

VIRUS-NeRF - Vision, InfraRed and UltraSonic based Neural Radiance Fields

Nicolaj Schmid^{1,*}, Cornelius von Einem^{1,*}, Cesar Cadena¹, Roland Siegwart¹, Lorenz Hruby^{2,*} and Florian Tschopp^{3,*}

Abstract— Autonomous mobile robots are an increasingly integral part of modern factory and warehouse operations. Obstacle detection, avoidance and path planning are critical safety-relevant tasks, which are often solved using expensive LiDAR sensors and depth cameras. We propose to use cost-effective low-resolution ranging sensors, such as ultrasonic and infrared time-of-flight sensors by developing *VIRUS-NeRF - Vision, InfraRed, and UltraSonic based Neural Radiance Fields*.

Building upon Instant Neural Graphics Primitives with a Multiresolution Hash Encoding (Instant-NGP), *VIRUS-NeRF* incorporates depth measurements from ultrasonic and infrared sensors and utilizes them to update the occupancy grid used for ray marching. Experimental evaluation in 2D demonstrates that *VIRUS-NeRF* achieves comparable mapping performance to LiDAR point clouds regarding coverage. Notably, in small environments, its accuracy aligns with that of LiDAR measurements, while in larger ones, it is bounded by the utilized ultrasonic sensors. An in-depth ablation study reveals that adding ultrasonic and infrared sensors is highly effective when dealing with sparse data and low view variation. Further, the proposed occupancy grid of *VIRUS-NeRF* improves the mapping capabilities and increases the training speed by 46% compared to Instant-NGP. Overall, *VIRUS-NeRF* presents a promising approach for cost-effective local mapping in mobile robotics, with potential applications in safety and navigation tasks. The code can be found at https://github.com/ethz-asl/virus_nerf.

I. INTRODUCTION

As automation advances, the demand for mobile robots is on the rise. In the realm of factories and warehouses, Autonomous Mobile Robots (AMRs) exhibit remarkable flexibility and perform tasks with greater intelligence compared to traditional Automated Guided Vehicles (AGVs). Typically, AMRs operate in semi-dynamic environments alongside human workers. The robot must effectively perceive its surroundings to facilitate smooth navigation and obstacle avoidance. Operating within the same workspace as humans requires a safe mapping algorithm, focusing particularly on the proximity of the robot (e.g. see Fig. 1). In this work, perception is studied in the simplified case of static environments and the mapping is evaluated in 2D space.

In industry, safety-critical tasks for AMRs, such as collision avoidance, often rely on costly sensors, such as 3D Light Detection And Ranging (LiDAR) sensors. Typically, local

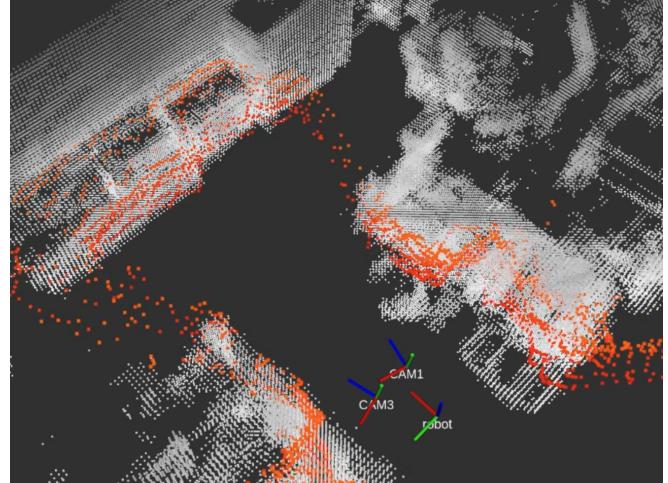


Fig. 1: *Office*: global map in white and *VIRUS-NeRF* predictions in orange. The axes are the reference frames of the LiDAR and of the two cameras. For this visualization in 3D, *VIRUS-NeRF* is inferred at multiple heights.

and instantaneous measurements are utilized being more robust and allowing faster scanning cycles than when using global mapping. This study seeks to employ cost-efficient sensors while maintaining local mapping capabilities similar to more expensive setups. For example, Ultrasonic Sensors (USSs) are widely adapted low-cost ranging sensors in the car industry [1] and many of the early mapping algorithms in mobile robotics utilize them [2]–[4]. However, their low angular resolution combined with traditional fusion methods lead to sub-optimal outcomes. Similarly, other cheap time-of-flight sensors, e.g. Infrared Sensors (IRSs), result in limited mapping performances due to their sparse measurements and reduced range. Contemporary approaches frequently rely on more advanced sensors like LiDARs or depth cameras. Despite their impressive performance, these setups incur substantial costs, making them less accessible for widespread adoption. Recent advancements in stereo vision [5], [6] and Monocular Depth Estimation (MDE) [7]–[9] show significant progress in camera-based mapping solutions. However, the lack of robustness, high computational requirements and the intrinsic scale ambiguity of monocular cameras remain open research challenges. Sensor fusion may address some of these drawbacks, e.g. depth completion [10]–[12]. Nevertheless, most of these techniques return to using costly LiDAR or RGB-D sensors.

*Authors contributed equally to this work

¹Authors are members of the Autonomous Systems Lab, ETH Zurich, Switzerland; {firstname.lastname}@mavt.ethz.ch

²Author is with Filics GmbH, Munich, Germany; hruby@filics.eu

³Author is with Voliro AG, Zurich, Switzerland; fts@voliro.com

This work was supported by the ETH Mobility Initiative under the project *LROD-ADAS*.

This study suggests using the framework of Neural Radiance Fields (NeRFs) to fuse color images with range measurements and to learn an implicit scene representation. In the context of mobile robotics, NeRFs incur two major drawbacks: First, they converge slowly, which makes them problematic for real-time usage, and second, they require dense data with a high view variation of the environment [13], [14]. More recent works address the convergence speed by proposing various improvements [11], [13]. Most mobile robots, especially in warehouse and factory environments, are constrained to move along pre-defined trajectories on a 2D plane, severely limiting their viewpoint variability. Therefore, we developed *VIRUS-NeRF* - a *Vision, InfraRed and UltraSonic based Neural Radiance Fields*. *VIRUS-NeRF* is based on Instant Neural Graphics Primitives with a Multiresolution Hash Encoding (Instant-NGP). Similar to other works using LiDARs [15], [16] or depth cameras [17], [18], *VIRUS-NeRF* complements the image-based training by depth measurements. Notably, *VIRUS-NeRF* is the first NeRF algorithm utilizing low-cost depth sensors, i.e. USSs and IRSs. The contributions of this work are the following:

- A novel real-time, NeRF-based sensor fusion method for integrating low-cost and low-resolution USSs and IRSs with RGB cameras. The low-cost sensors provide depth supervision, to the normally purely image-based training of NeRFs, resulting in more accurate and robust reconstructions of the environment.
- Improvements to the occupancy grid of Instant-NGP using a probabilistic Bayesian formulation, which permits direct occupancy updates by depth measurements.
- An evaluation of *VIRUS-NeRF*'s mapping accuracy and coverage on real-world datasets and a direct comparison to instantaneous scans of LiDARs, USSs and IRSs.
- An in-depth ablation study comparing *VIRUS-NeRF* to Instant-NGP, analyzing the contribution of the depth supervision and the improved occupancy grid separately, and studying different isolated sensor modalities.

II. RELATED WORK

A. Occupancy Grids

One of the first mapping techniques is the occupancy grid, which is also utilized in Instant-NGP, the base model of *VIRUS-NeRF* (see section II-C). Occupancy grids are probabilistic maps describing the occupancy state of a discretized environment [2]. Many of the early publications explicitly use low-cost sensors, e.g. USSs [2], [19]–[21]. The most established implementation is based on the *Bayesian Updating Rule* to integrate new measurements into the map [3]. *VIRUS-NeRF* makes use of the Bayesian occupancy grid (see section III-B) which considers consecutive measurements and neighboring cells independently to reduce the updating complexity. These strong assumptions may be dropped by addressing the correlation between successive samples [20] or between neighboring cells [21]. Multiple works extend occupancy grids to handle dynamic objects using *Bayesian Occupancy Filters* [22], *Particle Filters* [23] or *Markov Chains* [24], [25].

B. Depth Completion

Monocular cameras provide extremely rich information and are available for moderate prices, therefore being extensively used in mobile robotics. *VIRUS-NeRF* employs monocular cameras in the framework of NeRFs. However, there exist some viable methods based on deep learning: For example, MDE predicts the pixel-wise depth by using color images [8]. RGB images have no inherent depth information and scale ambiguity limits the accuracy of MDE. Additionally, most MDE algorithms do not estimate the uncertainty of their predictions [26], which makes sensor fusion difficult. Contrary, depth completion tries to leverage the early fusion of RGB images with depth measurements. Most approaches are based on deep learning and complement the camera with a LiDAR sensor [11]. As an alternative to LiDAR point clouds, radar scans can be used for depth completion [27]–[29]. These models are relatively new because of the availability of large-scale open-source datasets containing radar data [30] and the improved resolution of radar sensors in the past few years [28]. Existing methods developed for LiDAR do not transfer well to radars, due to the sparsity and high noise levels of the data, as well as the smaller vertical Field of View (FoV) [27], [29]. To the best of our knowledge, there is no published work on utilizing USSs or IRSs for depth completion and there does not exist any large-scale open-source dataset containing such low-cost sensors.

C. Neural Radiance Fields

In 2020, Mildenhall *et al.* introduce NeRFs [31]. NeRFs use Multi-Layer Perceptrons (MLP) to learn the geometry and the lighting of one particular three-dimensional static scene. During inference, the model can render a new view from any position and viewing direction. The required data is composed of images and their respective camera poses.

The original NeRF implementation [31] has some important limitations: The training of one scene takes a long time (up to ~ 12 h on a GPU [13]). In addition, NeRFs are designed for small scenes, i.e. single objects or small rooms, and struggle with unbounded environments [16]. Dense data having a high view variation is required for training [13], [14] and surfaces can be rugged due to a lack of geometrical constraints [32]. Many of these drawbacks are addressed in subsequent publications, as presented below.

Instant-NGP reduces the training time from several hours to a few minutes while achieving a similar accuracy [33]. Instead of a sinusoidal encoding like in the original implementation, Instant-NGP uses a multi-resolution hash encoding. In addition, Instant-NGP proposes to use a 128^3 occupancy grid, making the ray marching more efficient. The grid is updated by a heuristic rule using density predictions and thresholded by a fixed value to distinguish occupied space, where points are sampled, from unoccupied areas, which are skipped. The current state-of-the-art in terms of surface reconstruction (including large outdoor scenes) is Neuralangelo [34]. Neuralangelo is based on Instant-NGP optimizing the hash grid with a coarse-to-fine approach.

Similar to NeuS [32], Neuralangelo uses signed distance functions and is improved by adding smoothing constraints.

Urban Radiance Fields [15], CLONeR [16], iMAP [17] and NICE-SLAM [18] reduce the demand for dense data by adding depth supervision. Urban Radiance Fields and CLONeR use LiDAR point clouds in addition to color images. CLONeR separates the occupancy and the color into two MLPs where the occupancy MLP is trained with LiDAR point clouds and the color MLP with images. iMAP [17] and NICE-SLAM [18] are Simultaneous Localization and Mapping (SLAM) algorithms and employ depth supervision from an RGB-D camera. More recent implementations, e.g. NeRF-SLAM [35] and Orbeez-SLAM [36], perform SLAM utilizing only monocular cameras by separating pose estimation from neural scene representation and leveraging visual odometry. For example, NeRF-SLAM uses Droid-SLAM [37] as a tracking module and Instant-NGP for scene representation.

NeRF is the preferred approach in this research because of its implicit sensor fusion of color images and range measurements (see section III-A). Meanwhile, it is a mapping framework having the following advantages: NeRFs are continuous and not discrete which allows in general a higher resolution. Implicit scene representations are more memory-efficient than explicit ones. For example, the smallest room with a volume of about 130 m^3 is represented by less than 32 MB in our experiments. Comparably, a 3D occupancy grid with 1 cm^3 resolution is more than 16 times larger. Besides occupancy, NeRFs learn color and lighting properties which could be used for further tasks, e.g. object classification or scene segmentation.

III. VIRUS-NeRF

VIRUS-NeRF is based on Instant-NGP considering its fast convergence speed and its wide adaption [34], [35]. We propose two improvements on top of the base model: First, similarly to other works [15]–[18], depth supervision is added to the color-based training, reducing the demand of dense data with a high view variation. However, instead of using expensive LiDARs or depth cameras, *VIRUS-NeRF* is based on low-cost USSs and IRSs (see chapter III-A). Second, the occupancy grid of Instant-NGP is updated by the depth measurements (see chapter III-B), making ray marching more efficient and improving the results.

A. Depth Supervision

1) *Color Rendering*: In general, NeRFs are trained as follows: During ray marching, a ray is traced in the viewing direction of every pixel. M pairs of positions and directions are sampled along this ray. These samples are the input to a MLP that is only a few layers deep. The network predicts the color ($\hat{\mathbf{c}}_j$) and density (σ_j) of each sample. Then, through volume rendering, the actual color of the ray ($\hat{\mathbf{C}}_i$) is estimated:

$$\hat{\mathbf{C}}_i = \sum_{j=1}^M T_j(1 - e^{-\sigma_j \delta_j}) \hat{\mathbf{c}}_j \quad (1)$$

where δ_j is the distance between adjacent samples and T_j is the light transmittance:

$$T_j = \exp\left(-\sum_{l=1}^{j-1} \sigma_l \delta_l\right) \quad (2)$$

Finally, the squared error between all estimated ray colors ($\hat{\mathbf{C}}_i$) and the corresponding pixels (\mathbf{C}_i) is calculated for one batch of N pixels. This loss (\mathcal{L}_c) is used for back-propagation:

$$\mathcal{L}_c = \sum_{i=1}^N \|\hat{\mathbf{C}}_i - \mathbf{C}_i\|_2^2 \quad (3)$$

2) *Depth Rendering*: As shown in iMAP [17], the depth \hat{D}_i of a pixel i can be estimated during volume rendering:

$$\hat{D}_i = \sum_{j=1}^M \omega_j d_j = \sum_{j=1}^M T_j(1 - e^{-\sigma_j \delta_j}) d_j \quad (4)$$

where d_j is the depth of sample j , $\delta_j = d_{j+1} - d_j$ is the distance between adjacent samples and T_j is the light transmittance (see equation 2). The depth rendering described in equation 4 is equivalent to the color rendering of equation 1, except that the predicted depths d_j are summed up instead of the colors $\hat{\mathbf{c}}_j$.

IRS measurements are considered to be point-like, i.e. one measurement D_i corresponds to one or few camera pixels in a close neighborhood. Analogue to the colors, the depth loss is the squared error between all estimated depths \hat{D}_i and the depth measurements D_i :

$$\mathcal{L}_{IRS} = \sum_{i=1}^N \|\hat{D}_i - D_i\|_2^2 \quad (5)$$

USSs have a wide opening angle and the exact location of the object reflecting the sound wave is unknown. This prohibits applying the same depth loss as for the IRS. However, neglecting complete absorption and specular reflections of sound waves, an error can be calculated for all predictions that are closer than the measurement:

$$\mathcal{L}_{USS} = \sum_{i=1}^N \|\hat{D}_i - D_i\|_2^2, \text{ for all } i \text{ where } \hat{D}_i < D_i - \epsilon_{USS} \quad (6)$$

where ϵ_{USS} corresponds to the accuracy of the USS. The total loss is given by the following equation:

$$\mathcal{L}_{tot} = \mathcal{L}_c + \mathcal{L}_{IRS} + \mathcal{L}_{USS} \quad (7)$$

3) *Rendering Bias*: Volume rendering is a weighted sum of distances d_j with weights $\omega_j = T_j(1 - e^{-\sigma_j \delta_j})$ (see equation 4). Let's assume that the densities σ_j are described by a positive symmetric function around the surface of an object (e.g. normal distribution: center = predicted surface location, std = uncertainty). Then, the second part of the weights ($1 - e^{-\sigma_j \delta_j}$) adopts the same symmetry as the densities. The light transmittance T_j is a monotonically decreasing function (see equation 2). Hence, the weighting ω_j is on average larger for samples before the surface of the object than afterwards and therefore, the depth \hat{D}_i is underestimated systematically.

However, the NeRF is not tied to model symmetric density functions and the bias can be absorbed into the neural network. Moreover, a few samples of high density may determine the depth estimation completely because the transmittance T_j decreases exponentially and converges fast to zero.

B. Occupancy Grid

1) *Bayesian Updating Rule*: The occupancy grid of *VIRUS-NeRF* is based on a dual updating mechanism using the NeRF predictions similar to Instant-NGP and additionally the depth measurements. The key difference to Instant-NGP is that the occupancy grid of *VIRUS-NeRF* contains values in $[0, 1]$ instead of $[0, \infty)$. This allows for a probabilistic formulation and the use of the *Bayesian Updating Rule* [3], which updates the probability of a cell being occupied c_i^{occ} or empty c_i^{emp} based on the probability $P(M_n|c_i^{occ})$ of making a measurement M_n :

$$\begin{aligned} P(c_i^{occ}|M_1, \dots, M_n) &= \frac{P(M_1, \dots, M_n|c_i^{occ})P(c_i^{occ})}{P(M_1, \dots, M_n)} \\ &= \frac{P(M_n|c_i^{occ})P(c_i^{occ}|M_1, \dots, M_{n-1})}{P(M_n|c_i^{occ})P(c_i^{occ}) + P(M_n|c_i^{emp})P(c_i^{emp})} \end{aligned} \quad (8)$$

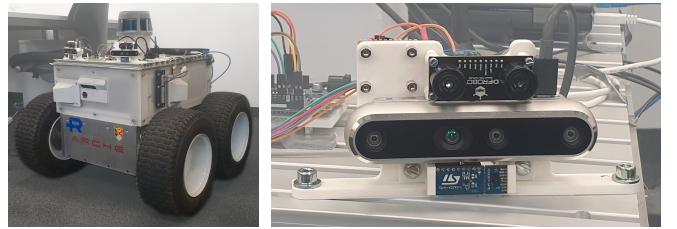
where $P(c_i^{occ}|M_1, \dots, M_n)$ is the posterior probability given measurements M_1, \dots, M_n and $P(c_i^{occ}|M_1, \dots, M_{n-1})$ is the occupancy grid before integrating M_n . $P(c_i^{occ/emp})$ can be assumed to be equal to 0.5, or it can be derived from initial information about the map.

2) *Depth-Update*: The probability of a depth sensor making a measurement $P(M_n|c_i^{occ})$ is given by the *Multiple Target Model* from the *MURIEL* method [20] proving to be the best real-time occupancy grid in a benchmark comparing different algorithms [38]. Early-stage testing shows that the low angular resolution of USSs prevent the occupancy grid from refining. Therefore, the *Depth-Update* is done uniquely based on IRS measurements.

3) *NeRF-Update*: The *NeRF-Update* uses no conventional measurement. However, the term $P(M_n = \sigma_i|c_i^{occ})$ can be thought of as the probability of the NeRF predicting the density σ_i given that a cell is occupied. The MLP outputs density predictions between zero and infinity. To be able to use the *Bayesian Updating Rule*, we introduce the projection of the density from $[0, \infty)$ to $[0, 1]$:

$$P(M_n = \sigma_i|c_i^{occ}) = \frac{1}{1 + (\frac{\sigma_T}{\sigma_i})^\zeta} \quad (9)$$

where σ_i is the predicted density by the NeRF, ζ is the slope of the mapping function and σ_T is the density threshold. If $\zeta \rightarrow \infty$, then the projection $P(M_n = \sigma_i|c_i^{occ})$ becomes a step function. If $\sigma_i > \sigma_T$, then the occupancy probability $P(M_n = \sigma_i|c_i^{occ})$ is larger than 0.5 and vice versa. The weights of the density MLP are randomly initialized in $[-\frac{1}{\sqrt{32}}, \frac{1}{\sqrt{32}}]$ leading to small σ_i values at the beginning of the training. If σ_T is fixed, then $P(M_n = \sigma_i|c_i^{occ})$ would vanish in the first few cycles. Therefore, σ_T is defined as a function of σ_i as follows: $\sigma_T = \min(\sigma_{Tmax}, \frac{1}{N} \sum_{i=1}^N \sigma_i)$ where σ_{Tmax} is the maximum density threshold and N is the batch size. For all tested models, σ_T becomes constant after a few training steps, s.t. $\sigma_T = \sigma_{Tmax}$.



(a) Super Mega Bot (b) Sensor stack: USS, camera and IRS

Fig. 2: Experimental setup

IV. EXPERIMENTS

A. Implementation

In this work, the *Taichi implementation*¹ [39] is employed because the original implementation is written in *Cuda* and therefore does not run on a CPU which is required for development.

B. Dataset

1) *Environment*: The dataset is collected by a mobile robot in two environments: *Office* (room with tables, chairs and cupboards, 72 m²) and *Common Area* (room with tables, couches and kitchen corner, 216 m²). The scenes are captured quasi-statically containing only minimal movements, e.g. a person writing on a keyboard. The robot explores the environments on 2D trajectories and makes the measurements with sensors having an overlapping FoV. This leads to a relatively low view variation of the scene.

2) *Hardware*: The Super Mega Bot (SMB) is used - a differential drive robot designed by *Inspector Bots*² (see Fig. 2a). On top of the SMB, a *RoboSense RS-LiDAR-16* and two sensor stacks are fixed. The sensor stack mounts the USS *DFRobot URM37*³, the IRS *STMicroelectronics VL53L5CX*⁴ and the camera *Intel RealSense D455*⁵ (see Fig. 2b). They are designed such that all sensors of one stack are oriented in the same direction and are as close as possible. The IRS is a time-of-flight sensor measuring an array of 8x8 pixels. The stacks are approximately 27 cm apart and point in ±14.5° to the driving direction. The configuration of the cameras having an overlapping FoV is chosen to use the calibration tool *Kalibr* [40] which estimates the intrinsic and extrinsic camera parameters simultaneously. The calibration concerning the LiDAR is done using the *Camera-LiDAR Calibration - V 2.0* [41]. The influence of an orientation error between the IRS and the camera is tested by simulating IRS measurements and adding an artificial angular error. The Nearest Neighbour Distance (NND) is not affected by an error up to 3° and therefore, the IRS and USS calibration obtained from Computer Aided Design (CAD) is sufficient. The sensors are not hardware-synchronized but recorded on

¹<https://github.com/Linyou/taichi-ngp-renderer>

²<https://www.inspectorbots.com/>

³https://wiki.dfrobot.com/URM37_V5.0_Ultrasonic_Sensor_SKU_SEN0001

⁴<https://www.st.com/en/imaging-and-photronics-solutions/vl53l5cx.html>

⁵<https://www.intelrealsense.com/depth-camera-d455/>

the same clock and the temporally closest measurements are assigned during post-processing for each sensor stack.

3) *Localisation*: The NeRF requires the measurement poses for training. Therefore, the poses are estimated by *KISS-ICP* based on the LiDAR point clouds [42] and optimized with [43] a bundle adjustment for LiDAR mapping.

C. Metrics

Traditionally, safety-relevant navigational tasks are done using instantaneous depth measurements. Therefore, *VIRUS-NeRF* is compared to momentary scans of USSs, IRSs and LiDARs. For every scene, a global map is created by projecting the LiDAR point clouds to a grid in world coordinates using the optimized poses. This global map has a 3 cm^3 cube size and is thresholded at a minimum of two points per voxel to reduce noise. For every test point, the Ground Truth (GT) consists of a 360° 2D depth scan at the height of the cameras within the global map. The depth predictions are not compared directly to the global map to mitigate the erroneous association of objects. For example, a depth prediction can be too far away, but another object is present at this location in the global map. This would lead to a small error even though the prediction is false. For *VIRUS-NeRF*, the predictions consist of a 360° 2D depth scan at the height of the cameras, which is created by volume rendering (see equation 4). For the depth sensors, the predictions are obtained by collapsing the measurements to a 2D representation in a vertical range of $\pm 5\text{ cm}$ above and below the camera height.

To compare the scans, the NND is calculated which is less sensitive to orientation errors than the Root Mean Square Error (RMSE). The NND can be calculated in two directions: The distance from every prediction point to the closest GT point is a measure of accuracy. The inverse direction describes the coverage of the GT by the prediction. The NND is given by the mean of all points and an inlier-outlier metric. Inliers are defined as points where the NND is less than 10 cm . The assessment of accuracy, coverage, and inlier ratio aligns with the evaluation criteria employed in iMAP [17] and NICE-SLAM [18], wherein the mapping coverage is denoted as *completion* and the proportion of inliers as *completion ratio*, using a threshold of 5 cm instead of 10 cm . Finally, all metrics are determined for three zones defined by the GT depth. The zones roughly represent different applications: The first zone ($0 - 1\text{ m}$) concerns safety applications, the second one ($0 - 2\text{ m}$) tasks like obstacle detection and the third one ($0 - 100\text{ m}$) path planning.

D. Mapping

1) *Results*: The average statistics for all test points and over 10 runs are summarized in Fig. 3 for the *office* and the *common area* environment. While the amplitude of the metric depends on the particular environment, the tendency is everywhere likewise: The USS has the worst accuracy of all sensors. Up to zone 2 ($0 - 2\text{ m}$), its coverage is close to the one of the LiDAR due to its large opening angle. However, in the third zone ($0 - 100\text{ m}$), the coverage of the USS worsens significantly. The IRS achieves the best accuracy while having

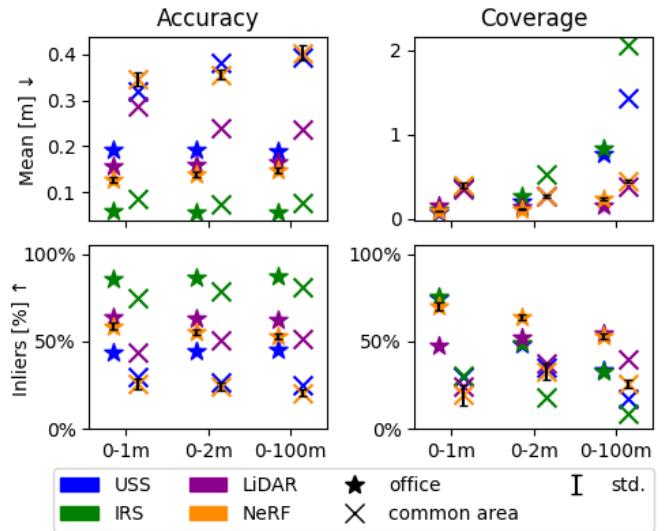


Fig. 3: *Office* and *Common Area* NND: The first column describes the accuracy and the second one the coverage. The rows show the mean NND and the inlier ($NND < 10\text{ cm}$) percentage. Each metric is calculated for three zones defined by the GT depth. *VIRUS-NeRF* is evaluated for 10 runs and the error bar indicates the standard deviation.

the worst coverage due to sparse measurements. The inferior accuracy of the LiDAR sensor compared to the IRS cannot be explained by the higher range of the LiDAR sensor because it is also present in zones 1 and 2. LiDARs retain the best coverage in the third zone ($0 - 100\text{ m}$).

VIRUS-NeRF scores a comparable coverage than the LiDAR. The accuracy of the NeRF depends on the scene: For smaller scenes, e.g. the *office*, it is slightly better than LiDARs but for larger ones, e.g. the *common area*, it exhibits performance akin to USS. The outliers ($NND > 10\text{ cm}$) can be separated into predictions that are *too close* to the robot or *too far* away relative to the GT. Analyzing this distinction for the coverage in zone three ($0 - 100\text{ m}$) shows that the largest part of all predictions is *too close* with approximately three-quarters of all outliers in the *common area* and 90% in the *office*. This trend is also visible in Fig. 4: *VIRUS-NeRF* tends to underestimate the distance to objects, leading to false-positive predictions.

2) *Discussion*: The results reflect the theoretical advantages and weaknesses of the sensors: The accuracy of the USS is limited by the poor angular resolution. The acceptable coverage at short range is in line with common USS applications where coverage is more important than accuracy, e.g. for car parking assistants [1]. The IRSs have an impressive accuracy, being at least two orders of magnitude cheaper than the LiDAR sensor. However, the LiDAR sensor seems to be the best trade-off between accuracy and coverage.

The reduced accuracy of *VIRUS-NeRF* in the *common area* compared to the *office* may be explained by the IRS having a short range of 4 m . In the *office*, 35.7% of all IRS measurements are valid compared to only 10.3% in the *common area* where the objects are more spread out.

However, for safety-relevant tasks, e.g. obstacle detection, the coverage is more important than accuracy and this one remains comparable to LiDAR point clouds in all environments. *VIRUS-NeRF* predicts more outliers *too close* leading to false-positive predictions. When visualizing the results as in Fig. 4, the estimation in front of the robot is usually accurate and the hallucinations are either sideways to the trajectory (e.g. $x = 0\text{ m}$, $y = 0\text{ m}$ in Fig. 4b) or further away from the current position of the robot (e.g. $x = 2.5\text{ m}$, $y = 3\text{ m}$ in Fig. 4b). Lateral to the path only a few measurements are taken causing erroneous predictions. This could be addressed by adding sensors pointing in these directions. Distant hallucinations are most likely caused by the volume rendering, which is biased towards underestimating the depth as explained in chapter III-A.3. If the robot moves closer to a particular region, then fewer dispensable samples influence the volume rendering and the false-positive predictions disappear, e.g. compare ($x = 2.5\text{ m}$, $y = 0\text{ m}$) in Fig. 4a and 4b. In contrast to path planning, hallucinations are not as critical for safety-relevant tasks, e.g. collision avoidance, especially if they vanish when moving closer.

E. Ablation Study

1) *Results*: In the ablation study, *VIRUS-NeRF* is compared to Instant-NGP. Additionally, the contribution of the depth supervision and the improved occupancy grid are studied, and different sensor modalities are analyzed. *VIRUS-NeRF* is developed and the hyper-parameters are fine-tuned in the *office* environment and the results of the *common area* show if the model generalizes well. The mean and inlier percentage are shown in table I. For the *common area*, the RGB-D camera outperforms all other sensor constellations because its depth images are denser and more accurate than low-cost alternatives (compare Fig. 5a and 5b). The second-best results are achieved by using *VIRUS-NeRF*. When removing either the USS or the IRS, the performance drops. The original Instant-NGP implementation is significantly worse in all metrics. When adding the depth losses and still using the occupancy grid of Instant-NGP, the results improve but do not reach the same level as *VIRUS-NeRF*. The same ablation study is repeated for the *office* environment: The results are

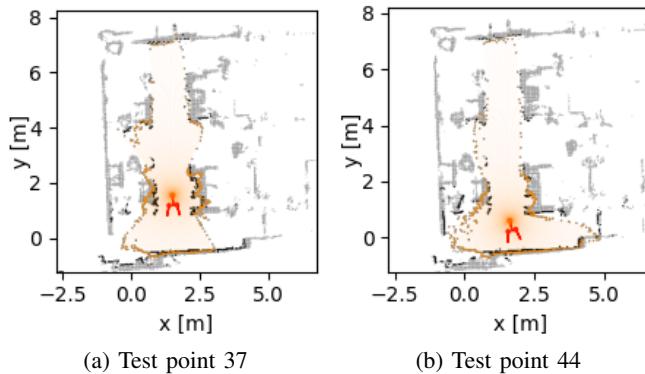


Fig. 4: *Office*: robot in red, global map in grey, GT scan in black and *VIRUS-NeRF* (USS, IRS & camera) in orange.

Occ. Grid / Sensors	Scene	Mean [m] ↓		Inliers [%] ↑	
		Acc.	Cov.	Acc.	Cov.
Instant-NGP [33]	C	0.712	1.287	0.056	0.059
	O	0.281	0.497	0.221	0.201
Instant-NGP	C	0.509	0.501	0.131	0.22
	O	0.164	0.214	0.497	0.506
<i>VIRUS-NeRF</i>	C	0.704	1.412	0.052	0.056
	O	0.277	0.476	0.231	0.215
<i>VIRUS-NeRF</i>	C	0.49	0.568	0.146	0.146
	O	0.262	0.349	0.212	0.176
<i>VIRUS-NeRF</i>	C	0.625	0.643	0.097	0.169
	O	0.23	0.374	0.415	0.46
<i>VIRUS-NeRF</i>	C	0.324	0.378	0.324	0.389
	O	0.154	0.139	0.575	0.633
<i>VIRUS-NeRF</i>	C	0.728	0.922	0.113	0.186
	O	0.168	0.238	0.531	0.554
<i>VIRUS-NeRF</i>	C	0.403	0.448	0.206	0.256
	O	0.148	0.237	0.528	0.531

TABLE I: Ablation Study: The accuracy and coverage of the NND is calculated for zone 3 (0 – 100 m) and averaged over 10 runs. The first column shows which occupancy grid and sensors are used for training (CAM is an RGB camera) and the second one the scene of interest (C for the *Common Area* and O for the *Office*). The first row is Instant-NGP [33] and the last one *VIRUS-NeRF*. Row 7 results from *VIRUS-NeRF* when not optimizing the poses with [43].

similar, with the exception that omitting pose optimization is less severe or even better in terms of inlier metrics than in the *common area*, as smaller scenes are less affected by odometry drift. The occupancy grid of Instant-NGP has slightly better mean coverage than *VIRUS-NeRF* in the *office*.

2) *Discussion*: It may be surprising that mapping without depth supervision produces poor results while standard NeRF [31] as well as Instant-NGP [33] are solely based on images showing good results in their respective studies, and other variants can represent large environments [34]. However, the dataset of this project reflects a real robot trajectory and therefore sparse measurements and a low view diversity compared to most other datasets. For example, iMAP [17], NICE-SLAM [18] and NeRF-SLAM [35] are assessed on the indoor scenes of the *Replica* dataset [44] sampling twice

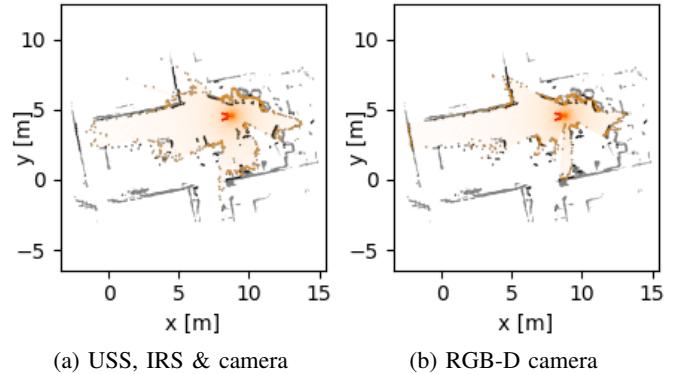


Fig. 5: *Common Area* test point 23: robot in red, global map in grey, GT scan in black and *VIRUS-NeRF* in orange.

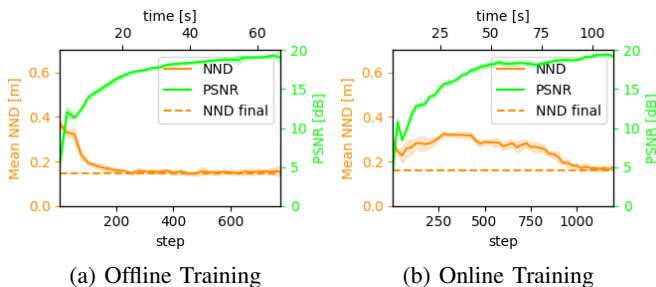


Fig. 6: *Office*: The colored area indicates the standard deviation over 10 runs. The NND is the accuracy in zone 3 (0 – 100 m).

as many images along a random trajectory. Therefore, the images of *Replica* are characterized by significantly greater variations in viewing perspectives.

F. Training Speed

All training is done on a *Nvidia Titan Xp* GPU. The absolute speed of the presented results is expected to be approximately 20% faster if using the *Cuda* implementation of Instant-NGP instead of the *Taichi* implementation utilized in this study [39]. Fig. 6a shows the average NND over 10 runs for the *office* environment during training. Using the entire dataset, *VIRUS-NeRF* converges after around 20 s while being significantly faster than Instant-NGP as explained in the following.

1) *Occupancy Grid*: On average, *VIRUS-NeRF* makes 11.64 training steps per second, in contrast to 7.96 when using the grid of Instant-NGP. This is a speed-up of 46% and can be explained by two factors: Instant-NGP samples during the first 256 training steps all 128^3 grid cells and afterwards a quarter, which is significantly more than 1024 samples used in *VIRUS-NeRF*. Additionally, the occupancy grid of *VIRUS-NeRF* relies partially on probabilistic sensor models (i.e. *Depth-Update*, see section III-B.2) instead of inferring with the NeRF network which is computationally less expensive. Accelerating the training process with the occupancy grid of *VIRUS-NeRF* could be very interesting for any real-time application based on depth supervised Instant-NGP, e.g. NeRF-SLAM [35].

2) *Offline vs. Online*: Up to here, all results are generated in offline mode where all the data is available from the beginning of the training. However, in most real-world applications the mapping algorithm should already be functional before all data is collected. Online operation is using uniquely the measurements that would be available up to this time point and data playback is performed in real-time. In this case, the training lasts for the duration of the experiment and is evaluated exclusively on already visited poses. For online operation, the NND converges after 90 s which is significantly slower than after 20 s during offline training (see Fig. 6). Similarly, the Peak Signal to Noise Ratio (PSNR) takes online much longer to converge. The final NND for online learning is 0.161 m compared to 0.148 m in the offline case. The *common area* has similar results. However, the metrics worsen again towards the end of the experiment.

The computational speed and the training data are possible limitations of the convergence speed. It seems that the available data is more important compared to the computational power because, despite making more training steps, the online algorithm converges much slower than the offline one. In the online training of the *office* environment, the NND starts to converge after half of the training process being approximately the moment when the robot turns around and drives back towards its starting position. Similarly, the metrics degrade in the *common area* when the robot makes a sharp turn and starts moving to an unseen part of the scene. These observations and the general performance difference between offline and online training suggest that a higher variety of viewpoints would improve the convergence significantly. This is in line with most NeRF algorithms that rely on a plurality of viewing angles [13], [14] and could be addressed by adding more sensors to the robot pointing sideways and backwards.

V. CONCLUSION

This study presents *VIRUS-NeRF - Vision, InfraRed and UltraSonic based Neural Radiance Fields* for local mapping. *VIRUS-NeRF* utilizes low-cost USSs and IRSs by adapting the depth supervision to sensors having a poor angular resolution. Additionally, the algorithm uses the sensors to update the occupancy grid introduced in Instant-NGP [33] which is used for ray marching. Two datasets are collected to evaluate the algorithm and to compare it to instantaneous scans of USSs, IRSs and LiDARs in 2D. The results show that *VIRUS-NeRF* has comparable coverage to LiDARs and is much better than USSs and IRSs. The accuracy of *VIRUS-NeRF* depends on the environment: For smaller scenes (*office*), it is slightly better than LiDARs but for larger ones, (*common area*), it exhibits accuracy akin to USS. Larger environments could be addressed by taking a IRS having a wider range.

The ablation study shows that the base model Instant-NGP has substantially worse results compared to *VIRUS-NeRF*. Adding the USS and the IRS to the RGB image-based training is very effective, even though the low-cost sensors have a poor angular resolution and make sparse measurements respectively. Only the more expensive RGB-D camera outperforms this sensor configuration. The occupancy grid of *VIRUS-NeRF* improves the metrics compared to the one of Instant-NGP and it makes the algorithm 46% faster. Generally, the convergence speed and accuracy could be improved by adding more sensors and taking measurements with a higher view variation. This research shows that *VIRUS-NeRF* is an effective method for local mapping based on a low-cost sensor setup.

REFERENCES

- [1] S. Mahmud, G. Khan, M. Rahman, H. Zafar, et al., “A survey of intelligent car parking system,” *Journal of applied research and technology*, vol. 11, no. 5, pp. 714–726, 2013.
- [2] H. Moravec and A. Elfes, “High resolution maps from wide angle sonar,” in *Proceedings. 1985 IEEE international conference on robotics and automation*, vol. 2. IEEE, 1985, pp. 116–121.
- [3] L. Matthies and A. Elfes, “Integration of sonar and stereo range data using a grid-based representation,” in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*. IEEE, 1988, pp. 727–733.

- [4] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.
- [5] S. Tijmons, G. C. H. E. de Croon, B. D. W. Remes, C. De Wagter, and M. Mulder, "Obstacle avoidance strategy using onboard stereo vision on a flapping wing mav," *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 858–874, 2017.
- [6] K. McGuire, G. de Croon, C. De Wagter, K. Tuyls, and H. Kappen, "Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1070–1076, 2017.
- [7] A. Masoumian, H. A. Rashwan, J. Cristiano, M. S. Asif, and D. Puig, "Monocular depth estimation using deep learning: A review," *Sensors*, vol. 22, no. 14, p. 5353, 2022.
- [8] X. Dong, M. A. Garratt, S. G. Anavatti, and H. A. Abbass, "Towards real-time monocular depth estimation for robotics: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 16940–16961, 2022.
- [9] T. Ehret, "Monocular depth estimation: a review of the 2022 state of the art," *Image Processing On Line*, vol. 13, pp. 38–56, 2023.
- [10] J. Hu, C. Bao, M. Ozay, C. Fan, Q. Gao, H. Liu, and T. L. Lam, "Deep depth completion from extremely sparse data: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [11] M. A. U. Khan, D. Nazir, A. Pagani, H. Mokayed, M. Liwicki, D. Stricker, and M. Z. Afzal, "A comprehensive survey of depth completion approaches," *Sensors*, vol. 22, no. 18, p. 6969, 2022.
- [12] Z. Xie, X. Yu, X. Gao, K. Li, and S. Shen, "Recent advances in conventional and deep learning-based depth completion: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [13] K. Gao, Y. Gao, H. He, D. Lu, L. Xu, and J. Li, "Nerf: Neural radiance field in 3d vision, a comprehensive review," *arXiv preprint arXiv:2210.00379*, 2022.
- [14] M. Debbagh, "Neural radiance fields (nerfs): A review and some recent developments," *arXiv preprint arXiv:2305.00375*, 2023.
- [15] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari, "Urban radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12932–12942.
- [16] A. Carlson, M. S. Ramanagopal, N. Tseng, M. Johnson-Roberson, R. Vasudevan, and K. A. Skinner, "Cloner: Camera-lidar fusion for occupancy grid-aided neural representations," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2812–2819, 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*, 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*, 2022, pp. 12786–12796.
- [19] A. Elfes, "A tessellated probabilistic representation for spatial robot perception and navigation," in *JPL, California Inst. of Tech., Proceedings of the NASA Conference on Space Telerobotics, Volume 2*, 1989.
- [20] K. Konolige, "Improved occupancy grids for map building," *Autonomous Robots*, vol. 4, pp. 351–367, 1997.
- [21] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous robots*, vol. 15, pp. 111–127, 2003.
- [22] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière, "Bayesian occupancy filtering for multitarget tracking: an automotive application," *The International Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, 2006.
- [23] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and tracking the driving environment with a particle-based occupancy grid," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1331–1342, 2011.
- [24] J. Saarinen, H. Andreasson, and A. J. Lilienthal, "Independent markov chain occupancy grid maps for representation of dynamic environment," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 3489–3495.
- [25] D. Meyer-Delius, M. Beinhofer, and W. Burgard, "Occupancy grid models for robot mapping in changing environments," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 26, no. 1, 2012, pp. 2024–2030.
- [26] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "On the uncertainty of self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3227–3237.
- [27] J.-T. Lin, D. Dai, and L. Van Gool, "Depth estimation from monocular images and sparse radar data," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10233–10240.
- [28] Y. Long, D. Morris, X. Liu, M. Castro, P. Chakravarty, and P. Narayanan, "Radar-camera pixel depth association for depth completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12507–12516.
- [29] O. Abdulaaty, G. Schroeder, A. Hussein, F. Albers, and T. Bertram, "Real-time depth completion using radar and camera," in *2022 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE, 2022, pp. 1–6.
- [30] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11621–11631.
- [31] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [32] 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," *arXiv preprint arXiv:2106.10689*, 2021.
- [33] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.
- [34] Z. Li, T. Müller, A. Evans, R. H. Taylor, M. Unberath, M.-Y. Liu, and C.-H. Lin, "Neuralangelo: High-fidelity neural surface reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8456–8465.
- [35] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 3437–3444.
- [36] C.-M. Chung, Y.-C. Tseng, Y.-C. Hsu, X.-Q. Shi, Y.-H. Hua, J.-F. Yeh, W.-C. Chen, Y.-T. Chen, and W. H. Hsu, "Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9400–9406.
- [37] 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. 16558–16569, 2021.
- [38] T. Collins and J. Collins, "Occupancy grid mapping: An empirical evaluation," in *2007 mediterranean conference on control & automation*. IEEE, 2007, pp. 1–6.
- [39] T. Lang, "Taichi lang: Instant-npp implementation," <https://docs.taichi-lang.org/blog/taichi-instant-npp>, accessed on 2024-01-24.
- [40] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1280–1286.
- [41] D. Tsai, S. Worrall, M. Shan, A. Lohr, and E. Nebot, "Optimising the selection of samples for robust lidar camera calibration," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 2631–2638.
- [42] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 2, pp. 1029–1036, 2023.
- [43] Z. Liu, X. Liu, and F. Zhang, "Efficient and consistent bundle adjustment on lidar point clouds," *IEEE Transactions on Robotics*, 2023.
- [44] 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.