5], [2], or WiFi [14]. Fox et al. [8] propose an adaptive sampling strategy for MCL to significantly improve its efficiency. Yilmaz et al. [40] propose self-adaptive MCL which is improved to make the algorithm suitable for autonomous vehicles. Such MCL methods often use a 2D LiDAR sensor and an occupancy grid map to estimate the robot's pose and are quite robust.

Recent learning-based localization algorithms also achieve high precision global localization. Lu et al. [17] propose L^3Net, which optimize a deep neural network to optimize the robot's pose using a 3D LiDAR scan with a pre-built 3D point cloud map. L^3Net achieves centimeter-level accuracy in an urban environment but needs as prior an estimate of the robot's pose. This approach is often referred to as pose tracking. Chen et al. [3] exploit a convolutional neural network to predict the overlap between a real LiDAR scan and a virtual scan from the pre-build map, which is used as an observation model in an MCL framework. Zimmerman et al. [44] combine 2D LiDAR-based localization using occupancy grid maps in the MCL framework and text spotting using an additional camera to enhance the robustness of the indoor localization.

Our proposed IR-MCL approach exploits an implicit representation of the environment to model the scene and define the observation model for localization. The classic map representation used in robot localization [4], [5], [7], [31] is the discrete occupancy grid map, which is limited by the resolution of grid cells that loses the detailed geometric information of the scene. To cope with this challenge, continuous Gaussian process grid maps [25], [41], Hilbert maps [28], and a feature-based implicit representations [42] have been proposed to represent the 2D world. Often these functions may not generalize well for describing the world precisely and therefore can limit the global localization capabilities.

In recent years, deep learning-based methods for representing the environment are widely used in the computer vision. Mildenhall et al. [19] propose a method to learn an implicit function to represent the scene by modeling a neural radiance field. It can predict realistic scene-aware views for arbitrary input poses to support many applications such as virtual/augmented reality or robot navigation using cameras [1].

In contrast to the vanilla NeRF that predicts the volume density, Xu et al. [36] propose a generative occupancy field to represent surfaces by predicting the occupancy probability of the space. Similarly, Oechsle et al. [23] propose a volume ray-tracing algorithm for the occupancy field to render surface-aware images. Based on the occupancy fields, several works exploit the depth information as a strong geometry constraint from RGB-D sensors [33], [43], or depth estimation algorithms [6], [16], [29] together with color information to train NeRF for rendering of novel depth images or even point clouds. For example, UrbanNeRF by Rematas et al. [29] train a large-scale NeRF model with high-precision 3D LiDAR to perform more realistic 3D reconstruction at city-scale. These works suggest that geometric information supports the learning of NeRFs to obtain appealing performance. Beyond the NeRF, iSDF [24] and CNM [38] learn a signed distance function (SDF) through a neural network as the map representation to trade-off between accuracy and efficiency. LASER [20] exploits the latent space for robotic visual localization.

Recently, neural implicit representations have also been used to support robot mapping and localization. For example, iNeRF by Lin et al. [39] estimates the camera pose by inverting the training process of NeRF to optimize the camera pose with a trained NeRF. There are also works that exploit an implicit scene representation in localization and mapping for mobile robots. Moreau et al. [22] propose to use NeRF to synthesize observations and enhance the mapping and localization results under limited amount of real data. They later also propose ImPosing [21], which uses the implicit representation to achieve real-time loop closing at city-scale. Adamkiewicz et al. [1] build a vision-only navigation system based on a pre-trained NeRF to forecast the measurement of the future robot state for optimizing the trajectory. Concurrent to our work, Loc-NeRF [18] was released, which exploits the implicit map representation for visual localization. To the best of our knowledge, our proposed IR-MCL system is the first work that uses an implicit neural representation as an observation model for LiDAR-based global localization in indoor environments.

III. Approach

To realize IR-MCL, we study the problem of generating 2D LiDAR scans at arbitrary sensor positions in a scene through a neural implicit representation for robot global localization. To this end, we propose a neural network to predict the occupancy probability for a given location to represent a 2D environment as detailed in Sec. III-A. Based on such estimated occupancy probabilities of samples along LiDAR rays, we render a synthetic LiDAR scan for a given pose of the robot as presented in Sec. III-B. Compared with the real measurements from 2D LiDAR during training, we optimize the weights of the network as described in Sec. III-C. After that, we use the trained network to build a novel observation model and integrate it into the MCL framework to achieve efficient global localization as presented in Sec. III-D. Fig. 2 shows an overview of our method.

A. An Implicit Representation: Neural Occupancy Field

We propose a neural network to predict the occupancy probability of an input 2D location $\boldsymbol{p} \in \mathbb{R}^2$ as the implicit scene representation, named neural occupancy field or NOF in short. Our approach uses a function F_Θ to implicitly represent a continuous 2D world. More specifically, This

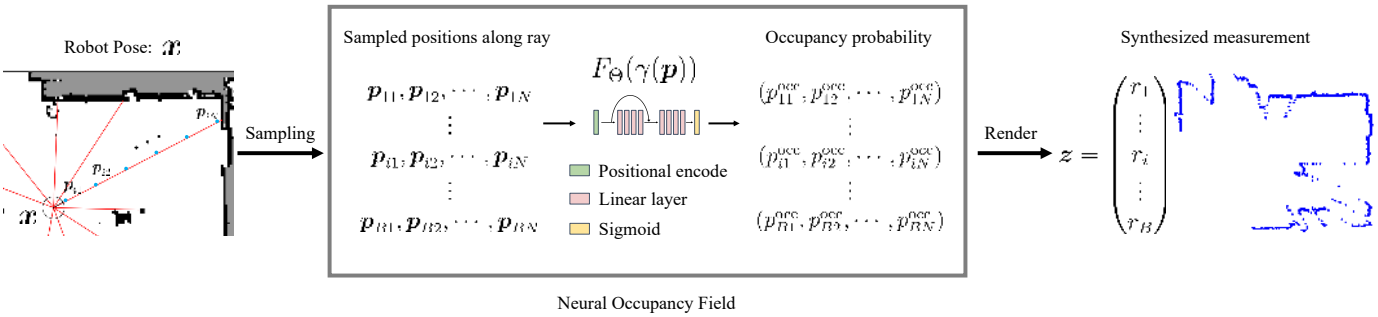

Fig. 2: Overview of our approach for rendering a synthesized measurement of LiDAR from our implicit scene representation model: Neural Occupancy Field (NOF). We uniformly sample multiple positions along each LiDAR beam, our NOF is a neural network that takes 2D position $\boldsymbol{p} = (x, y)$ as input and outputs an occupancy probability of it, we can synthesize range values with all predictions along LiDAR beams through volume rendering.

function takes a 2D location $\boldsymbol{p} = (x, y)^\top$ as input and outputs the corresponding occupancy probability p^{occ} as:

$$p^{\text{occ}} = F_\Theta(\gamma(\boldsymbol{p})). \tag{1}$$

We represent F_Θ using an MLP inspired by NeRF [19], where Θ represents the weights of the neural network. In line with NeRF, we also use positional encoding to project a 2D location to a high-dimensional space to encourage our model to encode higher frequency information of the world. We use $\gamma(\boldsymbol{p})$ with the positional encoding:

$$\begin{aligned}\gamma(\boldsymbol{p}) \;=\; & [\boldsymbol{p}, \sin(2^0\boldsymbol{p}), \cos(2^0\boldsymbol{p}), \ldots, \\ & \sin(2^{L-1}\boldsymbol{p}), \cos(2^{L-1}\boldsymbol{p})],\end{aligned} \tag{2}$$

where we use $L = 10$ in our implementation.

The network is trained such that it can map from arbitrary input 2D coordinates to the corresponding occupancy probability. To accomplish this, our MLP consisted of 8 fully-connected layers, each followed by batch normalization [13] and a ReLu activation. Additionally, we adopt and include residual connections [11] to improve the accuracy of the predictions. We apply an additional fully-connected layer followed by a sigmoid activation on the output D-dimensional feature vector generated by the MLP to obtain the occupancy probabilities $p^{\text{occ}} \in [0, 1]$.

Our network predicts an occupancy probability $p^{occ} \in [0, 1]$, which can be used for representing the 2D scene. That is different from existing neural representations, such as NeRF [19], which represents the scene geometry from the predicted volume density. Our proposed network requires no threshold adjustment to get the occupancy state (free or occupied). Thus, it generalizes well to different scenes.

B. Novel View Rendering with NOF

Based on the proposed NOF representation, we can render a novel LiDAR scan for an arbitrary 2D pose in the environment through the ray casting algorithm.

More specific, given a current 2D pose $\boldsymbol{x} = (x, y, \theta)^\top$ of a robot, we determine the origin $\boldsymbol{o} = (x, y)^\top$ and the normalized direction vector $\boldsymbol{d} = (d_1, d_2)^\top$ of each LiDAR beam. The direction vector of a ray $\boldsymbol{d}$ is calculated from the robot orientation θ and the parameters of the 2D LiDAR sensor. We uniformly sample N points $\boldsymbol{p}_i = \boldsymbol{o} + m_i\boldsymbol{d}$ along the ray, where m_i is the distance from the origin $\boldsymbol{o}$ to the sampled point $\boldsymbol{p}_i$ limited by the valid measurement range of the 2D LiDAR sensor, i.e., $m_i \in [m_{\min}, m_{\max}]$. Similar to prior work [29], we model the termination weights α_i at the endpoint $\boldsymbol{p}_i$ along the ray as:

$$\alpha_i = p_i^{\text{occ}} \prod_{j=1}^{i-1} (1 - p_j^{\text{occ}}), \tag{3}$$

where we assume that all occupancy probabilities p_i^{occ} are independent. With this, we can compute a range $r \in \mathbb{R}$ according to the termination weights of the samples $\boldsymbol{p}_i$ and their distances m_i along the ray by:

$$r = \sum_{i=1}^{N} \alpha_i m_i. \tag{4}$$

Repeating this procedure for each LiDAR beam, we can render a synthetic observation at any query location $\boldsymbol{x}$ based on our NOF, and use this scan in comparison to the real scan for the MCL observation model.

C. Training the NOF

Based on the above-introduced rendering algorithm, we can train the neural network F_Θ using recorded 2D LiDAR scans and the corresponding pose as done when building the map for traditional MCL. Each scan $\boldsymbol{z}_t$ at time t is recorded from a pose $\boldsymbol{x}_t = (x, y, \theta)_t^\top$. According to the parameters of the LiDAR sensor, each scan is comprised of B beams and each beam corresponds to a real range value $\hat{r}_i \in \mathbb{R}$ of the i^{th} beam. We use two loss functions for optimizing the weights Θ of our MLP network, a geometric loss, and an occupancy regularization.

1) Geometric Loss: We compute the geometric loss between the rendered range value r_i and the recorded range value $\hat{r}_i$ of the i^{th} beam using the L_1 loss:

$$\mathcal{L}_{geo} = \frac{1}{B} \sum_{i=1}^{B} |r_i - \hat{r}_i|. \tag{5}$$

We opt for using the L_1 instead of the L_2 loss to reduce the influence of the measurement noise of the employed 2D LiDAR sensor.

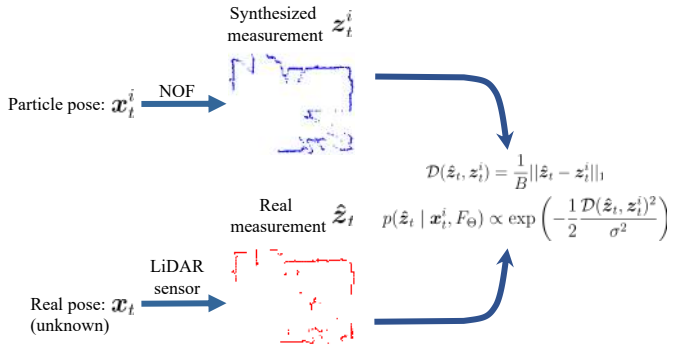

Fig. 3: The implicit representation-based observation model. We render a 2D LiDAR scan for each particle, then update the weights of the particle by comparing the synthesized measurement with the real measurement from the LiDAR sensor.

2) Occupancy Regularization: The predicted value in NOF is regarded as the occupancy value at the input location. Therefore, p^{occ} is expected to be equal to 1 for the occupied space and 0 for the free space. That means, ideally, the entropy of the prediction should be 0. Following similar work [36], we add a negative log-likelihood loss as regularization to reduce the entropy of predicted occupancy probabilities:

$$\mathcal{L}_{reg} = \frac{1}{N\,B}\sum_{i=1}^{N\,B} \log(F_\Theta(\boldsymbol{p}_i)) + \log(1 - F_\Theta(\boldsymbol{p}_i)), \tag{6}$$

where N is the number of sampled points along each beam.

The final loss function is then given by:

$$\mathcal{L} = \mathcal{L}_{geo} + \lambda \mathcal{L}_{reg}, \tag{7}$$

where λ is a hyperparameter to balance the influence of the occupancy regularization.

We train our NOF network using the Adam optimizer [15] with a batch size of 1024 for all datasets in all experiments. During rendering, we sample 256 points for each LiDAR beam, e.g. $N = 256$, and train the network for 32 epochs. The initial learning rate is 10^{-4} and decayed by 0.5 at epoch 4 and epoch 8, and weight decay is 0.001. The balancing coefficient of occupancy regularization is set to $\lambda = 10^{-5}$.

D. Implicit Representation MCL (IR-MCL)

Based on the rendered observations by our NOF network, we propose a novel observation model for MCL to achieve global localization. The global localization is formulated as a posterior probability estimation problem [34], where the objective is to estimate the belief $bel(\boldsymbol{x}_t)$ at the robot's pose $\boldsymbol{x}_t = (x, y, \theta)_t^\top$ at time t. The update of the belief $bel(\boldsymbol{x}_t)$ uses a recursive Bayes filter and is formulated as:

$$bel(\boldsymbol{x}_t) = \eta\, p(\boldsymbol{z}_t \mid \boldsymbol{x}_t, \mathcal{M})\, \overline{bel}(\boldsymbol{x}_t), \tag{8}$$

where $\overline{bel}(\boldsymbol{x}_t)$ is the predicted belief of the robot pose according to the motion controls and the last pose $\boldsymbol{x}_{t-1}$, which is also called motion model of the robot. The $p(\boldsymbol{z}_t \mid \boldsymbol{x}_t, \mathcal{M})$ is the likelihood of the sensor measurement $\boldsymbol{z}_t$ while the robot state is $\boldsymbol{x}_t$ in the map $\mathcal{M}$. It is also regarded as the observation model for correcting the estimate of motion model. The η is a normalization factor.

MCL exploits a particle filter to approximate the update of posterior $bel(\boldsymbol{x}_t)$ by a set of random samples drawn from the posterior. These random samples, so-called particles, denoted as $\mathcal{X}_t = \{(\boldsymbol{x}_t^1, w_t^1), (\boldsymbol{x}_t^2, w_t^2), \ldots, (\boldsymbol{x}_t^M, w_t^M)\}$, where w^i is the weight of the pose $\boldsymbol{x}^i$, and M is number of particles. After updating, the particles are re-sampled according to the particles' importance weights. Repeating this process, the particles eventually converge to a small region around the real pose.

In this work, we exploit our NOF model to implicitly represent the environment $\mathcal{M}$ to generate scans used in our observation model for MCL to achieve global localization. More specific, our IR-MCL treats each particle as a hypothesized robot pose at time t, e.g. $\boldsymbol{x}_t^i = (x^i, y^i, \theta^i)_t^\top$. We render an observation $\boldsymbol{z}_t^i$ at each particle location as described in Sec. III-B, and compare it with the real measurement $\hat{\boldsymbol{z}}_t$ obtained by the 2D LiDAR sensor, which is shown in Fig. 3. Following [4], we approximated the likelihood $p(\hat{\boldsymbol{z}}_t \mid \boldsymbol{x}_t^i, F_\Theta)$ of the i-th particles through a Gaussian distribution:

$$p(\hat{\boldsymbol{z}}_t \mid \boldsymbol{x}_t^i, F_\Theta) \propto \exp\left(-\frac{1}{2}\frac{\mathcal{D}(\hat{\boldsymbol{z}}_t, \boldsymbol{z}_t^i)^2}{\sigma^2}\right), \tag{9}$$

where $\mathcal{D}(\hat{\boldsymbol{z}}_t, \boldsymbol{z}_t^i)$ is the difference between the measurement $\hat{\boldsymbol{z}}_t$ and $\boldsymbol{z}_t^i$. We use the L_1 distance to calculate $\mathcal{D}$, i.e., $\mathcal{D}(\hat{\boldsymbol{z}}_t, \boldsymbol{z}_t^i) = \frac{1}{B}\|\hat{\boldsymbol{z}}_t - \boldsymbol{z}_t^i\|_1$. It is robust to the noise of the measurements and easy to use while also maintaining high efficiency. By comparing the current real-sensor measurement with the synthetic observations rendered at all particle locations, we update the likelihood $p(\hat{\boldsymbol{z}}_t \mid \boldsymbol{x}_t, F_\Theta)$.

To accelerate the runtime of our IR-MCL while using a large number of particles, e.g., $M = 100,000$, we build a predefined 2D grid to store the predicted probabilities for accelerating the rendering similar to [12], and call it neural occupancy grids (NOG). The NOG will cover the whole space of the current scene. During localization, we use the nearest neighbor cell of each sampling point along the ray in the NOG as the occupancy probability at this point. By exploiting NOG, our IR-MCL system achieves real-time performance even with a large number of particles.

IV. Experimental Evaluation

The main focus of this work is an implicit representation-based Monte-Carlo localization system for the global localization of a robot. We present our experiments to show the capabilities of our method and support our key claims, which are: (i) we are able to build an accurate observation model based on the proposed NOF for 2D LiDAR-based global localization, (ii) we achieve state-of-the-art localization performance compared to approaches using occupancy grid maps, (iii) our approach converges fast to globally localize a robot and operates online.

A. Experimental Setup

Datasets. We evaluate our method and compare it with the state-of-the-art methods in multiple datasets including, three typical publicly available datasets: Freiburg Building 079 (shortly Fr079) dataset, Intel Lab dataset, MIT CSAIL

Sequence	Method	Memory	Location RMSE (cm) ↓	< 5cm Pct. ↑	< 10cm Pct. ↑	< 20cm Pct. ↑	Yaw RMSE (degree) ↓	$< 0.5°$ Pct. ↑	$< 1°$ Pct. ↑	$< 2°$ Pct. ↑
Seq 1	AMCL	4 MB	11.57	24.44%	58.89%	92.22%	1.80	21.11%	44.44%	81.11%
	NMCL		-	17.36%	32.98%	81.71%	7.14	22.16%	39.12%	67.16%
	SRRG-Loc		6.36	49.11%	92.10%	99.85%	1.08	**47.25%**	**75.45%**	94.00%
	HMCL	0.01 MB	13.44	18.46%	32.98%	81.71%	3.33	19.57%	38.55%	67.50%
	IR-MCL	1.96 MB	**5.13**	**63.15%**	**97.27%**	**100.00%**	**1.05**	47.12%	74.59%	**94.24%**
Seq 2	AMCL	4 MB	10.65	17.11%	52.63%	98.68%	1.09	28.95%	55.26%	94.74%
	NMCL		23.52	19.78%	40.19%	74.96%	4.51	19.78%	40.96%	65.38%
	SRRG-Loc		8.83	25.79%	69.09%	**100.00%**	1.43	27.82%	53.54%	82.28%
	HMCL	0.01 MB	-	0.00%	0.00%	0.00%	-	0.00%	0.00%	0.00%
	IR-MCL	1.96 MB	**5.53**	**62.20%**	**92.85%**	**100.00%**	**0.81**	**48.88%**	**82.15%**	**98.16%**
Seq 3	AMCL	4 MB	-	30.77%	71.79%	84.62%	-	15.38%	23.08%	61.54%
	NMCL		-	0.00%	0.00%	0.00%	-	0.64%	0.96%	1.61%
	SRRG-Loc		50.36	20.91%	39.38%	77.98%	2.86	17.80%	22.25%	43.38%
	HMCL	0.01 MB	-	0.00%	0.00%	0.00%	-	0.00%	0.00%	0.00%
	IR-MCL	1.96 MB	**4.59**	**80.42%**	**98.33%**	**100.00%**	**0.65**	**60.18%**	**85.54%**	**100.00%**
Seq 4	AMCL	4 MB	-	20.55%	54.79%	83.56%	9.57	24.66%	42.47%	68.49%
	NMCL		-	0.00%	0.00%	0.00%	-	0.35%	0.70%	2.29%
	SRRG-Loc		**11.02**	16.17%	51.73%	**96.04%**	**1.15**	**45.79%**	68.40%	**89.44%**
	HMCL	0.01 MB	22.70	0.50%	6.77%	61.1%	4.42	13.28%	27.39%	52.23%
	IR-MCL	1.96 MB	11.54	**38.78%**	**67.82%**	92.74%	1.56	37.94%	**68.73%**	84.74%
Seq 5	AMCL	4 MB	-	15.87%	58.73%	92.06%	-	20.63%	53.97%	85.71%
	NMCL		47.15	7.78%	45.91%	84.83%	3.82	22.95%	39.12%	52.10%
	SRRG-Loc		-	34.97%	69.85%	94.49%	1.87	**38.24%**	**73.30%**	**90.78%**
	HMCL	0.01 MB	23.97	1.46%	9.99%	57.62%	5.76	9.30%	24.46%	49.87%
	IR-MCL	1.96 MB	**6.33**	**48.15%**	**90.27%**	**100.00%**	**1.47**	37.98%	60.03%	81.22%

TABLE I: Quantitative results on IPBLab dataset. We compare with various MCL-based global localization algorithms. We report the absolute pose error metrics in location and direction, the '-' means global localization failed in the sequence. We only report the accuracy for success cases. Alongside, we also report the ratio of frames with location and yaw angle errors less than the given threshold.

Lab (shortly MIT), and a self-recorded dataset, called IPBLab dataset. The three publicly datasets only contain one sequence of an indoor scenario, therefore, we split each sequence into three subsets for training, validation, and testing. The Fr079 contains 3448 frames for training, 384 and 959 frames for validation and testing respectively. The Intel Lab dataset contains 655 frames for training, 73 and 182 frames for validation and testing respectively. The MIT dataset contains 291 frames for training, 33 and 82 frames for validation and testing respectively. The IPBLab dataset was collected in our building at the University of Bonn using a Kuka YouBot platform equipped with several sensors, including a Hokuyo UTM-30LX LiDAR sensor and an up-facing camera. The up-facing camera is used for determining close to ground truth poses of the robot through localizing densely placed AprilTags on the ceiling that have been measured with a high-precision terrestrial laser scanner [44]. Additionally, the ground truth poses are optimized by aligning the scans with a highly precise dense point cloud map generated by a Faro terrestrial laser scanner and human-supervised scan matching. For training our NOF model, we collect a long sequence including $31,608$ frames as the training set. It consists of several indoor scenes: office, corridor and kitchen, and covers the whole scene. We additionally collect five shorter sequences for evaluating global localization. Each sequence traverses a sub-region of the scene, and the average length of the testing sequences is 1419 frames.

Baselines. We compare our method with three existing 2D LiDAR-based global localization algorithms: First, AMCL [8] as shipped with ROS[1], a widely used highly efficient MCL algorithm; Second, the MCL approach by Zimmerman et al. [44], which we call NMCL; Third, the approach by Sapienza Robust Robotics Group (SRRG), called SRRG-Localizer [9] shortly SRRG-Loc, which is a sophisticated MCL implemented by the team led by Giorgio Grisetti. We additionally re-implemented the HMCL approach using an observation model [35] with a continuous Bayesian Hilbert map [30]. Note that NMCL uses no text spotting to support the localization in our experiments. We kept the default parameters for all baseline methods and uses the same number of particles for all approaches for a fair comparison.

[1]http://wiki.ros.org/amcl

B. Global Localization Performance

The first experiment is designed to support the claim that exploits our devised implicit representation-based observation model, our IR-MCL achieves state-of-the-art accuracy in the global localization of a robot.

Our MCL system includes two stages: initialization and pose tracking. At the initialization stage, we uniformly sample $M = 100,000$ particles in the whole space at the beginning to achieve global localization without any priors. At the pose tracking stage, we reduce the number of particles to $M = 5,000$. HMCL uses the same number of particles as our method. For AMCL, the range of particle numbers is decreased from $100,000$ to $5,000$. We fixed particles number to $100,000$ for NMCL and SRRG-Loc, because they do not provide a mechanism for adjusting the number of particles. For baseline methods, we build an occupancy grid map or Bayesian Hilbert map by using the data from the training set as the scene representation. The size of the grid map is $50\,\mathrm{m} \times 50\,\mathrm{m}$ with $5\,\mathrm{cm}$ grid resolution. For all approaches, we use the same motion model and the odometry reading as control commands. Thus, the main difference between

#Particles	10,000	5,000	500	50
AMCL	10.50/1.52	10.45/1.52	10.65/1.74	18.67/3.44
NMCL	13.69/2.37	13.52/2.27	14.06/2.46	16.84/3.30
SRRG-Loc	8.52/1.48	8.48/1.19	8.44/1.54	**9.71/2.24**
HMCL	21.42/4.90	21.40/4.89	20.90/4.84	20.84/4.82
IR-MCL	**6.96/1.14**	**6.85/1.14**	**6.87/1.19**	12.78/2.39

TABLE II: Ablation study on number of particles in pose tracking stage. We report average APE in Location RMSE (cm)/Yaw RMSE (degree) format.

the different approaches lies in the observation model of the different MCL implementations.

Regarding the evaluation metrics, we calculate the absolute pose error (APE) between the estimated robot poses and ground truth poses in five testing sets of the IPBLab dataset. We show both the location error and yaw angle error in terms of the root mean square error (RMSE) of the location $(x, y)^\top$ and the direction θ w.r.t. the ground truth. For a fair comparison, we leave out the first $20\,\mathrm{s}$ as the initialization stage for the MCL and only calculate the localization error when the method converged, i.e., the initialization stage is excluded for calculating the APE and RMSE. The localization is regarded as failed if the method cannot converge in $20\,\mathrm{s}$, i.e., the location RMSE or yaw RMSE larger than a threshold. We use $50\,\mathrm{cm}$ for location RMSE and $5°$ for Yaw RMSE as a threshold in the experiments.

Besides, we also report the ratio of frames with location and yaw angle errors less than confident thresholds to evaluating the precision of localization results at different tolerance. The thresholds are $5\,\mathrm{cm}$, $10\,\mathrm{cm}$ and $20\,\mathrm{cm}$ for the location error, respectively, and $0.5°$, $1°$, and $2°$ for the yaw angle error. Tab. I shows the quantitative results of localization performance.

The experimental results show that our method outperforms the baseline methods in both location and yaw angle accuracy, and that it improves the location accuracy. As argued before, the main difference between the different methods lies in the observation models, where our method can directly generate a rendered scan from an implicit representation of the scene and does not rely on the discrete occupancy grid map. In addition, the memory consumption of our NOF representation is only half of the occupancy grid map. The Hilbert map has a lower memory footprint, but it sacrifices localization accuracy. The results show that our method has a good trade-off between performance and memory cost.

Fig. 4 shows the qualitative results on sequence 1 and 5 of the IPBLab dataset. We plot the trajectory of each method after the initialization stage, where color indicates the translation error. We can see that the proposed IR-MCL is much more accurate for global localization as the colors are always in the lower range of the spectrum. Furthermore, the figures show that the SRRG-Loc also performs well after convergence, but needs more time in the initialization stage.

Tab. II shows an ablation study on different numbers of particles for pose tracking. The particles are initialized by adding Gaussian noise to the ground truth pose of the first frame. The numbers of particles are the same for all methods. We report the average APE on five sequences of the IPBLab dataset in "Location RMSE (cm)/Yaw RMSE (°)" format. The experimental results show that our method outperforms all

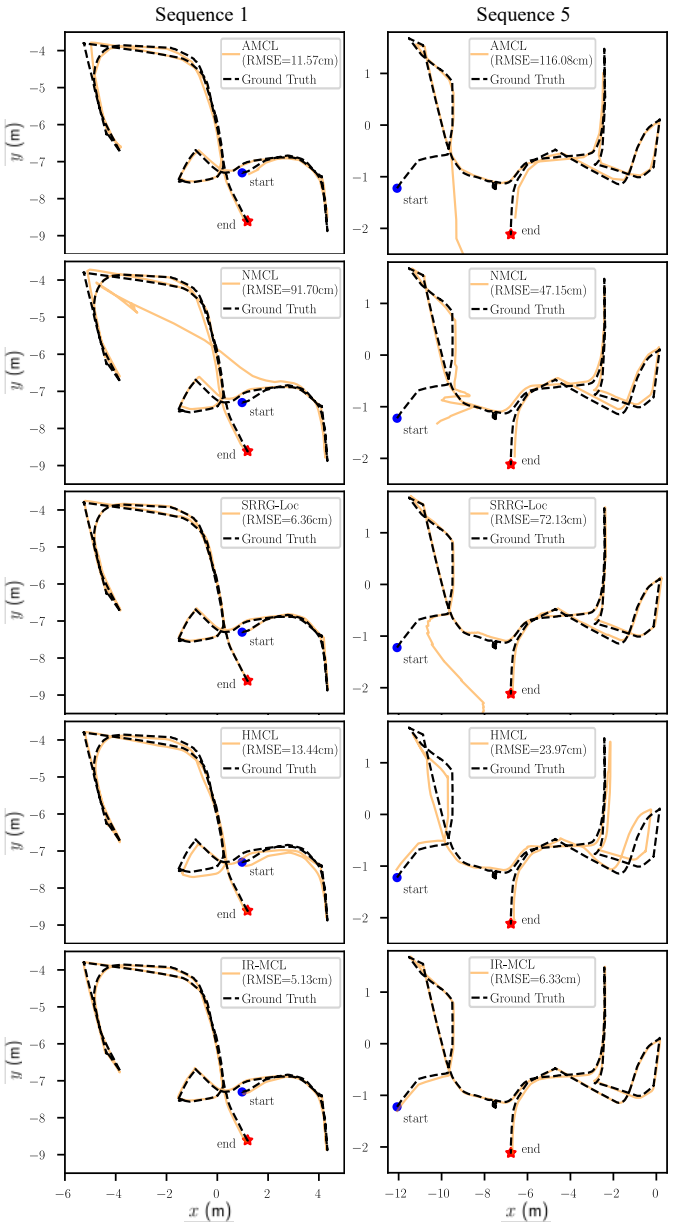

Fig. 4: Qualitative global localization results on the sequence 1 and 5 of IPBLab dataset. The dash line depicts ground-truth trajectory, and orange line is the estimated trajectory of the different approaches after convergence. RMSE inside the legend is the location RMSE. Our approach leads to more consistent and more accurate trajectories while achieving faster convergence.

baselines in most cases. SRRG-Loc slightly outperforms our method in the extreme case of only 50 particles. Moreover, our method stays accurate when particles number change from $5,000$ to 500. Therefore, particles number can be selected according to computing resources in practice. In this paper, we choose $5,000$ particles to ensure the robustness of our systems after satisfying online operation.

Additionally, the experimental results show that the NMCL and SRRG-Loc work well with fewer particle numbers if provided a good initial guess. However, we still fixed the particle number to $100,000$ in the global localization experiment. It ensures the procedure of these algorithms is not changed for the fairness of the experiment.

Dataset	Method	Avg Error (m) ↓	Acc ↑	CD (m) ↓	F ↑
Fr079	Ray-casting	0.33	88.41%	0.17	0.97
	NOF (ours)	**0.20**	**92.16%**	**0.16**	**0.98**
Intel Lab	Ray-casting	0.27	91.62%	**0.19**	**0.97**
	NOF (ours)	**0.18**	**92.54%**	**0.19**	**0.97**
MIT	Ray-casting	0.98	80.16%	0.57	0.92
	NOF (ours)	**0.45**	**81.33%**	**0.37**	**0.93**

TABLE III: Quantitative results for the observation model. We compare the rendered scans with the ground truth measurement from the 2D LiDAR. We compare our method (NOF) with the ray-casting methods rendering a scan from an occupancy grid map using Bresenham's algorithm (Ray-casting).

C. Evaluation of the Observation Model Computed on the Implicit Representation

The second experiment is presented to back up the claim that our proposed observation model for a 2D LiDAR sensor based on our proposed implicit representation is more accurate than existing models. In this experiment, we directly compare rendered scans from our NOF model given poses with the real LiDAR scans. We take the traditional ray-casting observation model of the beam-end model [34], as a baseline, which renders scans using Bresenham's algorithm using an occupancy grid map built by GMapping [10].

We evaluate the performance in Freiburg building 079 dataset, Intel Lab dataset, and MIT CSAIL Lab, which shows that our method is robust in different scenarios. Because these datasets only contain one sequence, these datasets are not suitable for evaluating global localization. Here, we split the sequence into subsets for training and evaluation of observation models.

Tab. III shows the quantitative evaluation of the observation models. We use similar error metrics introduced by Rematas et al. [29]. We compare synthesized scans $\boldsymbol{z}$ with the ground truth scans $\hat{\boldsymbol{z}}$ and report the average absolute error of the measurements. This metric is the same as the distance function $\mathcal{D}$ used in our observation model, see Eq. (9). Therefore, it directly reflects the errors brought to the localization system. We also report the accuracy (Acc) as the ratio of the LiDAR beams with a range error smaller than $0.5\,\mathrm{m}$ compared to the real scan measurements. Given a LiDAR pose, we can determine the LiDAR beams' origin $\boldsymbol{o}$ and direction $\boldsymbol{d}$. The corresponding 2D point cloud of a scan $\boldsymbol{z}$ is $\boldsymbol{p}_i = \boldsymbol{o} + z_i \boldsymbol{d}$. We do the same operation for the real scans $\hat{\boldsymbol{z}}$ to get the ground-truth. We compute the Chamfer Distance (CD) and F - score (F) using a threshold of $0.5\,\mathrm{m}$ between the rendered and the ground-truth point clouds.

As shown in Tab. III, our model generally achieves better performance in all metrics in all datasets. Our NOF model reduces the average absolute error than the ray-casting method. The reason is that our method synthesizes more accurate scans compared with the ray-casting method even if there are only training scans with noisy poses available from a SLAM algorithm. Note that our method significantly improves the accuracy on the MIT dataset by a factor of 2 considering average absolute error. In this case, it seems that the 291 frames are not sufficient to build an occupancy grid map to precisely represent the scene. But our implicit representation can make reasonable predictions for some places, which are unseen in the training set. It also supports that our methods have good generalization capabilities for small datasets. Besides, our method only gets minor improvement in F-score, since it is the harmonic mean of accuracy and completeness. The results show that our method can reconstruct the complete scene as the occupancy grid map, but is more accurate than it.

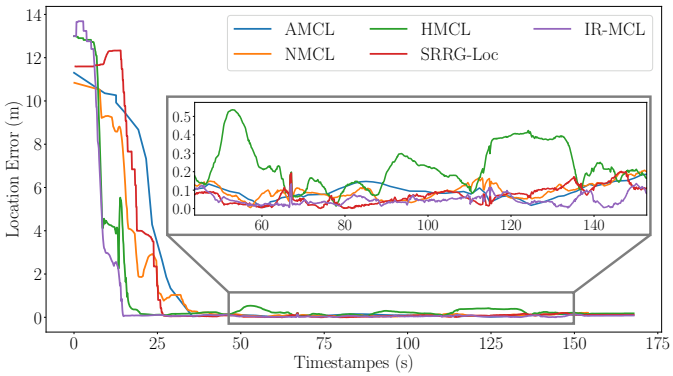

Fig. 5: The location error for each frame of sequence 5 of the IPBLab dataset. Our method converges faster than baseline methods, and the localization errors are most of the time lower than the baselines after convergence.

D. Runtime

The final experiment supports our claim that our IR-MCL achieves fast convergence for global localization and can operate online. We compare our method with all baseline methods on sequence 5 of the IPBLab dataset and we report the location RMSE of every frame of the sequence. As shown in Fig. 5, our method converges faster than other baseline methods and achieves higher accuracy after convergence.

We test the runtime of our IR-MCL in initialization stage ($100,000$ particles) and pose tracking stage ($5,000$ particles). On a PC with 10 CPU Cores at 3.7 GHz and 64 GB memory, and an NVIDIA Quadro RTX 5000 GPU, we achieve an average frame rate of $1.2\,\mathrm{Hz}$ during initialization stage, and $27\,\mathrm{Hz}$ after convergence, which supports our claim that computations can be executed fast and in an online fashion.

In summary, our evaluation suggests that we can build an accurate observation model, which provides competitive global localization performance for a mobile robot. At the same time, our method is fast enough for online processing.

V. Conclusion

In this paper, we presented a novel implicit representation-based online localization approach using a 2D LiDAR. Our method exploits a neural network-based scene representation to build an accurate observation model. This allows us to successfully localize a mobile platform in a given environment, and outperform existing gold standard MCL in terms of localization accuracy. We implemented and evaluated our approach on different datasets and provided comparisons to other existing techniques supporting all claims made in this paper. The experiments suggest that our approach achieves reliable and accurate global localization while operating online at the sensor frame rate after convergence. An avenue for

future work is to relax the requirement for accurate poses to learn the implicit representation and perform pose estimation at the same time.

REFERENCES

[1] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters (RA-L)*, 7(2):4606–4613, 2022.

[2] M. Bennewitz, C. Stachniss, W. Burgard, and S. Behnke. Metric Localization with Scale-Invariant Visual Features using a Single Perspective Camera. In H. Christiensen, editor, *European Robotics Symposium 2006*, volume 22 of *STAR Springer Tracts in Advanced Robotics*, pages 143–157. Springer Verlag, 2006.

[3] X. Chen, T. Läbe, L. Nardi, J. Behley, and C. Stachniss. Learning an Overlap-based Observation Model for 3D LiDAR Localization. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[4] X. Chen, I. Vizzo, T. Läbe, J. Behley, and C. Stachniss. Range Image-based LiDAR Localization for Autonomous Vehicles. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[5] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo Localization for Mobile Robots. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 1999.

[6] K. Deng, A. Liu, J.Y. Zhu, and D. Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[7] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In *Proc. of the National Conf. on Artificial Intelligence (AAAI)*, 1999.

[8] D. Fox. Kld-sampling: Adaptive particle filters. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2001.

[9] G. Grisetti. srrg-localizer2d (1.6.0). https://gitlab.com/srrg-software/srrg_localizer2d, 2018.

[10] G. Grisetti, C. Stachniss, and W. Burgard. Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters. *IEEE Trans. on Robotics (TRO)*, 23(1):34–46, 2007.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] P. Hedman, P.P. Srinivasan, B. Mildenhall, J.T. Barron, and P. Debevec. Baking neural radiance fields for real-time view synthesis. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[13] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. of the Intl. Conf. on Machine Learning (ICML)*, 2015.

[14] S. Ito, F. Endres, M. Kuderer, G. Tipaldi, C. Stachniss, and W. Burgard. W-RGB-D: Floor-Plan-Based Indoor Global Localization Using a Depth Camera and WiFi. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2014.

[15] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2015.

[16] J. Li, Z. Feng, Q. She, H. Ding, C. Wang, and G.H. Lee. Mine: Towards continuous depth mpi with nerf for novel view synthesis. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[17] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song. L3-Net: Towards Learning Based LiDAR Localization for Autonomous Driving. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] D. Maggio, M. Abate, J. Shi, C. Mario, and L. Carlone. Loc-nerf: Monte carlo localization using neural radiance fields. *arXiv preprint arXiv:2209.09050*, 2022.

[19] 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 *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[20] Z. Min, N. Khosravan, Z. Bessinger, M. Narayana, S.B. Kang, E. Dunn, and I. Boyadzhiev. Laser: Latent space rendering for 2d visual localization. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[21] A. Moreau, T. Gilles, N. Piasco, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle. Imposing: Implicit pose encoding for efficient camera pose estimation. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2023.

[22] A. Moreau, N. Piasco, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *Proc. of the Conf. on Robot Learning (CoRL)*, 2021.

[23] M. Oechsle, S. Peng, and A. Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[24] J. Ortiz, A. Clegg, J. Dong, E. Sucar, D. Novotny, M. Zollhoefer, and M. Mukadam. isdf: Real-time neural signed distance fields for robot perception. In *Proc. of Robotics: Science and Systems (RSS)*, 2022.

[25] S.T. O'Callaghan and F.T. Ramos. Gaussian process occupancy maps. *Intl. Journal of Robotics Research (IJRR)*, 31(1):42–62, 2012.

[26] J.J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[27] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger. Convolutional occupancy networks. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[28] F. Ramos and L. Ott. Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent. *Intl. Journal of Robotics Research (IJRR)*, 35(14):1717–1730, 2016.

[29] K. Rematas, A. Liu, P.P. Srinivasan, J.T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari. Urban radiance fields. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[30] R. Senanayake and F. Ramos. Bayesian hilbert maps for dynamic continuous occupancy mapping. In *Proc. of the Conf. on Robot Learning (CoRL)*, 2017.

[31] C. Stachniss and W. Burgard. Exploring Unknown Environments with Mobile Robots using Coverage Maps. In *Proc. of the Intl. Conf. on Artificial Intelligence (IJCAI)*, pages 1127–1132, Acapulco, Mexico, 2003.

[32] C. Stachniss and W. Burgard. Mobile Robot Mapping and Localization in Non-Static Environments. In *Proc. of the National Conf. on Artificial Intelligence (AAAI)*, 2005.

[33] E. Sucar, S. Liu, J. Ortiz, and A.J. Davison. imap: Implicit mapping and positioning in real-time. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[34] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[35] G. Vallicrosa and P. Ridao. H-slam: Rao-blackwellized particle filter slam using hilbert maps. *Sensors*, 18(5):1386, 2018.

[36] X. Xu, X. Pan, D. Lin, and B. Dai. Generative occupancy fields for 3d surface-aware image synthesis. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2021.

[37] F. Yan, O. Vysotska, and C. Stachniss. Global Localization on OpenStreetMap Using 4-bit Semantic Descriptors. In *Proc. of the Europ. Conf. on Mobile Robotics (ECMR)*, 2019.

[38] Z. Yan, Y. Tian, X. Shi, P. Guo, P. Wang, and H. Zha. Continual neural mapping: Learning an implicit scene representation from sequential observations. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[39] 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 *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2021.

[40] A. Yilmaz and H. Temeltas. Self-adaptive monte carlo method for indoor localization of smart agvs using lidar data. *Journal on Robotics and Autonomous Systems (RAS)*, 122:103285, 2019.

[41] Y. Yuan, H. Kuang, and S. Schwertfeger. Fast gaussian process occupancy maps. In *Proc. of the Intl. Conf. on Control, Automation, Robotics and Vision (ICARCV)*, 2018.

[42] J. Zhao, L. Zhao, S. Huang, and Y. Wang. 2d laser slam with general features represented by implicit functions. *IEEE Robotics and Automation Letters (RA-L)*, 5(3):4329–4336, 2020.

[43] 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 *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[44] N. Zimmerman, L. Wiesmann, T. Guadagnino, T. Läbe, J. Behley, and C. Stachniss. Robust onboard localization in changing environments exploiting text spotting. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2022.