

RadarNeRF: Scene Reconstruction Using mmWave FMCW Radar

Blerim Abdullai
University of Toronto
blerim@cs.toronto.edu

Yongqiang Wang
University of Toronto
ywang@cs.toronto.edu

Abstract

We propose a novel view synthesis and simulation method for Millimeter Wave Frequency Modulated Continuous Wave Scanning Radars (mmWave FMCW Radar). Millimeter Wave Radar has high transmittance properties, allowing the signal to propagate through objects like concrete walls. Radar waves also reflect heavily off of metal vehicles and cause multipath reflections. Current radar simulations are conditioned on other sensor modalities that exploit game engine rendering and may not accurately introduce radar-specific noise artifacts. We look to build upon the recent success of Neural Radiance Fields (NeRFs) to construct novel views that may be useful for downstream testing of perception systems.

1. Introduction and Motivation

FMCW Radars have recently gained popularity as a method of localization and perception in many autonomous driving systems [1] [2] [3] [4] [5]. Due to the logistical challenge and high cost of conducting real-world experiments, obtaining data to evaluate these methods at the scale needed for autonomous driving is challenging. Accurately simulating real-world data for autonomous systems is a widely pursued topic for many of the common sensor modalities found on these systems such as cameras and LiDARs.

FMCW radars have unique artifacts and noise profiles that can make perception for common driving tasks challenging, including multi-path reflections, limited resolution, spinning doppler effects, and speckle interference [6] [4]. These artifacts are reliant on the state of the environment, and therefore, any simulation of these artifacts must be conditioned on observations of environmental phenomena that induce these artifacts. Current work in simulating radar leverages simulation engines and mesh reconstruction of scenes to produce simulated radar observations, which may not be sufficient for or scale to verify perception systems for the operation of a real-world deployment.

We look to apply current NeRF techniques based on synthesizing scene synthesis from range sensors to datasets us-

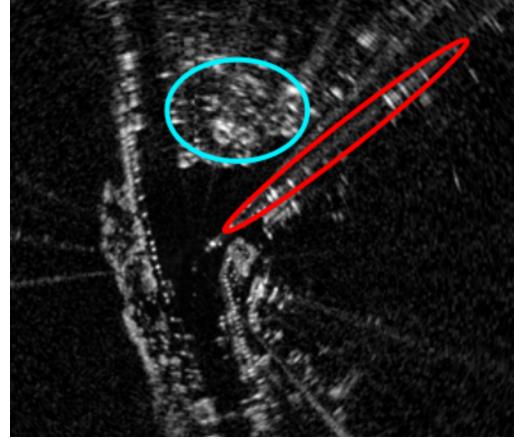


Figure 1. Figure from [5] Multipath reflections from the radar beam bouncing off the ground are shown in red. Common speckle interference is shown in cyan. In our work, we wish to emulate these artifacts and infer the physical phenomena that cause them.

ing mmWave Radar. We hope to generate novel radar scans within an existing scene with realistic artifacts.

2. Related Work

2.1. Neural Radiance Fields (NeRF)

NeRF is a novel method for synthesizing high-quality views of complex 3D scenes from a sparse set of input images [7]. This method uses a fully-connected deep neural network to model the continuous volumetric scene function, which maps 5D coordinates (spatial location (x, y, z) and viewing direction (θ, ϕ)) to volume density and view-dependent emitted radiance. Key contributions of NeRF include the model for high-fidelity view synthesis, using a differentiable volume rendering technique to project the modeled scene onto image planes, and implementing a hierarchical sampling strategy to render scenes with intricate geometry and appearance efficiently.

Based on NeRF, recent work has begun to explore using NeRFs to reconstruct range sensor readings. One such example is LiDAR-NeRF [8], which applies NeRF to the syn-

thesis of LiDAR views, addressing the shortcomings of conventional LiDAR simulators and neural networks in generating lifelike LiDAR imagery. This framework introduces a differentiable LiDAR renderer, facilitating the direct learning of 3D point geometries and attributes. Its efficacy is demonstrated through enhanced performance on the KITTI-360 and the novel NeRF-MVL datasets, marking a significant advancement over existing model-based methods.

NeRFs have also been applied to reconstruct underwater objects using forward-looking sonar [9]. The authors used a forward-looking sonar to obtain 2D observations of an underwater object and created a Neural Signed Distance field. They then used space carving to construct a final mesh. This work provides a formalism that is the most similar to ours in that they are also applying a 2D range sensor for 3D reconstruction. However, this work is mainly focused on underwater object reconstruction and not inducing realistic sensor simulation for downstream perception testing.

2.2. Radar Simulation

To the best of the author's knowledge, at the time of writing this proposal, two works have sought to simulate automotive radar. The first is Radarays, a physics-based simulation technique that leverages virtual 3D environments with the ability to tune the reflectance properties of the materials in the environment [10]. This work relies ray tracing and accurate estimation of the reflectance properties of a scene, which we hope to infer using Neural Rendering. The other is There and Back Again, which uses a combination of cyclic losses to model real radar observations using partially observed LiDAR points generated from the CARLA simulator and unaligned real radar observations [6]. The simulated observations used to condition the generative model within this work may not accurately capture enough information about the world to induce *scene specific* radar artifacts.

3. mmWave Radar View Synthesis

3.1. mmWave Radar

Autonomous driving technologies have traditionally depended on optical sensors like cameras and lidar to map out their surroundings. These sensors have led to the development of sophisticated methods for both pinpointing a vehicle's position and recognizing objects [11] [12]. However, these light-based sensors can falter under conditions where lighting is poor or obstructed.

Radar, which operates at longer wavelengths, has the theoretical advantage of being able to cut through obstructions like smoke, dust, fog, rain, and snow [13]. Given its ability to maintain performance where other sensors may fail, integrating radar into current sensor frameworks, or using it as a primary sensor, presents a promising avenue for enhancing

the measurement and interpretation capabilities essential for autonomous vehicles.

MmWave radar refers to radar technology that operates with electromagnetic wavelengths spanning from 1 to 10mm, corresponding to frequencies between 30 and 300GHz. Several national authorities have allocated particular frequency bands, notably around 76-81GHz, for vehicular uses, including Advanced Driver-Assistance Systems (ADAS). Consequently, the majority of mmWave radars utilized in commercial settings are tuned to this specified frequency range.

3.1.1 Radar Cross Section

When mmWave radar emits an electromagnetic pulse, it bounces off objects and returns. The Radar Cross Section (RCS), influenced by an object's material, size, and shape, determines the strength of the reflected signal. Essentially, RCS is a measure of an object's reflectivity, akin to the reflective area of a theoretical sphere. Large objects like vehicles and concrete walls have higher RCS values than pedestrians or minor obstacles. The "radar intensity" of a reflection, which is key for identifying objects or aiding in navigation, is the reflected power multiplied by the target's RCS. High-intensity signals often indicate significant landmarks.

3.1.2 FMCW Radar

FMCW radar, unlike simple CW radar, emits a continuous signal but varies its frequency during operation, allowing it to modulate the transmission signal in frequency or phase. This modulation enables radar measurements through runtime analysis, which is not feasible with unmodulated CW radar due to its inability to determine target range. CW radar lacks a timing reference, essential for measuring the distance to stationary objects. FMCW radar overcomes this by modulating the transmitted signal's frequency, creating a timing mark. As shown in Figure 2, the transmitted signal's frequency periodically increases or decreases, and the received echo signal's frequency shift, delayed by Δt due to the runtime shift, is used to measure distance. Unlike pulse radar, which measures runtime directly, FMCW radar assesses the phase or frequency differences between the transmitted and received signals to determine distance.

The distance R to the reflecting object can be determined by the following equation:

$$R = \frac{c\Delta t}{2} = \frac{c\Delta f}{2(\frac{df}{dt})} \quad (1)$$

where c is the speed of light, Δt is the delay time [s], Δf is the measured frequency difference [Hz], df/dt is the frequency shift per unit of time.

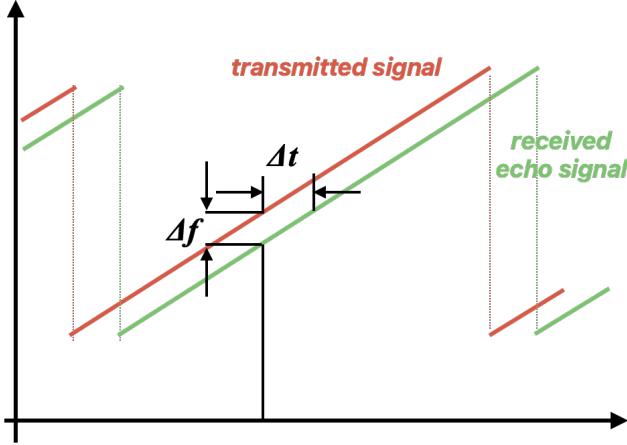


Figure 2. Ranging with an FMCW Radar system

Sensor	Specifications
Navtech CIR304-H Radar	0.0438m range solution 0.9° horizontal resolution 250m range 4Hz

Table 1. Radar Specifications

If the change in frequency is linear over a wide range, then the radar range can be determined by a simple frequency comparison. The frequency difference Δf is proportional to the distance R .

3.1.3 Scanning Radar

The two main types of mmWave radar are System-on-Chip (SoC) Radar and Scanning Radar. Scanning radar systems capture and store data differently from SoC radars. They record samples based on a single chirp and the antenna's angle during the chirp's emission. This type of radar generates two-dimensional data that maps the intensity measured across various scan angles, which can be visualized in a radar image. Processing involves applying a Fast Fourier Transform (FFT) to these samples, followed by a Constant False Alarm Rate (CFAR) filter to refine the measurement of intensity across scanned angles, resulting in a two-dimensional radar image that represents the scanned environment. The whole procedure is illustrated in Figure 3.

The mmWave FMCW Radar used for scanning is Navtech CIR304-H Radar [14], and its specifications are shown in Table 1.

3.2. Radar View Synthesis

After acquiring 2D images from the scanning radar, the subsequent phase involves constructing an image formation

model. This model simulates the outgoing radiance at any spatial coordinate, denoted by $\mathbf{x} = (r, \theta, \phi)$. The purpose of this model is to synthesize new perspectives or views by manipulating the spatial data gathered in the 2D radar images. This is achieved by projecting the captured radiance onto new coordinate sets, which allows for the creation of images from viewpoints that were not originally directly imaged by the radar.

3.2.1 Image Formation Model

As shown in Figure 4, in a radar 2D image denoted as \mathcal{I} , the pixels represent discretized bins in range and azimuth, given as (r_i, θ_i) . The value of each pixel is the cumulative energy from all the reflecting points within that bin, where each point is defined by coordinates $\{\mathbf{P}_i = (r_i, \theta_i, \phi_i); \phi_{min} \leq \phi_i \leq \phi_{max}\}$, with ϕ_i representing the elevation angle of a particular point. However, the information about the elevation angle ϕ_i is not retained in the image. This is because each azimuth column θ_i in the image is essentially a projection of a circular sector π_i , which is limited by the radar's vertical aperture range (ϕ_{min}, ϕ_{max}) and includes the z axis, onto the plane where $z = 0$.

Due to the similarities in the operation of scanning Sonar and Radar we describe the image formation models used in [9]. Considering the radar's rendering equation, let E_e denote the energy emitted by the radar towards a minuscule reflective segment \mathcal{P}_i situated on the arc $\mathcal{A}(\phi) \in \pi_i$, intersecting the point $(r_i, \theta_i, 0)$ as shown in Figure 5. The energy reflected from \mathcal{P}_i and captured by the radar is estimated by the equation:

$$E_r(r_i, \theta_i, \phi_i) = \int_{r_i - \epsilon}^{r_i + \epsilon} \frac{E_e}{r^2} \underbrace{e^{-\int_0^{r_i} \sigma(r', \theta_i, \phi_i) dr'}}_T \sigma(r, \theta_i, \phi_i) r dr \quad (2)$$

where 2ϵ represents the thickness of the reflective segment, σ is the density of particles at \mathcal{P}_i , and the term $\frac{1}{r^2}$ reflects the reduction in intensity due to the spherical spread of the radar signal. The transmittance T , denoted as $e^{-\int_0^{r_i} \sigma(r', \theta_i, \phi_i) dr'}$, accounts for the exponential weakening of the signal as it passes through particles that absorb some of its energy.

Now imagine a surface made up of numerous similar patches. The energy detected by the radar is the cumulative reflected energy from all these patches, denoted as $\{\mathcal{P}_i\} \in \mathcal{A}(\phi)$, which collectively approximate the surface. From this, we derive the image formation model:

$$\begin{aligned} I(r_i, \theta_i) &= \int_{\phi_{min}}^{\phi_{max}} \int_{r_i - \epsilon}^{r_i + \epsilon} \frac{E_e}{r^2} e^{-\int_0^{r_i} \sigma(r', \theta_i, \phi_i) dr'} \sigma(r, \theta_i, \phi) r dr d\phi \\ &= \int_{\phi_{min}}^{\phi_{max}} \int_{r_i - \epsilon}^{r_i + \epsilon} \frac{E_e}{r} T(r, \theta_i, \phi) \sigma(r, \theta_i, \phi) r dr d\phi \end{aligned} \quad (3)$$

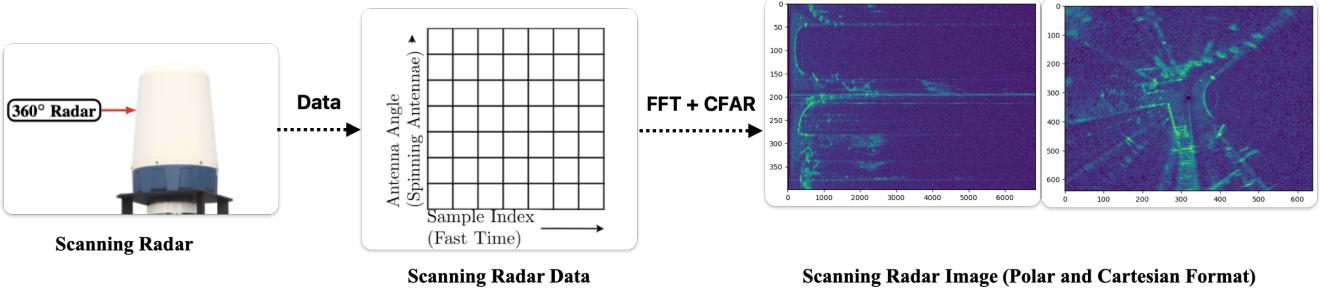


Figure 3. Data Processing Procedure for Scanning Radar

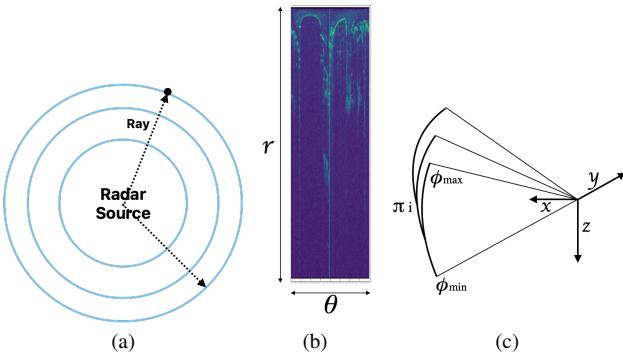


Figure 4. Radar 2D Image Formation: (a) Scanning radar projects beams across a full 360° range. (b) A representative 2D image from the Boreas dataset [14], with the array transposed, contains pixels at coordinates (r, θ) that reflect the cumulative intensity of all points on the corresponding elevation arc. (c) The columns in the image at θ_i represent the collapse of the circular sector π_i onto the $z = 0$ plane.

The surface \mathcal{S} is defined by the zero level set of \mathbf{N} :

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{N}(\mathbf{x}) = 0\} \quad (4)$$

For network training with the rendering loss, as detailed in Equation 3, we apply an approach from Wang et al. [15] to compute the density value $\sigma(\mathbf{x})$ using the Signed Distance Function (SDF):

$$\sigma(\mathbf{x}) = \max\left(\frac{-d\Phi_s(\mathbf{N}(\mathbf{x}))}{d\mathbf{x}}, 0\right) \quad (5)$$

where $\Phi_s(\tau) = (1 + e^{-s\tau})^{-1}$ represents the sigmoid function, serving as a smooth transition function approximating the occupancy indicator.

3.2.2 Sampling Process

To implement the rendering loss outlined in Eq. 3, we need to sample points along the arc at positions $p_i = (r_i, \theta_i)$ and further sample points along each radar beam. Again due to

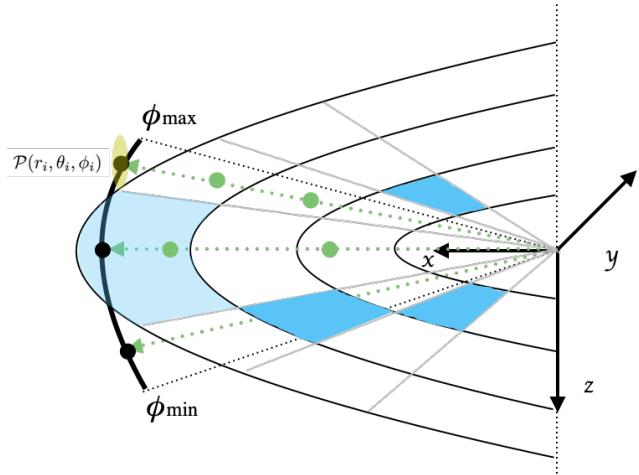


Figure 5. 1) All points $\mathbf{P} = (r, \theta, \phi)$ along the arc are mapped onto the elevation plane where $z = 0$. 2) A tiny segment of the arc is highlighted in yellow to serve as an example. 3) Our sampling strategy is visualized as follows: pixels that have been sampled in the 2D image are marked in blue, with the corresponding points on the arc depicted in black. For each of these arc points, a radar ray is drawn (indicated by a green arrow), and specific points along this ray are sampled (represented by green dots).

the similarities in data, we describe a similar sampling procedure to [9]. To ensure a dataset that includes both zero and non-zero intensity sample points, we select N_{P1} random pixels from the image and N_{P2} pixels where the intensity $I(r_i, \theta_i)$ exceeds a predefined threshold. We denote the collection of these sampled pixels as \mathcal{P} .

For every pixel $p_i \in \mathcal{P}$, stratified sampling is employed to generate point samples along the corresponding arc. The elevation angle range $[\phi_{\min}, \phi_{\max}]$ is segmented into N_A uniform intervals, resulting in an angular increment of $\Delta\phi = \frac{\phi_{\max} - \phi_{\min}}{N_A}$. Again to achieve variation, a set of random values, drawn from a Uniform(0,1) distribution and scaled by $\Delta\phi$, are added to these angles, yielding a collection of arc points $\mathbf{A}_p = \{\mathbf{P}_p = (r_i, \theta_i, \phi_{\mathbf{P}_p})\}$.

For each point \mathbf{P}_p we sample, a radar ray $\mathcal{R}_{\mathbf{P}_p}$ is formed, originating from the radar's center and extending to \mathbf{P}_p .

Along this ray, we sample $N_{\mathcal{R}} - 1$ points. This is done by selecting $N_{\mathcal{R}} - 1$ range values r' that are less than r and follow the pattern $r' = i\epsilon_r$ where i is a positive integer and ϵ is the radar's range resolution. This process generates a set of points $\mathbf{R}_{\mathbf{P}_p} = \{\mathbf{p} = (r', \theta, \phi_{\mathbf{P}_p})\}$. Together with the arc point, these form a set of $N_{\mathcal{R}}$ points $\mathbf{R}_{\mathbf{P}_p} \cup \mathbf{A}_p$ along the ray, comprising $N_{\mathcal{R}} - 1$ sampled points plus the original point on the arc. To improve robustness, we adjust the range of all points in this set by adding a small, uniformly distributed noise, $\text{Uniform}(0, 1)\epsilon_r$, to each r' value. Figure 5 depicts the sampling method where, for each radar beam, range bins are initially selected, followed by the sampling of a single point within each bin, indicated by the green points, forming the set $\mathbf{R}_{\mathbf{P}_p}$. The black point represents the modified point on the arc, \mathbf{P}_p .

The collection of $N_{\mathcal{R}}$ points, $\mathbf{R}_{\mathbf{P}_p} \cup \mathbf{A}_p$, are initially given in spherical coordinates, must be converted into a universal reference frame. This involves translating these points into Cartesian coordinates, the transforming them to the global reference frame $T_W = \begin{bmatrix} R_W & t_W \\ \mathbf{0}^T & 1 \end{bmatrix}$.

The transformed set of points, are now represented in the world frame as $\mathbf{R}_{\mathbf{P}_p}^W \cup \mathbf{A}_p^W$, and are the input for the NeRF, denoted by \mathbf{N} . The direction for each ray is the established by a unit vector:

$$\mathbf{D}(\mathbf{P}_p) = \frac{T_W \mathbf{P}_p - t_W}{|T_W \mathbf{P}_p - t_W|} \quad (6)$$

3.2.3 Discretized Image Formation Model

For computational purposes, we need to discretize the image formation model. The discretized version of Eq. 3 is presented as follows:

$$\hat{I}(r, \theta) = \sum_{\mathbf{P}_p \in \mathcal{A}_p} \frac{1}{r_{\mathbf{P}_p}} T[\mathbf{P}_p] \alpha[\mathbf{P}_p] \mathbf{M}(\mathbf{P}_p) \quad (7)$$

here \mathcal{A}_p is the arc located at (r, θ) , $r_{\mathbf{P}_p}$ is the range of the disturbed point \mathbf{P}_p on the arc, $\mathbf{M}(\mathbf{P}_p)$ is the predicted intensity at \mathbf{P}_p by the neural renderer.

The discrete opacity at consecutive samples along the ray \mathbf{p}_i and \mathbf{p}_{i+1} is:

$$\alpha[\mathbf{p}_i] = 1 - \exp \left(- \int_{\mathbf{p}_i}^{\mathbf{p}_{i+1}} \sigma(p) dp \right) \quad (8)$$

which is equivalent to :

$$\alpha[\mathbf{p}_i] = \max \left(\frac{\Phi_s(\mathbf{N}(\mathbf{p}_i)) - \Phi_s(\mathbf{N}(\mathbf{p}_{i+1}))}{\Phi_s(\mathbf{N}(\mathbf{p}_i))}, 0 \right) \quad (9)$$

The discrete transmittance value at \mathbf{P}_p (endpoint of the ray) is :

$$T[\mathbf{P}_p] = \prod_{\mathbf{p}^1 \in \mathbf{R}_{\mathbf{P}_p}} (1 - \alpha[\mathbf{p}^1]) \quad (10)$$

It is the product of one minus the opacity α of all points along the ray except α at \mathbf{P}_p .

3.2.4 Training Loss Definition

We use three terms of losses: intensity term, eikonal term, l_1 regularization term.

The intensity loss is:

$$\mathcal{L}_{\text{int}} \equiv \frac{1}{N_{\mathcal{P}}^1 + N_{\mathcal{P}}^2} \sum_{p \in \mathcal{P}} \|\hat{I}(p) - I(p)\|_1 \quad (11)$$

this loss term encourages the predicted intensity to match the intensity of the raw input radar images.

The eikonal loss [16] is:

$$\mathcal{L}_{\text{eik}} \equiv \frac{1}{N_{\mathcal{R}} N_{\mathcal{A}} (N_{\mathcal{P}}^1 + N_{\mathcal{P}}^2)} \sum_{\mathbf{x} \in \mathbf{X}} (\|\nabla \mathbf{N}(\mathbf{x})\|_2 - 1)^2 \quad (12)$$

this loss term is an implicit geometric regularization term used to regularize the SDF encouraging the network to produce smooth reconstructions.

Hence, the final training loss term is:

$$\mathcal{L} = \mathcal{L}_{\text{int}} + \lambda_{\text{eik}} \mathcal{L}_{\text{eik}} \quad (13)$$

4. Training Dataset and Network Architecture

4.1. Training Dataset

Raw radar scans [14] are 2D polar images: M azimuths $\times R$ range bins. The dataset follows Oxford's convention and embeds timestamp and encoder information into the first 11 columns (bytes) of each polar radar scan. The first 8 columns represent a 64-bit integer, the UTC timestamp of each azimuth. The next 2 columns represent a 16-bit unsigned integer, the rotational encoder value. The encoder values can be converted into azimuth angles in radians with: azimuth = encoder * np.pi / 2800. The next column is unused, preserved compatibility with Oxford's format. This gives us a raw image resolution of 400 azimuths by 2800 range bins with a maximum range of 200 m. The sample images for both polar format and Cartesian format are shown in Figure 8. The ground truth poses are obtained using a post-processed Applanix GPS, and the dataset's authors quote the accuracy of this GPS to be approximately 2-4 cm. We also intensity thresholded the radar images at a threshold of 0.2, to help the model learn the intensities of the scene.

4.2. Network Architecture

The neural implicit surface representation \mathbf{N} and the neural renderer \mathbf{M} are both multi-layer perceptrons (MLPs) with 4 hidden layers and a feature dimension of 64, as shown in Figure 7. The neural representation \mathbf{N} processes

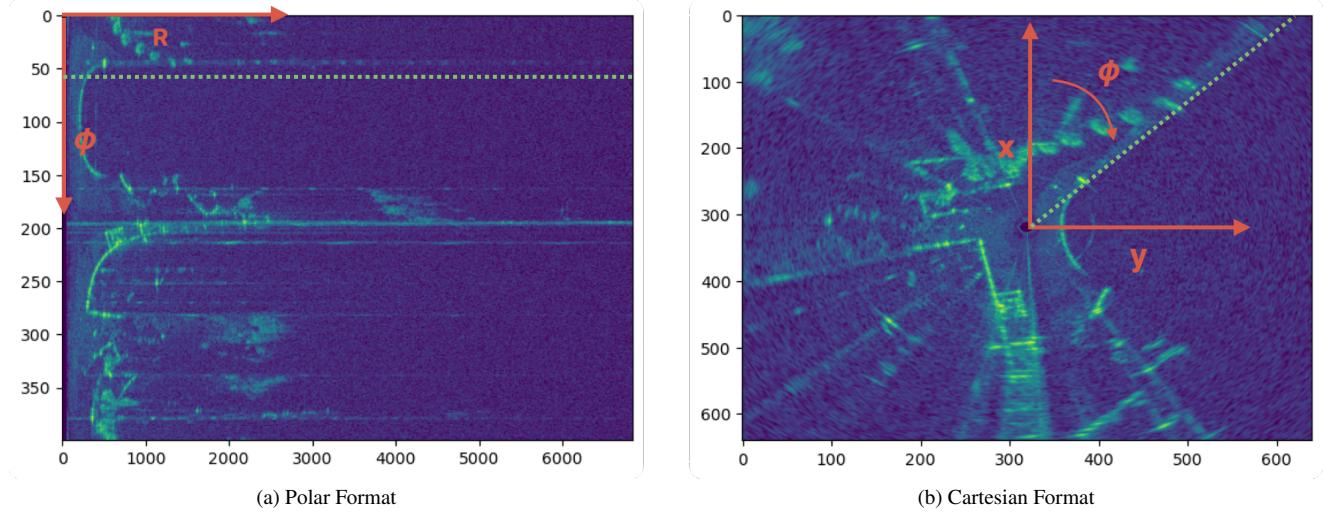


Figure 6. Raw Radar Scans 2D Images from Boreas Dataset

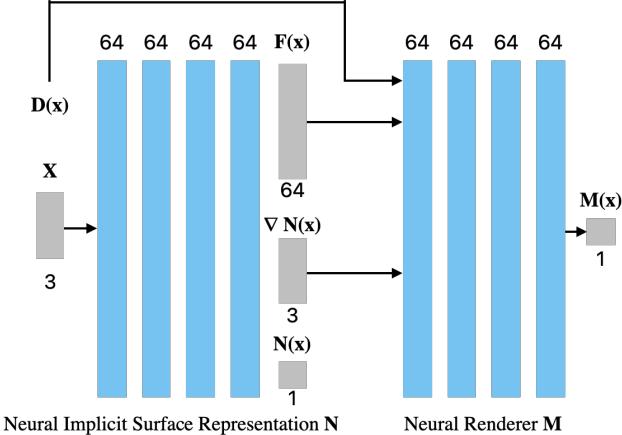


Figure 7. Network Architecture from [9]

3D spatial points \mathbf{x} , yielding their signed distance from a surface $\mathbf{N}(\mathbf{x})$ and a corresponding learned feature vector $\mathbf{F}(\mathbf{x})$. The derivative of $\mathbf{N}(\mathbf{x})$ with respect to \mathbf{x} , denoted $\nabla \mathbf{N}(\mathbf{x})$, is computed to ascertain the density $\sigma(\mathbf{x})$ (refer to Eq. 5). Moreover, the vector $\mathbf{D}(\mathbf{x})$ represents the orientation of each ray, as defined in Eq. 6. These components, $\mathbf{F}(\mathbf{x})$, $\nabla \mathbf{N}(\mathbf{x})$, and $\mathbf{D}(\mathbf{x})$, serve as inputs to the neural renderer \mathbf{M} , which then produces the outgoing radiance required for creating synthesized views.

5. Evaluation and Experimental Setup

We trained and tested two different amounts of radar scans in the Boreas Dataset [14] using the 2020 – 11 – 26 – 13 – 58 scene for training and evaluation. The raw sensor data has a high resolution and long range, which made sampling rays directly from the raw sensor computationally in-

feasible and produced unusable results. To circumvent this, we downsampled the polar images to a (200, 450) resolution. For sequence 1, we used 80 radar frames to train and held out every 5th frame for evaluation starting from frame 20 in the 2020 – 11 – 26 – 13 – 58 scene. For sequence 2 we used 400 radar frames for training and held out every 5th frame for evaluation starting again from the 20th frame in the 2020 – 11 – 26 – 13 – 58 scene. To evaluate the quality of the synthesized scans, we calculate the RMSE and MAE in the pixel space using ground truth observation from that pose and our rendered image. A comparison between other radar simulation approaches would be an interesting direction of future work; however, due to computing restrictions, time constraints, and setup complexity, we could not run any baseline comparisons. For our training hyperparameters, we sample 8 points along each ray θ and 4 samples along each ϕ per θ with 40% samples being sampled from the ground truth image and 1000 random pixels. We train for 300 epochs with a learning rate of $5e^{-4}$ and the Adam optimizer [17] and $\lambda_{eik} = 0.05$. One single training run took approximately 1 day on an NVIDIA RTX 4090 with sequence 2. We also provide qualitative results and analysis in figure 8. To render images from the NeRF we sample 30,000 random pixels in the image with the same θ and ϕ parameters.

Metric	Sequence 1	Sequence 2
RMSE	0.0429	0.0408
MAE	0.0101	0.0096

Table 2. Evaluation of the NeRF’s performance across two runs on validation set.

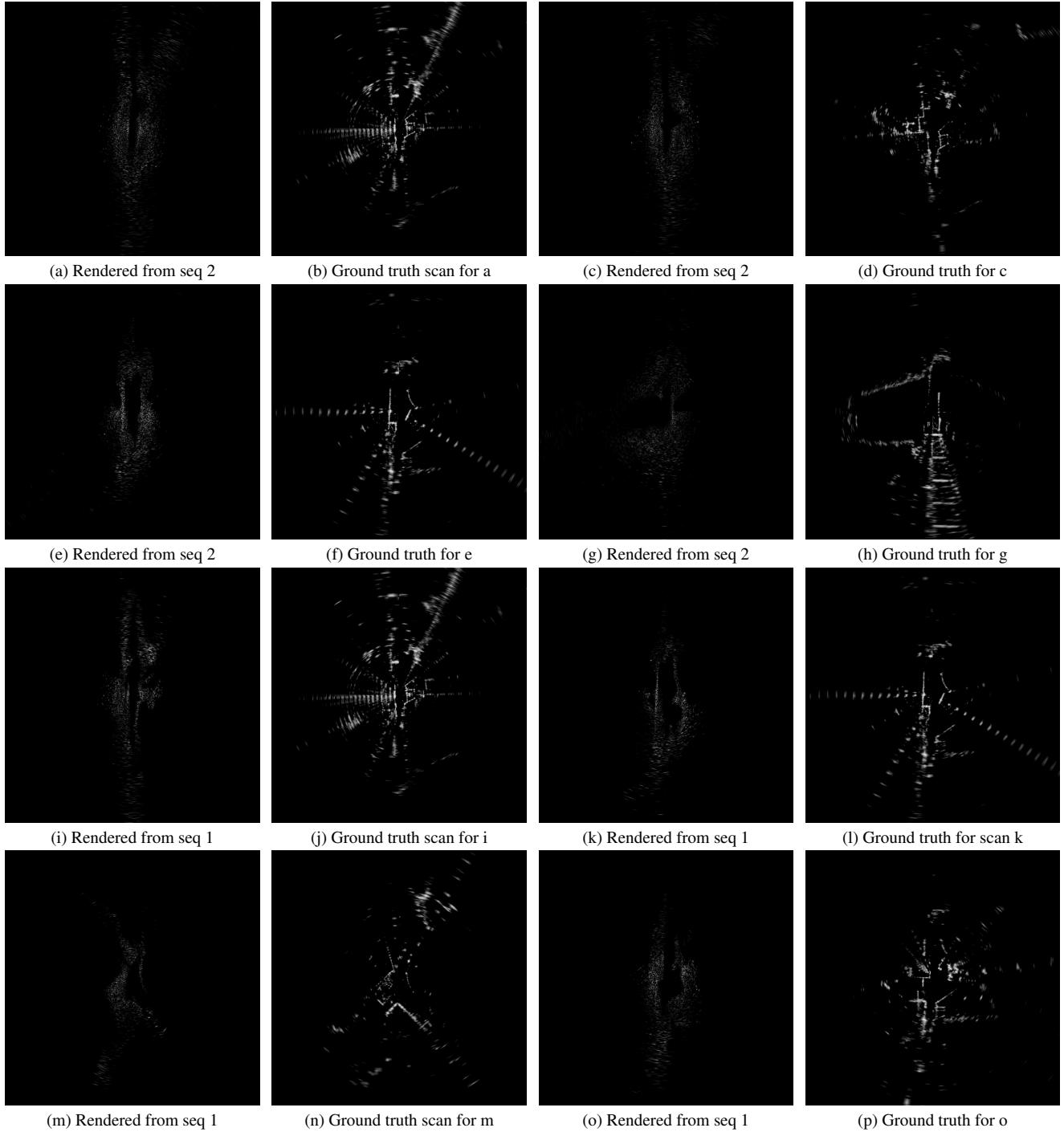


Figure 8. Qualitative comparison between synthesized view and ground truth. All images are rendered by sampling 30,000 random pixels in the polar images and 8 points along each θ with 4 points along each ϕ . As can be seen from the recovered image, the model has many “floater” artifacts that are common in NeRF models. The model also did not capture as many multipath artifacts as we had hoped, which may mean that our light transport model may not be suitable for the sensor. The model did seem to recover occupied and free space and recovered the general shape of the scan. We can also see that the model can recover more details when trained on fewer frames, suggesting that the current architecture may be underfitting.

6. Conclusion and Future Work

We have found the unique characteristics of radar to be challenging to work within the context of view synthesis. Due to the heavy propagation of radar waves and low SNR ratio, we found that the model struggles to identify regions of empty space and requires heavy amounts of sampling to obtain any results. The model is also very computationally expensive to train, requiring multiple days of training to see improvements when using a small number of observations. We believe that a variable encoding such as those used in methods like InstantNGP [18] is necessary to achieve better results. It is also well known that NeRFs tend to struggle to represent transparency, which occurs frequently within mmWave radar data, as each azimuth will often have many returns. This requires many more samples to be captured fully within the NeRF. The increase in range of up to 200 m and full 360 FOV creates a heavy computational burden on the training process. It causes us to obtain limited experimental results and, ultimately, a system that may not be feasible. We also believe the method can be improved by learning an embedding for static and moving objects similar to BlockNeRF [19]. This is because the driving dataset often contains many moving vehicles. Another interesting direction for future research would be using widely available satellite imagery to do image-to-image transfer methods for driving sequences with a clear overhead view. We believe that the method of using NeRF to recover driving sequences may be feasible by aggregating data across multiple sequences, more efficient sampling, and spatial encoding.

References

- [1] Rob Weston, Sarah Cen, Paul Newman, and Ingmar Posner. Probably unknown: Deep inverse sensor modelling in radar, 2019. 1
- [2] Tim Y. Tang, Daniele De Martini, Dan Barnes, and Paul Newman. Rsl-net: Localising in satellite images from a radar on the ground, 2020. 1
- [3] Arthur Venon, Yohan Dupuis, Pascal Vasseur, and Pierre Meriaux. Millimeter wave fmcw radars for perception, recognition and localization in automotive applications: A survey. *IEEE Transactions on Intelligent Vehicles*, 7(3):533–555, 2022. 1
- [4] Keenan Burnett, Angela P. Schoellig, and Timothy D. Barfoot. Do we need to compensate for motion distortion and doppler effects in spinning radar navigation? *IEEE Robotics and Automation Letters*, 6(2):771–778, 2021. 1
- [5] Kyle Harlow, Hyesu Jang, Timothy D. Barfoot, Ayoung Kim, and Christoffer Heckman. A new wave in robotics: Survey on recent mmwave radar applications in robotics, 2024. 1
- [6] Rob Weston, Oiwi Parker Jones, and Ingmar Posner. There and back again: Learning to simulate radar data for real-world applications, 2020. 1, 2
- [7] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. 1
- [8] Tang Tao, Longfei Gao, Guangrun Wang, Yixing Lao, Peng Chen, Hengshuang Zhao, Dayang Hao, Xiaodan Liang, Mathieu Salzmann, and Kaicheng Yu. Lidar-nerf: Novel lidar view synthesis via neural radiance fields, 2023. 1
- [9] Mohamad Qadri, Michael Kaess, and Ioannis Gkioulekas. Neural implicit surface reconstruction using imaging sonar, 2022. 2, 3, 4, 6
- [10] Alexander Mock, Martin Magnusson, and Joachim Hertzberg. Radarays: Real-time simulation of rotating fmcw radar for mobile robotics via hardware-accelerated ray tracing, 2023. 2
- [11] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 2
- [12] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017. 2
- [13] Andrew Kramer and Christoffer Heckman. Radar-inertial state estimation and obstacle detection for micro-aerial vehicles in dense fog. In *Experimental Robotics: The 17th International Symposium*, pages 3–16. Springer, 2021. 2
- [14] Keenan Burnett, David J. Yoon, Yuchen Wu, Andrew Zou Li, Haowei Zhang, Shichen Lu, Jingxing Qian, Wei-Kang Tseng, Andrew Lambert, Keith Y. K. Leung, Angela P. Schoellig, and Timothy D. Barfoot. Boreas: A multi-season autonomous driving dataset, 2023. 3, 4, 5, 6
- [15] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 4
- [16] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 5
- [17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 6
- [18] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 8
- [19] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis, 2022. 8