2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

View Synthesis and 6DoF Pose Estimation in mmWave Radar Neural Radiance Fields

Ahmad Kamari, Hemant Kumar Narsani, Nan Wu, Amirreza Hajrasouliha, Bo Han, Parth Pathak

***Abstract*— Estimating a device's 6DoF pose (i.e., location and orientation) within the environment is a fundamental problem in robotics, and beyond. Millimeter-wave (mmWave) radars have emerged as an attractive alternative to optical sensors (e.g., RGB cameras) in these tasks due to their ability to operate in poor lighting and adverse conditions such as smoke and fog. This paper presents mmNeRF, a view synthesis and 6DoF pose estimation system based on neural radiance fields (NeRF) designed specifically for mmWave radars. mmNeRF requires only radar range-angle heatmaps collected in a given environment to construct its implicit neural representation, ensuring multi-view consistency and producing high-quality view synthesis. It then builds a 6DoF pose estimation framework that queries the neural model with particle filters to perform scan & matching operations to yield an accurate 6DoF pose. We evaluate mmNeRF using over 50K radar frames collected in six different indoor environments on a handheld rig equipped with a radar. Our results show that mmNeRF achieves median translation and rotation errors of 0.34m and 17.15° for single-spectrum 6DoF pose estimation and Absolute Trajectory Error (ATE) of 0.66m and 0.81 radians for continuous 6DoF pose tracking, considerably outperforming state-of-the-art solutions.**

I. INTRODUCTION

Six-degree-of-freedom (6DoF) pose estimation is a fundamental problem in robotics [1], [2], [3]. Conventional solutions rely on optical sensors such as RGB cameras, first to *scan* a given environment for collecting a set of images and identifying prominent key points (i.e., features). The pose of a mobile device can then be estimated by *matching* its current camera view with the collection of images via Perspective-n-Point (PnP) algorithms [4], [5], [6] or convolutional neural network-based architectures [7], [8]. However, these vision-based solutions do not work well in featureless and/or visually degraded environments (e.g., low light, smoke, fog, etc.). The recent emergence of mmWave radars has shown to address these challenges through the ability to operate in poor lighting and adverse conditions such as smoke or dirt [9], [10].

Direct application of scan & matching-based localization for mmWave radars is challenging primarily due to their low resolution, sparsity, and multi-path reflections. Recent efforts, such as RadarHD [10] on mmWave radar SLAM (simultaneous localization and mapping), rely on enhancing the data resolution but require LiDAR supervision to train the models. Radarize [11] addresses these limitations using angle-Doppler maps and machine learning (ML) models for rotation and velocity estimation. However, these solutions are limited to 2D mapping and pose estimation (i.e., location and yaw). 3D pose estimation is critical in numerous applications [12], [13] where the device can perform 6DoF motion, such as a drone. Furthermore, these approaches yield relative localization through multiple observations (e.g., by a moving agent, as in SLAM [11]) instead of global localization using a single radar observation.

We propose mmNeRF, a 6DoF pose estimation system based on neural radiance fields (NeRF) of mmWave radar. NeRF [14], [15] has attracted considerable attention in recent years due to its ability to create implicit neural representations of a given environment and synthesize novel views with remarkable photographic quality.

mmNeRF designs a two-step framework rooted in NeRF. First, mmNeRF leverages range-angle heatmaps built from various views in a given environment collected by mmWave radar to create a neural representation. Second, mmNeRF builds a 6DoF pose estimation framework that utilizes the neural representation of the environment as a map and performs view matching to predict the precise location and orientation of the radar. At the core of this process is our neural radar synthesis model that can create novel views for matching during the particle filter-based Monte Carlo localization [16], yielding a highly accurate location and orientation in 3D.

mmNeRF addresses various challenges when designing mmWave radar NeRF mapping and 6DoF pose estimation systems: First, a key challenge in realizing NeRF for radars is that commonly available radar output (i.e., range-angle heatmaps) has coarse resolution and is sparse due to specularity. Second, radar-based 6DoF pose estimation presents unique challenges. Unlike cameras that rely on color and visual features to extract key points [1], [12], radar is invariant to such features. The radar data, especially the range-angle heatmaps, lack dense and distinctive features typically found in vision data, making it extremely challenging to perform key point matching. Additionally, the inherent sparsity of specular reflections and noise in radar measurements makes it challenging for scan-and-matching systems to achieve high accuracy. Recent efforts, such as super-resolution [10] and rotating panoramic radar [17], address the sparsity problem but require supervision from LiDAR during training. Furthermore, the mechanically rotated radar, such as the one used in PanoRadar [17], is unsuitable for small robots and drones.

A. Kamari, H. Kumar, N. Wu, A. Hajrasouliha, B. Han, P. Pathak are with the Department of Computer Science, George Mason University, Fairfax, VA, USA, {akamari, hkumar4, nwu5, araj20, bohan, phpathak}@gmu.edu

*This work was supported by NSF grants 2402992, 2235049, 2346621, and ARL grant W911NF22-1-0216

mmNeRF represents the first attempt to design a 6DoF pose estimation system that relies entirely on a radar neural representation of the environment as the underlying map. A key advantage of this approach is its ability to produce high-fidelity view synthesis of radar images. This, in turn, enables key point/feature matching directly between synthesized views and the observed view (similar to visual scan & matching systems) to determine the radar's location and orientation. We employ classical Monte Carlo localization [16] with particle filters, utilizing radiometric loss, which measures how well a synthesized particle view aligns with observations, for 6DoF pose estimation from a single range-angular image (i.e., global localization) and real-time tracking.

We implement and evaluate mmNeRF using a large radar dataset collected using a rig equipped with a VLP16 LiDAR, IMU, Microsoft HoloLens 2 and a cascade mmWave radar [18]. Our dataset is collected in six different indoor environments with a total of more than 50K radar frames and over 1.8 km of walking trajectories.

Our results show that mmNeRF achieves a high SSIM (structural similarity index measure) [19] of 0.713 for range-angle heatmaps in novel view synthesis via its neural representation of the environment. When using the mmNeRF model as a neural representation of the environment for 6DoF pose estimation through particle filters, our scheme achieves a mean translation error of 0.34 meters and a median rotation error of 17.15 degrees on average across six scenarios for single-spectrum pose estimation. mmNeRF also achieves Absolute Trajectory Error (ATE) of 0.66 meters and 0.81 radians for translation and rotation, respectively, when performing 6DoF pose tracking over long trajectories. This is a significant improvement in both location and orientation tracking compared to state-of-the-art (SOTA) radar SLAM systems [20].

Summary of contributions. The main contributions of this paper are as follows:

(1) We introduce mmNeRF, a mmWave radar NeRF-based mapping and 6DoF pose estimation system that requires only range-angle maps for creating the neural representation of the environment and performing scan & matching-based localization and orientation detection.

(2) mmNeRF shows that a direct 6DoF pose estimation similar to visual scan & matching is feasible with mmWave radar range-angular data due to accurate synthesis of the environment from its NeRF model. It advances the state-of-the-art by increasing 6DoF pose estimation accuracy and enabling 3D orientation detection.

(3) We implement and evaluate mmNeRF through real-world datasets from a hand-held rig equipped with multiple sensors and demonstrates its superior performance in terms of mapping accuracy and 6DoF pose estimation.

II. RELATED WORK

Nueral radiance fields. Since the original NeRF [14], it has been extensively studied (survey in [15]) for improvement in training and speed, depth supervision and point clouds, and scene composition and modification. Recent studies mainly focus on improving the efficiency of NeRF with faster rendering [21], [22] and content compression [23]. Other NeRF-related research [24], [25], [26] focuses on improving the view-dependent synthesis through modeling of various factors including roughness, density, diffusion, etc. We note that the direct adaptation of these models for mmWave radar data is challenging due to the limited resolution of radars compared to cameras. NeRF-based visual localization has been studied [27], [28], [29], [30], [31], [32], [33]. However, the focus there has been on RGB images instead of sparse radar images, as in our work.

Recent works [34], [35], [36] have applied NeRF to RF signal propagation modeling. These works, however, focus on predicting wireless channels (e.g., angular spectra in NeRF^2[34]) in a bistatic setup. On the other hand, our main focus in this work is to model the NeRF for monostatic mmWave radar using range-angle heatmaps. DART [37] proposed a mmWave radar NeRF model that uses range-Doppler images. Compared to mmNeRF, it primarily focuses on mapping (without the 6DoF pose estimation) and requires accurate velocity measurements from other sensors.

mmWave radar SLAM. Prior work [20], [38] on mmWave radar-based pose estimation used either conventional or deep neural network based methods to fuse the radar observations with IMU data. Recent works [9], [10] rely on super-resolution where either generative networks or encoder-decoder architecture is used to enhance the point clouds using LiDAR supervision. Unlike these works, mmNeRF does not need any supervision from other sensor modalities. Radarize [11] proposes a radar SLAM system that leverages deep learning models to directly predict velocity and rotations from range-Doppler maps. The approach is limited to 2D (location and yaw prediction), unlike mmNeRF, which can perform 6DoF pose estimation. Recent work [17] uses a mechanically rotating radar to achieve high-quality mapping and visual recognition. While this method can perform 3D SLAM, the solution is not suitable for mixed-reality headsets and small robots.

III. MMNERF DESIGN

A. NeRF for range-angle data

Our objective in this work is to leverage the range-angle data from mmWave radars for NeRF modeling, as (i) they are most commonly and readily available through most off-the-shelf FMCW radars, and (ii) they do not require any additional information (such as the velocity measurements in DART [37]) from other sensors. The NeRF model is subsequently employed as a map model for 6DoF pose estimation task, as it provides a compact implicit representation of the environment that is crucial for implementing scan and matching for radar images. Applying scan and matching method to the radar images needs a large collection of low-resolution and sparse radar images along with a data representation that can capture

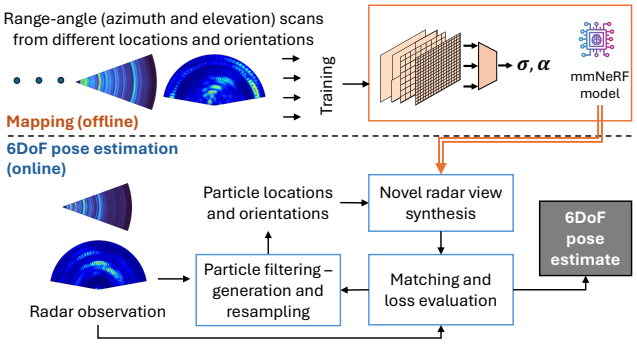


Fig. 1: Two modules of mmNeRF: (1) An offline mapping module that creates a multi-view consistent neural representation of the environment, and (2) an online module that leverages it to synthesize novel views and perform real-time 6DoF pose estimation.

the view dependency of environment voxels and allows for fast and efficient queries. NeRF provides such features by representing the scene as a continuous function that accepts the position of voxel along with the view direction and returns the view-dependent behavior of that voxel.

Network architecture. As shown in Fig. 1, we design a NeRF model, referred to as mmNeRF, to predict two parameters for each bin: the reflectance $\sigma'(r_i, \omega_i)$ and transmittance $\alpha'(r_i, \omega_i)$, where r_i is the range index and ω_i is the angle of incident specified by (θ_i, ϕ_i) for azimuth and elevation angles of the bin, respectively. mmNeRF takes a similar approach to the Instant Neural Graphics Primitives (InstantNGP) [23], leveraging its adaptive multi-resolution grid structure for efficient spatial representation. In general, the volumetric function can be written as $F(\mathrm{x}, \theta, \phi) \rightarrow (\alpha', \sigma')$ where x is the location of the bin.

Radar data representation and ray tracing. We use ray tracing to calculate the radar's predicted signal strength for a given range-angle bin. The process can then be repeated for all range-angle bins to synthesize the novel radar view as a range-angle map.

Wave propagation in radar systems is radial, meaning the spatial interpretation of each bin varies at each step of data processing (e.g., range, Doppler, and angle estimation) and corresponds to a coarse area in space. We use a specific representation where each range-angle bin corresponds to a spherical shell, essentially growing in size as we go farther away from the sphere's origin, which is the radar position. This is demonstrated in Fig. 2a.

Let us consider that we want to calculate the received signal for a bin b with range, azimuth, and elevation index i, j, and k, respectively, for a radar located at position ϑ. We randomly generate Q_{jk} rays along the azimuth and elevation in direction of $\omega(j,k) = (\theta_j, \phi_k)$ such that $(j-1) * \mathrm{bin}_{az} < \theta < j * \mathrm{bin}_{az}$ and $(k-1) * \mathrm{bin}_{el} < \phi < k * \mathrm{bin}_{el}$ where bin_{az} and bin_{el} are the bin angular resolution for azimuth and elevation, respectively. We randomly sample the rays along the range dimension, which can be written as $\vartheta + r_i \omega(q_{jk})$ where r_i is the 1D distance of the sample i on the ray. For a sample at r_m on ray q_{jk}, our mmNeRF

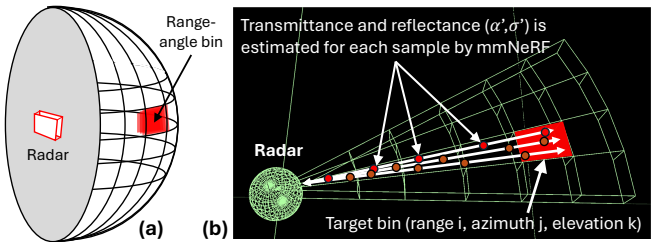


Fig. 2: The received signal for a target bin is calculated by integrating the transmittance and reflectance for all samples (as predicted by the mmNeRF model) along each ray for all rays passing through the bin.

model returns $(\alpha'(q_{jk}, m), \sigma'(q_{jk}, m))$. Then, the received signal for ray q_{jk} can be expressed as

$$R(i, q_{jk}) = \int_{m=r_{i-1}}^{r_i} g(q_{jk})\, \alpha(q_{jk}, m)\, \sigma'(q_{jk}, m)\, PL^2(r_m)\, dm \tag{1}$$

where g is the gain of the antenna for the direction of ray q_{jk}, r_m represents the distance of the sample m on the ray, PL is the free space path loss at distance r_m, $\alpha(q_{jk}, m)$ is the accumulated transmittance or the occlusion up to sample m along the ray q_{jk} represented by $\prod_{m'=1}^{i-1} \alpha'^2(q_{jk}, m')$, and $\sigma'(q_{jk}, m)$ is the reflectance for sample m. We consider the square of PL and α' to account for the forward and backward paths.

This calculation is only for one ray, and if we consider the azimuth and elevation dimensions, multiple rays passing through the same bin should be accounted for, as shown in Fig. 2b. If multiple reflectors fall into the same bin, the resulting value is the superimposed complex value. Since our model is only trained on the amplitude values, we consider the value of each bin as the average amplitude of all rays. Therefore, the received signal for bin $b(i, j, k)$ can be calculated as an integral across the rays as follows

$$R(i, j, k) = \frac{\int_j \int_k R(i, q_{jk})\, dj\, dk}{Q_{jk}} \tag{2}$$

Network Training. If we denote the prediction values from mmNeRF model as R' and the ground truth values as R, we can use the following loss function to train mmNeRF model

$$\mathrm{L} = |R - R'|^2 \tag{3}$$

In contrast to other NeRF models (e.g., [34]) that use 2D data as input, mmNeRF uses the loss function to minimize the difference between the ground truth and prediction in three dimensions (i.e., range, azimuth, and elevation).

*B. 6DoF Pose Estimation in **mmNeRF***

Compared to classical solutions of camera-based pose estimation [4], [8], [1], which rely on creating and comparing against a large database of visual key points, we benefit from neural scene representation. Our mmNeRF model encodes 3D geometry and the environment's appearance in a compact neural format. Thus, the problem

can be stated as follows: *given a radar range-angle observation collected with some unknown 6DoF pose, how can we use the novel view synthesis from mmNeRF to match and compare against the observation and to determine the location and orientation of the radar*?

To address this challenge, we use a probabilistic framework for real-time radar 6DoF pose estimation using the classical Monte Carlo localization method [16] based on particle filters that leverage the neural scene representations as the map. Similar approach has also been explored for RGB images [27]; however, as we show, relying just on radar range-angle data require further adaptations. Given the neural representation of the environment $\mathbb{F}$ from the mmNeRF model, a sequence of radar observations R_t, and relative motion measurements Δ_t between time $t-1$ and t, our system estimates the full 6DoF radar pose P_t at each timestep t. According to the Monte Carlo method [16], the goal here is to estimate the posterior probability $\mathbb{P}(P_t|\mathbb{F}, R_{1:t}, \Delta_{1:t})$ using sampling-based methods, where $R_{1:t}$ and $\Delta_{1:t}$ are the set of radar observations and the motion measurements from the beginning of the measurement until current time t, respectively.

Our approach employs a particle filter to estimate the posterior probability distribution $\mathbb{P}$. This estimation is done using a set of N weighted particles

$$S_t = \{(P_t^i, w_t^i)|i = 1, ..., N\} \tag{4}$$

where P_t^i represents the 6DoF pose of particle i at time t, and $w_t^i \in [0,1]$ denotes the corresponding importance weight. Each pose is represented as a 4×4 transformation matrix $P_t^i = \begin{pmatrix} \Psi_i & T_i \\ 0 & 1 \end{pmatrix}$ where $\Psi_i \in SO(3)$ represents the rotation and $T_i \in \mathbb{R}^3$ represents translation. The goal is to recursively update the set of particles according to the new radar and odometry measurements at each time step. A key challenge is to develop a function that can efficiently calculate a particle's weight given the observed range-angle and the particle's synthesized range-angle. Our particle filter-based localization iterates through the following three steps:

(1) Particle motion prediction. At each time step t, we start with particles from the previous step S_{t-1} and the motion measurement Δ_t for the time from $t-1$ to t. We update the pose of each particle according to the motion Δ_t as $P_t = P_{t-1}\Delta_t$.

(2) Particle weight update. After generating a set of particles, mmNeRF can synthesize each particle's view of the scene. We model this likelihood by using $\mathbb{P}(R_t|P_t^i, \mathbb{F})$, which shows the likelihood of radar observation R_t from pose P_t^i given the neural representation $\mathbb{F}$. The likelihood of these views to the radar observation will determine how close they are to the true location and orientation. We can compute and adjust the particle weights based on the model's loss value. For computational efficiency, we evaluate the likelihood using a sparse set of K randomly sampled ray directions (in both azimuth and elevation) instead of calculating it for the entire 3D radar image.

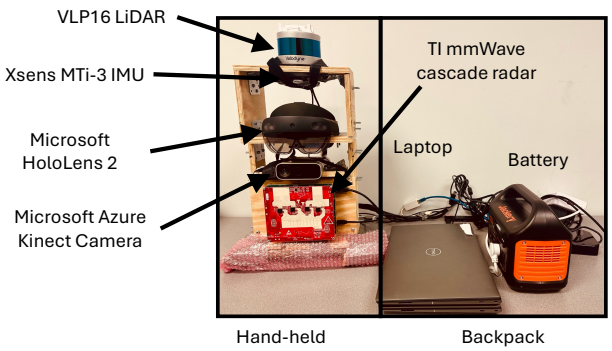


Fig. 3: Data collection rig.

We use the following function to approximate the likelihood/weight for particle i at time t

$$w_t^i = 1 - \text{norm}\left(\frac{\Sigma_{j=1}^{K}\Sigma_{l=1}^{M}(R_t(b_j^l) - R'(r(b_j^l, P_t^i)))^2}{K \times M}\right) \tag{5}$$

where M is number of bins in the range dimension, $R_t(b_j^l)$ is the radar received power for bin b_j^l, R' is the synthesis output of the model for ray $r(b_j^l, P_t^i)$ towards bin b_j^l from the pose P_t^i. The loss value essentially captures the difference in received power between the model's synthesized output for a particle i and the observed received signal over K rays and M range-angle bins. We calculate the average loss, normalize it, and subtract it from 1 to get the final weight. The weights are then normalized to sum up to 1 as they are used as probabilities for resampling in the next step. The weight of each particle essentially indicates how well the mmNeRF synthesized view at the particle's 6DoF and the current observation of the radar at time t match.

(3) Particle resampling. In this step, we maintain particles with higher weights (w_t^i) while eliminating the unlikely ones. To do so, we resample each particle i with probability w_t^i and add replacement particles to yield an updated S_t.

Our 6DoF pose estimation runs iteratively, and the final pose estimate is calculated using the weighted average of the particle poses.

One-shot prediction vs. tracking. We note that our 6DoF pose estimation framework can achieve both one-shot prediction and continuous tracking. For one-shot prediction, our algorithm assumes that the radar is stationary which makes Step 1 on motion prediction unnecessary during the iterations since $\Delta_t = 0$. For continuous tracking, the motion vectors can be integrated to adapt the particle positions before recalculating their weights based on matching and resampling. We evaluate the performance of both approaches in Sec. V.

IV. IMPLEMENTATION

A. Hardware and data collection

Radar platform. We create a data collection rig that includes a mmWave cascaded radar, a Velodyne Lidar's Puck, an Xsens MTi-3 IMU, a Microsoft HoloLens 2, and

a Microsoft Azure Kinect Camera. Fig. 3 shows our data collection rig. All devices are connected to a battery and two laptops, carried in a backpack for synchronized data collection and storage.

Our data collection leverages TI-MMWCAS cascaded MIMO radar [18] that delivers data by combining four TI radar chips, each equipped with 3 TX and 4 RX antennas. This configuration forms a 12 TX and 16 RX MIMO radar system, emulating a virtual antenna array with up to 192 elements. It achieves an azimuth and elevation angular resolution of 1.4° and 18°, respectively. The radar transmits FMCW signals at 77 GHz with a bandwidth of 3 GHz, providing a range resolution of 5 cm. We use a data capture module [39] to collect raw I/Q samples.

Lidar and IMU data are used to estimate the ground truth pose and map using Cartographer SLAM [40]. Hololens 2 is used to collect only pose data, which represents the output of visual-SLAM methods.

Data collection. We collect traces in six different environments, each ranging from 6 to 8 minutes. We perform the data collection using two methods: (1) A user holds our data collection rig and moves through the environments freely, changing the 6DoF pose by varying motion vertically and horizontally, and (2) the rig was mounted on a Jackal Unmanned Ground Vehicle (UGV) from Clearpath Robotics[41]. These traces have been carefully selected to account for different situations, such as feature-rich environments (e.g., classroom (S1, S3), atrium (S2), and auditorium (S4)) and areas with fewer features and repeating patterns (e.g., corridor spaces (S5, S6)).

B. Model implementation

We split the data for each trace with 80% for training and hold out 20% for either evaluating the view synthesis or 6DoF estimation. During the model training, we use a batch size of 22,784 rays, each with 231 samples. We use the Adam optimizer [42] with a learning rate that begins at $1e$-4 with weight decays $5e$-6. The training for a single scene typically takes around 15K iterations (approximately 45 minutes) on one NVIDIA V100 GPU. For our base model, we use the InstantNGP [23] configured with a hash table size of 2^{20}, 2 features per level, and an output MLP with two hidden layers consisting of 64 units. mmNeRF uses a PyTorch implementation that integrates tiny-cuda-nn multi-resolution hash grid for positions and spherical-harmonic with coefficients up to degree 8 for directional encoding. Once the mmNeRF model is trained, the 6DoF pose estimation essentially relies on inference and comparison in real time.

V. RESULTS

We evaluate our system in two stages. First, we focus on the performance of mmNeRF in view synthesis. The second part concentrates on using the model for one-shot prediction and tracking of 6DoF poses.

A. mmNeRF View Synthesis Evaluation

Comparison. We compare mmNeRF with two baselines that can operate on range-angular maps. All methods for comparison use the same set of input views for training.
(1) Radar simulations based on LiDAR scans (LiDAR): We use the point cloud from LiDAR scans and estimate the normals for all points to generate an occupancy grid of the environment, which subsequently can be used in a ray-tracing radar simulator. Since there are no material annotations, we assume fixed reflectance and no transmittance for the occupied grids. This baseline represents the standard practice in radar simulation[43], [44].
(2) Nearest neighbor (NN): We implement a simple nearest neighbor matching that finds the closest training point to the novel viewpoint.

Metrics. The output of the synthesis models is a tensor that represents range, azimuth, and elevation dimensions. For each elevation cut, we normalize each image to the [0, 1] range and calculate the structural similarity index measure (SSIM) metric [19]. We follow the process used in [37] for SSIM calculation to exclude empty regions by filtering them out based on pixel sample mean to avoid incorrectly high SSIM returned in case of sparse radar images. We then report the mean SSIM across all elevation cuts as the final SSIM for each 3D radar image.

Novel view synthesis. Figs. 5a and 5b show the CDF for mmNeRF SSIM compared with NN and LiDAR and between all six scenarios, respectively. Here, the SSIM is calculated between the synthesized radar range-angle spectrum maps and ground truth radar observation in our data. mmNeRF outperforms both baselines, where the median SSIM for LiDAR, NN, and mmNeRF are 0.5475, 0.6524, and 0.7128, respectively. LiDAR returns the true position and orientation (in most cases) of the reflectors based on the estimated normals of the reflectors, but lacks the correct radar returns' scale due to the lack of material specification and radar behaviors like the noises (e.g., multipath, smearing, etc.). NN delivers real radar images. However, since it is not feasible to measure every pose, the resulting images may appear inconsistent or incorrectly aligned. In comparison, mmNeRF extends the ability of other methods and achieves a higher SSIM. Fig. 4 shows the visual comparison of synthesized range-azimuth maps for different schemes. We note that these SSIM values are comparable to DART [37], which uses radar Doppler images and ground-truth velocity measurements to train the model. Alternatively, mmNeRF relies on only range-angle maps for its modeling without accurate velocity measurements. We note that we usually observe more prominent noise along the angle dimension due to antenna array design and lower resolution of cascade radar. However, this noise is reduced in our NeRF modeling. As a reference point for the above SSIM values, Fig 5a benchmarks mmNeRF against ground-truth images with synthetic Gaussian noise (25–40 dB PSNR, indicated by vertical lines). Our reconstructions achieve a fidelity

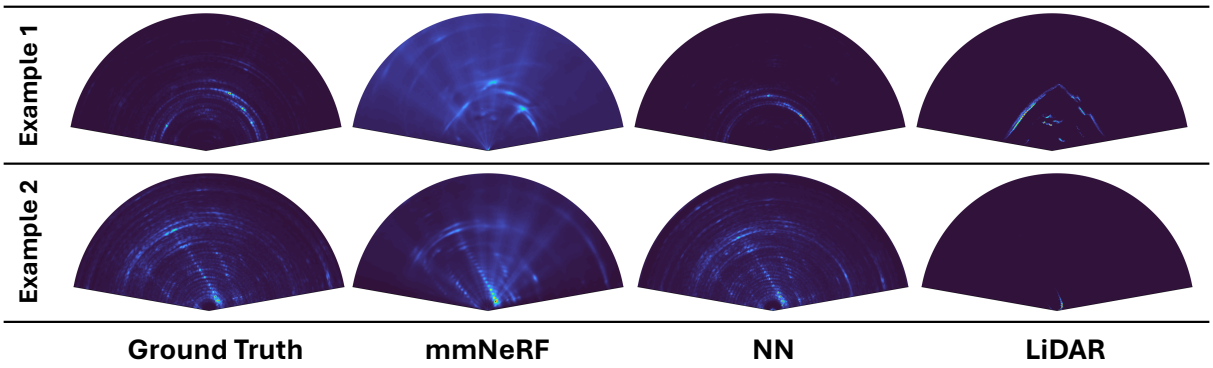
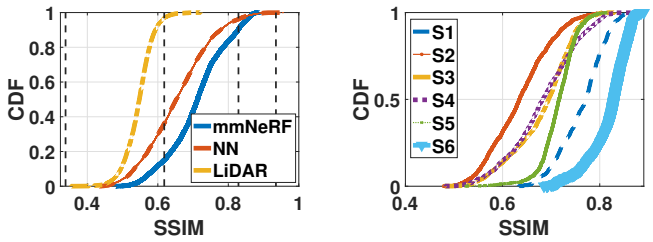
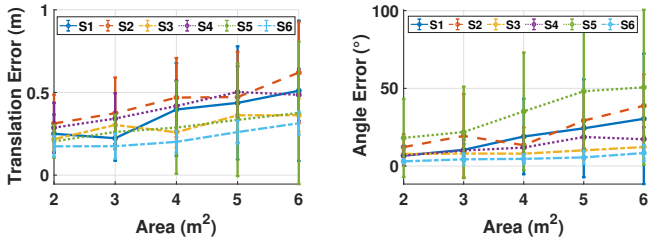


Fig. 4: Synthesis of radar views in azimuth direction: mmNeRF synthesis closely resembles the ground truth, NN returns the closest radar image, and LiDAR is unable to accurately model the radar behavior.



(a) (b)

Fig. 5: (a) SSIM comparison with NN and LIDAR along with mean SSIM for Gaussian noise (25/30/35/40dB PSNR) for reference, (b) mmNeRF SSIM for individual scenarios.



(a) (b)

Fig. 6: Particle initialization area vs. the translation and accuracy; Error increases with area, in some scenarios(S5) algorithm even struggles in small areas.

comparable to a 30–35 dB signal.

B. 6DoF Pose Estimation Accuracy

We divide the evaluation into two parts: (i) single-spectrum 6DoF pose estimation and (ii) 6DoF pose tracking over time for different trajectories.

Single-spectrum Pose Estimation. Single-spectrum pose estimation refers to estimating the radar's 6DoF pose using a single observation of a single range-angle map (a 3D matrix of range × azimuth × elevation) in a given environment. While effective for sequential data processing, SLAM methods face limitations when applied to single-frame localization because of the dependence on temporal continuity and frame-to-frame motion estimation [11]. This section explores how various initial guesses affect the pose estimation when NeRF is used as the map.

Method	ATE (m)	ATE (rad)	RE (m)	RE (rad)
milliEgo	4.904	2.191	0.087	1.106
HoloLens2	0.169	0.142	0.031	0.079
mmNeRF	**0.662**	**0.809**	**0.056**	**0.413**

TABLE I: Tracking performance averaged across scenarios.

We tested 8 data points from each of six scenarios, using various initial distances around the target pose. The z distance is fixed at 2 meters, while x and y vary. The particle filter is initialized with 300 particles, each with 128 rays, and runs on the mmNeRF model without motion prediction (see Sec. III-B).

Figs. 6a and 6b show how translation and rotation errors change across six scenarios as we vary particle initialization area. Initially, pose estimation is accurate, with mean translation and rotation errors of 0.24 meters and 9.11 degrees. As the area increases, errors rise to 0.45 meters and 26.29 degrees. Larger particle areas reduce particle density (e.g., 300 particles are not enough to cover the area), requiring more particles. This increases processing time and leads to similar radar images, raising the likelihood of incorrect convergence due to sparsity and low resolution. This happens more especially in scenarios with fewer features and repeating patterns, such as S5, which is a corridor. Another factor influencing the matching process is the accuracy of synthesized images from each particle. A better-performing NeRF model in view synthesis is more tolerant to observed errors, as seen in S6.

Our single-spectrum pose estimation takes, on average, 4.8 seconds to run on an NVIDIA GeForce RTX 4090 GPU. As we show next, our pose tracking can leverage motion vectors for real-time pose tracking where the 6DoF pose is available at more than 10 Hz.

Pose Tracking. We now demonstrate mmNeRF's ability to track 6DoF poses over time on given trajectories. Similar to single-spectrum 6DoF estimation, the particle filter algorithm is initialized with 300 particles and 128 rays for each particle. Upon receiving a new radar observation, the prediction step (Sec. III-B) is triggered. In the prediction, the pose of each particle from the previous time step is updated according to the IMU data (from Xsens MTi-3 IMU). The weights of particles are then updated based on the new radar observation, and resampling happens

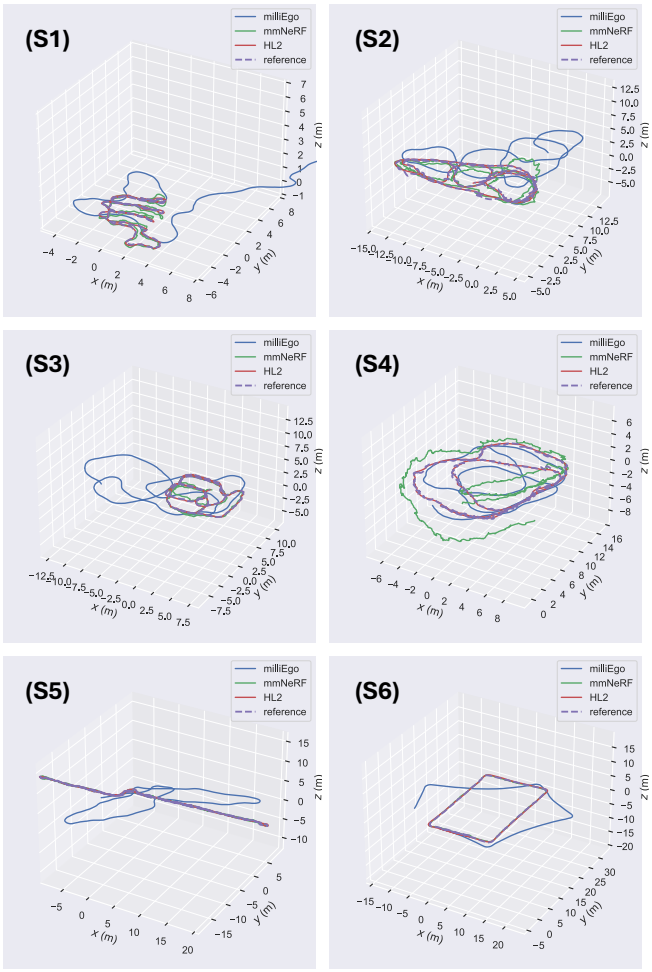


Fig. 7: mmNeRF 6DoF pose tracking over example trajectories in S1, S2, S3, S4, S5 and S6 scenarios.

based on the new weights. This process repeats for the subsequent points on the trajectories. After convergence, we reduce the number of particles to 100. Each iteration takes approximately 100 ms on a PC with a NVIDIA GeForce RTX 4090 GPU, which allows the tracking to happen at approximately 10 Hz. For each trajectory, we run the pose tracking 10 times to account for the effect of random ray selection.

Comparison. We compare mmNeRF's tracking with two baselines.

(1) milliEgo [20]: Radar-inertial odometry estimation is performed with sparse point clouds, with a cross-attention mechanism to combine point cloud-derived motion estimates and IMU data. This baseline serves as the benchmark for SOTA 3D odometry systems using mmWave radars. Because we use a cascaded radar, we train the model with our training set data from all six scenarios.

(2) HoloLens2: The headset primarily uses four RGB cameras, a time-of-flight (ToF) IR depth sensor, and an IMU to calculate the pose. This baseline is considered representative of the visual SLAM systems.

Metrics. We use two key metrics [45]: Absolute Trajectory Error (ATE) and Relative Error (RE) for both position and rotation. ATE measures overall trajectory deviation, while RE evaluates local consistency over short segments.

Table I shows the translation and rotation errors averaged over the trajectories for the six scenarios. mmNeRF achieves a high-accuracy pose estimation with mean APE translation and rotation errors of 0.662 meters and 0.809 radians, respectively, compared to 4.904 meters and 2.191 radians for milliEgo. milliEgo performs within acceptable parameters when exposed to smooth and consistent movements, but it exhibits susceptibility to divergence when exposed to lateral movement or fast rotations. HoloLens pose estimation is expected to achieve the same accuracy as the ground truth in small areas. However, due to the long trajectories, we observe a small deviation when passing through the same location multiple times. Also, Fig. 7 shows six example trajectories of mmNeRF pose tracking along with ground truth and baselines.

Our results show three factors have a significant impact on the accuracy of pose tracking: (1) In case of sudden rotations, the algorithm observes slightly higher error before converging again. Fig. 7 shows this phenomenon in the S1-4 scenarios. (2) In the prediction step, the motion vectors available through noisy IMU data can cause the particles to drift away from the trajectory. However, our algorithm resets the error every 100 ms when a new 6DoF pose estimate is available, eliminating the long-term accumulation of errors as in dead reckoning. (3) The diversity in the features and the materials of the environment can make the signatures unique and help the particle filter converge fast. In environments with few distinguishing features, such as corridors, multiple locations can produce similar synthesis results, increasing pose estimation errors.

VI. CONCLUSION & FUTURE WORK

This paper presents mmNeRF, a novel neural radiance field-based mapping and 6DoF pose estimation system for mmWave radars. Using only range-angle maps from mmWave radar, mmNeRF builds a view-consistent neural representation for scan & matching and 6DoF pose estimation. Our evaluation shows that mmNeRF outperforms state-of-the-art solutions in novel view synthesis and 3D location and orientation estimation. Future work focuses on reducing computational complexity and sensitivity to dynamic objects. While the instant-NGP backbone improves upon original NeRF architectures, achieving real-time on-device training requires further optimization through methods like hash-grid pruning. For inference, we will explore efficient spatial data structures like 3D Gaussian Splatting [46] alongside adaptive techniques to adjust ray and particle counts. Currently, mmNeRF is vulnerable to localization errors if its selected random ray subset hits moving objects, as the filter penalizes the pose estimate, interpreting the power mismatch as a positioning error rather than a scene change. Potential solutions include implementing dynamic object detection to filter them out before ray selection in our pipeline.

References

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

[2] T. Hu, T. Scargill, F. Yang, Y. Chen, G. Lan, and M. Gorlatova, "SEESys: Online Pose Error Estimation System for Visual SLAM," in *Proceedings of ACM SenSys*, 2024.

[3] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "PVN3D: A Deep Point-Wise 3D Keypoints Voting Network for 6DoF Pose Estimation," in *Proceedings of IEEE/CVF CVPR*, 2020.

[4] S. Suwajanakorn, N. Snavely, J. J. Tompson, and M. Norouzi, "Discovery of latent 3d keypoints via end-to-end geometric reasoning," *Advances in neural information processing systems*, vol. 31, 2018.

[5] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-dof object pose from semantic keypoints," in *IEEE ICRA*, 2017, pp. 2011–2018.

[6] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," in *Proceedings of IEEE/CVF CVPR*, 2018, pp. 292–301.

[7] M. Schwarz, H. Schulz, and S. Behnke, "Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features," in *IEEE ICRA*, 2015, pp. 1329–1335.

[8] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[9] C. X. Lu, S. Rosa, P. Zhao, B. Wang, C. Chen, J. A. Stankovic, N. Trigoni, and A. Markham, "See through smoke: robust indoor mapping with low-cost mmwave radar," in *Proceedings of ACM MobiSys*, 2020, p. 14–27. [Online]. Available: https://doi.org/10.1145/3386901.3388945

[10] A. Prabhakara, T. Jin, A. Das, G. Bhatt, L. Kumari, E. Soltanaghai, J. Bilmes, S. Kumar, and A. Rowe, "High resolution point clouds from mmwave radar," in *IEEE ICRA*, 2023, pp. 4135–4142.

[11] E. Sie, X. Wu, H. Guo, and D. Vasisht, "Radarize: Enhancing radar slam with generalizable doppler-based odometry," in *Proceedings of ACM MobiSys*, 2024, p. 331–344. [Online]. Available: https://doi.org/10.1145/3643832.3661871

[12] F. Ahmad, H. Qiu, R. Eells, F. Bai, and R. Govindan, "CarMap: Fast 3D Feature Map Updates for Automobiles," in *Proceedings of USENIX NSDI*, 2020.

[13] D. Zhang, P. Zhou, B. Han, and P. Pathak, "M5: Facilitating Multi-User Volumetric Content Delivery with Multi-Lobe Multicast over mmWave," in *Proceedings of ACM SenSys*, 2022.

[14] 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.

[15] 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.

[16] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)*, vol. 2. IEEE, 1999, pp. 1322–1328.

[17] H. Lai, G. Luo, Y. Liu, and M. Zhao, "Enabling visual recognition at radio frequency," in *Proceedings of ACM MobiCom*, 2024, p. 388–403. [Online]. Available: https://doi.org/10.1145/3636534.3649369

[18] "mmWave cascade imaging radar RF evaluation module," https://www.ti.com/tool/MMWCAS-RF-EVM.

[19] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[20] C. X. Lu, M. R. U. Saputra, P. Zhao, Y. Almalioglu, P. P. de Gusmao, C. Chen, K. Sun, N. Trigoni, and A. Markham, "milliego: single-chip mmwave radar aided egomotion estimation via deep sensor fusion," in *Proceedings of ACM SenSys*, 2020.

[21] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance Fields without Neural Networks," in *Proceedings of the IEEE/CVF CVPR*, 2022.

[22] Z. Chen, T. Funkhouser, P. Hedman, and A. Tagliasacchi, "Mobilenerf: Exploiting the Polygon Rasterization Pipeline for Efficient Neural Field Rendering on Mobile Architectures," in *Proceedings of the IEEE/CVF CVPR*, 2023.

[23] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127

[24] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-NeRF: Structured view-dependent appearance for neural radiance fields," *CVPR*, 2022.

[25] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," 2021.

[26] 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 *IEEE CVPR*, 2023.

[27] D. Maggio, M. Abate, J. Shi, C. Mario, and L. Carlone, "Loc-nerf: Monte carlo localization using neural radiance fields," in *IEEE ICRA*, 2023, pp. 4018–4025.

[28] J. Han, L. L. Beyer, G. V. Cavalheiro, and S. Karaman, "Nvins: Robust visual inertial navigation fused with nerf-augmented camera pose regressor and uncertainty quantification," in *IEEE/RSJ IROS*, 2024, pp. 12 601–12 608.

[29] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," *arXiv preprint arXiv:2210.13641*, 2022.

[30] J. Liu, Q. Nie, Y. Liu, and C. Wang, "Nerf-loc: Visual localization with conditional neural radiance field," in *IEEE ICRA*, 2023.

[31] Z. Zhu, S. Peng, V. Larsson, Z. Cui, M. R. Oswald, A. Geiger, and M. Pollefeys, "NICER-SLAM: Neural Implicit Scene Encoding for RGB SLAM," in *Proceedings of 3DV*, 2024.

[32] S. Katragadda, W. Lee, Y. Peng, P. Geneva, C. Chen, C. Guo, M. Li, and G. Huang, "Nerf-vins: A real-time neural radiance field map-based visual-inertial navigation system," in *IEEE ICRA*, 2024, pp. 10 230–10 237.

[33] P. Marza, L. Matignon, O. Simonin, D. Batra, C. Wolf, and D. S. Chaplot, "Autonerf: Training implicit scene representations with autonomous agents," in *IEEE/RSJ IROS*, 2024, pp. 13 442–13 449.

[34] X. Zhao, Z. An, Q. Pan, and L. Yang, *NeRF2: Neural Radio-Frequency Radiance Fields*, 2023. [Online]. Available: https://doi.org/10.1145/3570361.3592527

[35] H. Lu, C. Vattheuer, B. Mirzasoleiman, and O. Abari, "Newrf: a deep learning framework for wireless radiation field reconstruction and channel prediction," in *Proceedings of ICML*. JMLR.org, 2024.

[36] X. Chen, Z. Feng, K. Sun, K. Qian, and X. Zhang, "Rfcanvas: Modeling rf channel by fusing visual priors and few-shot rf measurements," in *Proceedings of ACM SenSys*, 2024, p. 464–477. [Online]. Available: https://doi.org/10.1145/3666025.3699351

[37] T. Huang, J. Miller, A. Prabhakara, T. Jin, T. Laroia, Z. Kolter, and A. Rowe, "Dart: Implicit doppler tomography for radar novel view synthesis," in *Proceedings of IEEE/CVF CVPR*, 2024.

[38] A. Kramer, C. Stahoviak, A. Santamaria-Navarro, A.-A. Aghamohammadi, and C. Heckman, "Radar-inertial ego-velocity estimation for visually degraded environments," in *IEEE ICRA*, 2020, pp. 5739–5746.

[39] "mmWave cascade imaging radar DSP evaluation module," https://www.ti.com/tool/MMWCAS-DSP-EVM.

[40] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1271–1278.

[41] "Clearpath Jackal UGV," https://clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/.

[42] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[43] S. Auer, R. Bamler, and P. Reinartz, "Raysar - 3d sar simulator: Now open source," in *IEEE IGARSS*, 2016, pp. 6730–6733.

[44] C. Schöffmann, B. Ubezio, C. Böhm, S. Mühlbacher-Karrer, and H. Zangl, "Virtual radar: Real-time millimeter-wave radar sensor simulation for perception-driven robotics," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4704–4711, 2021.

[45] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *IEEE/RSJ IROS*, 2018, pp. 7244–7251.

[46] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, July 2023. [Online]. Available: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/