IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

SHENRON - Scalable, High Fidelity and EfficieNt Radar SimulatiON

Kshitiz Bansal, Gautham Reddy, and Dinesh Bharadia

***Abstract*—Radar Simulations have become an essential tool in radar algorithm development and testing due to the lack of available high-resolution radar datasets and enormous difficulty in acquiring real-world data. However, simulating radar data is challenging as existing radar simulation tools are not easily accessible, require detailed mesh inputs and take hours to simulate. To address these issues, we present SHENRON, an open-source framework that efficiently simulates high-fidelity MIMO radar data using only lidar point cloud and camera images. We show that with SHENRON, one can generate simulated data that can be used to evaluate algorithms as effectively as on real data. Further, one can perform quick iterations through a vast parameter space of the radar to find the best set of parameters for any application, significantly aiding research in radar perception and sensor fusion.**

***Index Terms*—Object detection, segmentation and categorization, simulation and animation.**

I. Introduction

THE adoption of high-resolution millimeter-wave (mmWave) radars in autonomous systems has increased due to their reliability in adverse conditions such as fog or smoke. However, radar algorithm development and widespread deployment for various applications have not kept pace with lidars and cameras, primarily due to the scarcity of available high-resolution radar data, which hinders rapid algorithm development and testing. Additionally, the performance of radar algorithms depends on the configuration used during radar data collection, and each application requires a unique set of configurations. For instance, detecting high-speed vehicles on the road over long distances requires a high maximum range and speed configuration, while tracking and identifying humans in a warehouse may require a lesser maximum range or speed but better resolution in range and speed. These different configurations in radar are obtained by choosing different values for parameters, such as the bandwidth of signal or the time between frames. Finding a good configuration for a particular application requires iterations through the vast space of radar parameters which in turn requires extensive real-world data collection. Still, these parameters are only good for the application where they have been tested. Due to this, deploying radars for new applications quite often suffer from what is known as a "cold-start problem", i.e. a significant real-world data collection effort is required to even find a good set of parameters before algorithm development can start [1].

To tackle the problem of data scarcity, simulations provide a unique opportunity. For radars, having access to simulations can provide large-scale data for algorithm development and testing, and also allow quick iterations through the radar's vast parameter space to find a small set of good parameters for a particular application. Such a simulator should fulfill four crucial requirements; **R1** - *time and compute efficiency* for rapid iteration through vast parameter space, **R2** - *ease of defining the input* for wider applicability, **R3** - *scalability* for large-scale testing in diverse conditions and **R4** - *configurability* to test different kinds of radars and parameters.

The existing commercial radar simulation tools [2], [3] are not openly accessible and require users to input high-quality mesh and material definitions for defining a scene and its objects. This is simply not possible in many cases, as generating a detailed mesh requires expert knowledge and equipment, and a poorly designed mesh could significantly hurt the performance of simulations, violating requirement **R2**. Even if one generates a detailed mesh, scaling to new scenes again requires significant effort. This makes these tools difficult to use and hard to scale to new conditions, violating requirement **R3**. Further, for scenes with complex meshes, simulation time can reach an order of hours as it takes a long time to simulate the exact physics of electromagnetic interactions, violating **R1** and making it infeasible to scale up the simulations, violating **R3**. Moreover, most of these tools have only support limited radar configuration such as single transmitter and receiver, violating **R4**.

In this paper, we present SHENRON, a highly efficient, easy-to-use and scalable method of simulating high-resolution radar data. Rather than requiring a detailed mapping of the environment generated by an expert, we design a simulator that only needs a sparse lidar point cloud and a camera image to generate high-fidelity radar data (Fig. 1) (**R2**). This means that SHENRON can immediately convert all the existing large-scale labeled lidar datasets into radar datasets with user-defined parameters, allowing for rapid algorithm development and testing data large scale (**R3**). SHENRON's design is flexible and can be used to create simulation for any radar waveform or antenna configuration (**R4**). Most importantly, a user can simply scan any space with a sparse lidar and camera and use SHENRON to rapidly evaluate multiple radar configurations for their choice of application. Hence, SHENRON, for the first time, makes the rapid design, testing and parameter search for radar perception

Manuscript received 13 August 2023; accepted 21 November 2023. Date of publication 15 December 2023; date of current version 11 January 2024. This letter was recommended for publication by Associate Editor S. Cascianelli and Editor M. Vincze upon evaluation of the reviewers' comments. *(Corresponding author: Kshitiz Bansal.)*

The authors are with UC San Diego, LA Jolla, CA 92093 USA (e-mail: ksbansal@ucsd.edu; grreddy@ucsd.edu; dineshb@ucsd.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3343629, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3343629

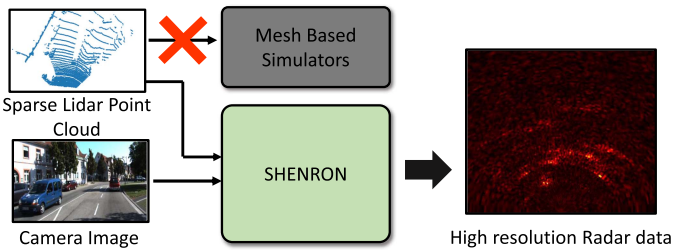


Fig. 1. Current radar simulators need high-quality mesh input to simulate radars. We present SHENRON that uses only a sparse lidar and camera input to generate high resolution radar data. Above is an example of automotive radar data generation using a lidar and camera from Kitti dataset.

possible, without needing expensive data collection effort or compute heavy mesh-based sensor simulations.

A major challenge in using raw lidar data as the basis for radar simulations is that it lacks the crucial material information required to model RF reflections. In SHENRON, we solve this problem by using the semantic information present in the camera to identify objects present in the scene. We map this semantic information to accurate RF reflection profiles of different objects using real-world measurements and physics-based modelling. With these derived reflection profiles, we are able to significantly improve the runtime of the simulator compared to commercial software (**R1**), while keeping the simulations high-fidelity.

The main objective of any simulation is to accurately predict the performance an algorithm would achieve when deployed in real world [4]. In our evaluations, we show that SHENRON's power profiles achieves a very high correlation value of **0.8433** when compared to real-world data. Further, we choose the commonly used indoor and outdoor algorithms of mapping and object detection and show how SHENRON can accurately predict the performance obtained by different algorithms in real world, only by using simulated data. We demonstrate the generalization capability of our system by evaluating it using two different radars operating at 77 GHz and 24 GHz frequencies. SHENRON is open-source and available at https://wcsng.ucsd.edu/shenron/.

In summary, we make the following contributions:

- We provide an open-source radar simulation framework, that eliminates the need for high-quality meshes and provides large-scale radar data only using sparse lidar point clouds.
- We show how a camera image can be used to obtain the missing material information in lidar point clouds.
- We derive highly accurate RF reflection profiles using real-world measurements that make the simulations fast and high-fidelity.
- We perform indoor and outdoor case studies showing how SHENRON helps in algorithm development, training and testing models and parameter tuning.

II. Related Work

Wireless Channel Solvers: Commercial wireless solvers use ray-tracing and EM wave models to provide the wireless channel for the scenes. HFSS SBR+ (shooting and bouncing rays) [2] uses geometric optics (GO), uniform theory of diffraction (UTD) and creeping wave physics to simulate radar data and Sligar et al. [5] uses HFSS SBR+ solver and generate range-doppler data. However, they do not account for the diffused scattering effects of radar signals. Remcom's Wavefarer [3], [6] further extend the framework to include diffused scattering using the models provided in [7]. The major shortcoming of these tools is the heavy dependence on the high-quality meshes that require expert equipment, are difficult to scale and have large time and memory requirements (requirements R1. R2 and R3). Furthermore, they do not provide the ability to simulate multi-antenna radar data, limiting their application for variety of use-cases (requirement R4). SHENRON presents a technique to accurately simulate high-resolution MIMO (multi-input multi-output) radar data without needing high-quality meshes, thereby reducing time and memory, as well as making it much more scalable.

Single channel radar: There have been efforts in the past to simulate radar either as a sensor to detect objects as point targets or simulate range-doppler data for single channel between transmit and receive [8], [9], [10]. Li et al. [11] uses a visibility based model of targets and radar range equation to simulate received power. However, they do not consider the effect of extended targets and the scattering phenomenon of radar signals. Machida et al. [12] propose a new method to perform SBR in an efficient manner. However, they use Torrance-sparrow model of reflection which is only an extension of specular reflections to rough surfaces. They also do not model scattering of radar signals and do not provide validation against the real data.

MIMO radar: To the best of our knowledge, the only past work that performs simulation for MIMO radar is from Schussler et al. [13]. However, as they also use meshes as input, it requires expert modelling and can not be used to convert existing lidar based datasets into radar (requirement R2). Moreover, they do not provide any details about how to obtain materials for large scale simulations, limiting their applicability to define large scale scenes (requirement R3).

Lidar to Radar translation: Deep learning based approaches L2R-GAN [14] and There and Back again [15] use a neural network to learn the sensor model of radar. However, using deep learning based simulations limits the flexibility of simulations. These approaches have only been used to simulate mechanical radar and are not easily scalable to other types of radars such as MIMO radar or radars with different antenna geometries (requirement R4). This severely constraints the applicability of these simulation frameworks. SHENRON uses physics based models to achieve high-fidelity radar simulations, hence making possible the simulation of any kind of radar signal and parameters.

III. Background and Motivation

An automotive MIMO radar uses antenna arrays as transmitter and receivers. A radar antenna radiates energy in the space defined by its field of view in form of millimeter waves [16]. These waves interact with all the objects in the scene based on their visibility from radar. The interaction of the waves with objects is in the form of specular reflections and diffused scattering [17]. The amount of energy making its way back to the radar from that object surface is determined based on the orientation of the surface and its material properties. During the reception in radar, all the reflections add up at the receiver and

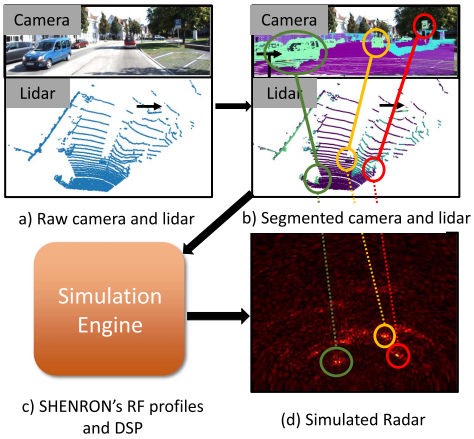


Fig. 2. Overview of SHENRON. We first obtain the scene from the lidar point clouds and infer material properties using camera images. This information along with derived RF reflection profiles is used to model accurate reflection power and is finally converted to radar data using the DSP module. Green, Yellow: Cars; Red: Traffic Sign.

sampled by an ADC (analog to digital converter). Radars either directly provide raw ADC samples or filtered point clouds. The latter are obtained by using a detection technique like CFAR (Constant False Alarm Rate) filtering [18], which detects the energy corresponding to any object's reflection, by estimating the local noise power. For either data format to be accurate, it is extremely important to correctly model the power return from the surface of objects. Failure to do so may cause the object to be indistinguishable from noise.

The objective of SHENRON is to accurately simulate the radar data for a given scene and simple object representations. In the next section, we first dive into the idea of how these object representations can be generated in a scalable way by piggy-backing on top of how a lidar perceives these objects, instead of requiring complicated meshes/environment geometries. Moving ahead, we would discuss how additional properties, like material types can be obtained to enhance the point clouds, as well as help model the reflections accurately for the radar signal. Fig. 2 provides an overview of SHENRON.

IV. SHENRON

A. Lidar Point Cloud as an Input

Typically in a simulation environment, when a mesh defines a surface, a ray-tracer returns all the points of interaction of the signal with the surfaces of objects present in the scene. This provides an impulse response of the scene in the form of interaction points and their power of reflection. This impulse response can be convolved with a radar waveform to generate the radar data. If one can obtain these interaction points in an efficient way and model the power reflected from each point accurately, it would obviate the need of using a mesh input. We make a key observation that lidar sensors scan the real environment by shooting rays. These rays travel from the lidar sensor, and reflect back towards the lidar after interacting with any surface. Due to this nature of operation, lidar directly captures the impulse response of a real world scene in the form of a point cloud, where each point denotes a point of interaction. Modern lidars have high angular resolution, and they generate point clouds with uniform polar density allowing them to cover the scene in great detail. Hence, SHENRON directly uses these lidar point clouds as input to represent the scene geometry in a simple and efficient form. This approach allows any user to directly collect real-world data of the domain of interest along with its finer details and efficiently simulate radar data. Moreover, there are abundant open-source lidar point cloud datasets that have data from real-life traffic scenarios. Using these datasets in SHENRON, one can generate large amounts of radar data, which use real-life scenes as input geometry.

B. Obtaining the Power of Reflections

Lidar point cloud provides us the points of interaction, but to faithfully capture the radar behavior, it is important to model the power of reflections very accurately. The two main factors that dictate the reflections from a surface are the surface geometry and its material properties. In simple words, the geometry or orientation of the surface determines the direction of reflection while the material properties such as permittivity, determines the amount of reflection power. For obtaining the orientation of the surface, we calculate the outward normal vectors from object surfaces using RANSAC plane fitting method in the neighbourhood of each lidar point. These normal vectors provide the orientation of the local surface of object at each lidar point. However, the information about the material properties is still absent from the lidar point cloud. In SHENRON, we provide a novel way to obtain these properties by using camera images, which are usually present in most large scale datasets along with the lidar point clouds.

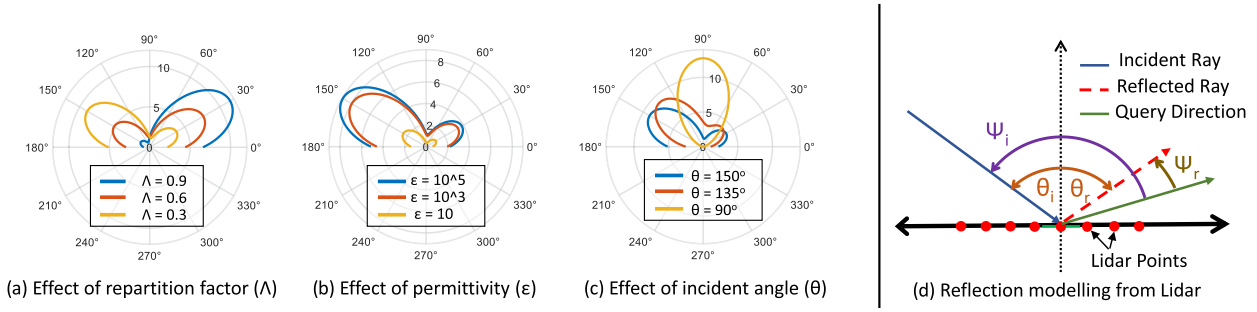
Material from camera: Recent advancements in camera semantic scene segmentation have shown the robustness of identifying and segmenting objects in a camera image [19]. We use a pretrained neural network [20] for obtaining the semantic maps of the camera image. After obtaining these semantics, we project the lidar point cloud on the camera image and associate each point with its corresponding object class from the semantic map. In the next sections, we will list the classes we used and how they were mapped to radar usable object models.

C. Modeling Radar Reflections From Lidar

In this section, first, we will describe how we physically model the radar reflections from the lidar point cloud, given the material properties. Finally, towards the end, we will describe how we mapped the classes obtained from camera to the required material properties using these physical models.

1) Interaction of EM Waves With Surfaces: We treat the EM wave interaction with a surface as a combination of penetrated energy and reflected energy. The reflected energy is further divided into specular energy and diffuse scattered energy. For an ideal planar surface, laws of reflection state that all reflected energy should go into specular direction, but due to the roughness of surfaces a lot of energy actually scatters after reflection. In our experiments, we found that this split sufficiently models the real radar data.

First, we determine the total incident power on each point of the point cloud. Each point in the point cloud is expanded to a surfel. The surfel size is chosen based on the angular separation



(a) Effect of repartition factor (Λ) (b) Effect of permittivity (ε) (c) Effect of incident angle (θ) (d) Reflection modelling from Lidar

Fig. 3. Scattering Profiles: Figure shows the effect of changing parameters on the scattering profiles. For each plot, we only change one parameter, keeping others fixed.

of lidar rays in vertical and horizontal directions. Incident power at each surfel in the point cloud is then estimated based on its distance and surface area. Now, the fraction of power reflected from each such surfel is determined by the material and the orientation of the local surface. The incident power is first split into penetrated and reflected power: $|\bar{E}_i|^2 = |\bar{E}_R|^2 + |\bar{E}_P|^2$, where $|\bar{E}_i|^2$ is the incident power, $|\bar{E}_R|^2$ is the total reflected power and $|\bar{E}_P|^2$ is the penetrated power. We use Fresnel coefficients of reflections and transmission to determine the power distribution among both terms: $\Gamma(\epsilon, \theta) = |\bar{E}_R|^2 / |\bar{E}_i|^2$ where $\Gamma(\epsilon, \theta)$ is the Fresnel reflection co-efficient, which is dependent on the permittivity (ϵ) and incident angle with respect to normal ($\theta = \theta_i$). Note that the permittivity (ϵ) is the first material property we need to determine for any material. Next, the total reflected power is split into the scatter and specular components.

2) Specular Power: We refer to specular reflections as the energy governed by the laws of reflection where the incidence angle equals the angle of reflection. At mmwave frequencies, these types of reflections carry a significant amount of power. However, the surface roughness causes some power to be spent in scattering: $|\tilde{E}_{SP}|^2 = |\bar{E}_i|^2 * \Gamma^2 * R^2$ where R^2 is the ratio of specular power ($|\tilde{E}_{SP}|^2$) over total reflected power ($|\bar{E}_R|^2$). The value of reduction ratio R depends on the surface roughness of a material [21]: $R = e^{-0.5(4\pi\sigma_h \cos\theta/\Lambda)^2}$ where σ_h is the standard deviation of surface roughness and also the second material property we need to determine for a material. Λ is the wavelength of EM wave. Since in radars, the receiver is situated close to transmitter, we only consider the specular power return when the incident ray direction is within 2 deg threshold of the normal.

3) Scattered Power: Scattered power is composed of the non-specular component of the total reflected power which is caused by surface roughness. The angular distribution of the scattered power has been studied comprehensively in past work [7]. We use the double lobe back scatter model defined for scatter reflections, as it contributes prominently in the case of co-located transmitter and receivers which is the case in radars. The double lobe model is given by

$$L^2 = \Lambda * \left(\frac{1+\cos\psi_r}{2}\right)^{\alpha_R} + (1-\Lambda)\left(\frac{1+\cos\psi_i}{2}\right)^{\alpha_I} \quad (1)$$

$$|\tilde{E}_{DS}|^2 = P_i * \Gamma^2 * (1-R^2) * L^2 \quad (2)$$

where L is the power distribution of scattered power ($|\tilde{E}_{DS}|^2$) as the function of angle; ψ_r is the angle between query and reflected ray direction, ψ_i is the angle between query and incident ray direction, Λ is the repartition factor between the amplitudes of front and back scatter lobe (Fig. 3(d)). The α_I, α_R parameter controls the lobe shape and is set from empirical evidence [7]. In case of radar, the query direction is same as the incident ray direction hence $\psi_r = 2\theta$ and $\psi_i = 0, \theta$ is the incident angle (also the query angle). Putting these values in (1):

$$L^2 = (\Lambda * (\cos^2\theta)^{\alpha_R} + (1-\Lambda)) \quad \alpha_R = 1, 2, \ldots \quad (3)$$

4) Determining Material Parameters: With the above formulation, we find that each material can be described by using the correct parameter values in above model. Specifically, for each material, we determine σ (permittivity) and σ_H (roughness standard deviation) to calculate the reflected energy. To determine these values for each material, we run real-life experiments and empirically tune the value that best describes the real-world power patterns. Specifically, we collect data for different materials at different orientations and distance from both lidar and radar. Then we compare the power profiles generated by our simulations and real world radar. We compare the total power and correlation values. We use this comparison to set the parameters of our model. Fig. 3 shows how the scattering profile changes by choosing different values for parameters in our model. Table I contains the values we used in our simulation.

TABLE I
MAPPING OF CAMERA SEMANTIC CLASSES TO MATERIAL PARAMETERS USED IN SIMULATION

Camera Class	Material	Permittivity (ε)	Roughness (mm)
Trees	Wood	2	1.7
Road, Walls	Concrete	5.24	1.7
Pedestrians	Humans	2	0.1
Cars, Trucks	Metals	100000	0.05

D. Digital Signal Processing (DSP) Implementation

Having interpreted the scene as a collection of points which reflect energy through specular or scatter means, the received signal is generated as a sum of individual reflections to get radar ADC samples. In SHENRON, the signal processing chain is modeled after a MIMO-FMCW [16] radar. This block is flexible which allows various radar system parameters including radar waveform to be changed.

To generate the ADC samples for a MIMO (Multiple Input Multiple Output) radar, reflected waves from each point are constructed with delay based on their time of flight (ToF),

amplitude based on path loss and specular/scatter reflection power return. In the MIMO setup, the ToF at each receive antenna in the array is calculated, accounting for the difference in path lengths with respect to each other. Finally, the attenuated and phase shifted reflected waves are cumulatively added with AWGN (Additive White Gaussian Noise) noise to derive the complete radar-received signal. This creates a 2D matrix, with respect to time and antenna dimension, providing range and angle information of objects in the scene. The DSP block computes the reflected signal from each point i as an attenuated and ToF delayed FMCW waveform equation. The final received waveform is the sum of all individual reflections and additive gaussian noise (4):

$$Y = \sum_{i=1}^{N} \left(L_i^{refl} * L_i^{path} * e^{j2\pi(fc+0.5*k*(t-\tau_i))(t-\tau_i)} \right) + n_{AWGN} \tag{4}$$

where L_i^{refl} is the specular/scatter loss, L_i^{path} is the path loss, τ_i is the ToF, f_c is the carrier frequency and k is the frequency change rate of the FMCW waveform. n_{AWGN} is the AWGN noise whose amplitude is obtained from real data.

Radar doppler estimation: The doppler information of radar data provides valuable insights into the movement of objects relative to the radar. It is measured using the phase changes across multiple radar chirps within a frame. To simulate accurate doppler characteristics, lidar point clouds are needed at a frame rate that matches that of the chirps, which is typically of the order of a few microseconds. However, lidar datasets often lack such high frame rates, necessitating the use of labels present in the datasets to track objects through frames. By determining the position of an object in different frames and the time between frames, the instantaneous speed can be calculated. Additionally, the relative doppler values are obtained by using the ego speed of radar measured by an IMU sensor. The relative speed of each point with respect to the radar is then used to calculate the change in phase generated due to that speed across chirps, thus simulating the doppler.

V. Evaluation

A. Evaluating Power Profiles of SHENRON

Experimental setup: We collect data using a co-located setup of 64-channel ouster lidar, Intel Real Sense Camera and a TI imaging radar as shown in Fig. 4). To exhaustively cover all the views of an object a radar would see, we consider 3 types of scenes in a controlled experiment. For *scene 1* the object moves in a circle in front of the sensor setup. In this scene, the sensor sees all the orientations of the object allowing us to evaluate quality of simulations with respect to different orientations. For *scene 2* the object moves in cross-range direction and at multiple distances. For Scene 3, the object approaches or moves away from the setup. Scene 2 and 3 represent a more common view of the objects as generally encountered while driving. Fig. 4 shows a schematic of each of these scenes. For each scene we collect 8 different runs. Each run consists of 200 frames of sensor data from each sensor, collected at a rate of 10 Hz. We repeat all these experiments for both a car and a pedestrian.

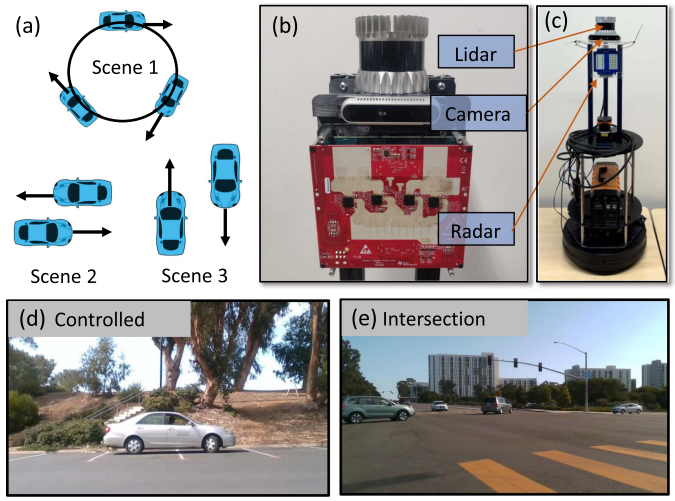


Fig. 4. (a) Different maneuvers for controlled experiments. (b) Sensor setup with TI 77 GHz radar for outdoor experiments. (c) Sensor setup for 24 GHz radar for the indoor case study. (d) Example scene from controlled experiments. (e) Example scene from a busy intersection.

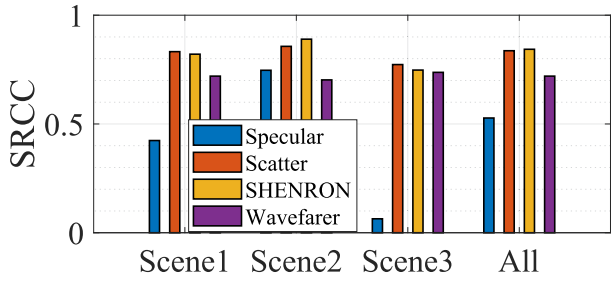


Fig. 5. Correlation coefficient values for total power reflected from the object. We also show the effect of considering specular and scattering profiles on the correlation.

Correlation of Power Return: For each scene with car, we calculate the pearson's correlation between the power profile generated by simulated and real data. For SHENRON's simulated power profile, we also show the effect of using the specular reflection model and scattering model separately to better understand the contribution of each component. While comparing with real data, commercial simulator Wavefarer [3] obtains a correlation value of 0.7199, while SHENRON maintains a high correlation value of **0.8433** when averaged among all scenarios (Fig. 5). One thing to note is that scattering alone gets a high correlation value, but only by using both specular and scatter profiles, SHENRON can obtain the correct absolute power values. This is also shown in Fig. 6 that shows the power profile for a single run from scene 1. When the car is at 90 ° orientation, the contribution of specular component is significant, while in all other orientations, scattering accounts for most of the power. This shows the importance of considering both specular and scattering components in radar simulations, to accurately model the power reflected by an object.

Qualitative comparison: We present a qualitative analysis of the contributions of each component of SHENRON in Fig. 7. Four different cases are considered: no material (where everything is set as metal), specular only, scattering only, and complete

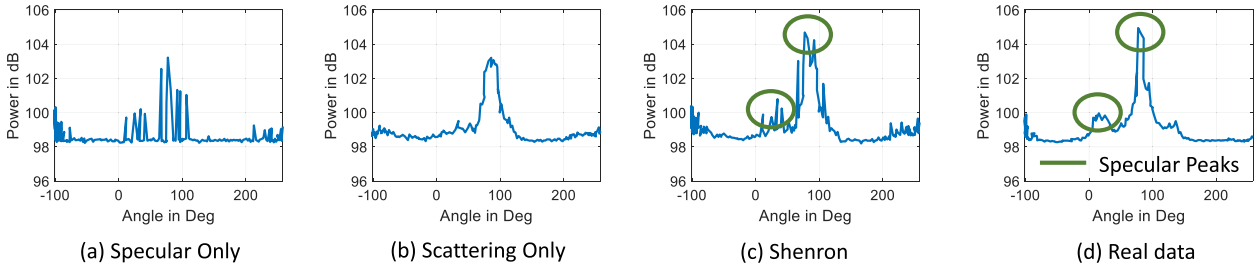


Fig. 6. Power profile of a car compared against real data for different reflection models. Specular model (a) provides correct power at specular incidence (+− 90), while scattering model (b) gets the power at almost all angles of incidence. SHENRON (c) accurately models the power profile by suitably combining the specular and scattering reflection models.

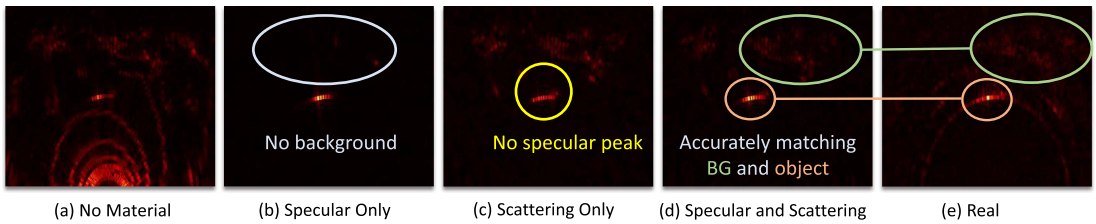


Fig. 7. Qualitative ablation study for SHENRON. The data corresponds to the image shown in Fig. 4 scene 1. The figure shows how the different components (a)–(c) of SHENRON (d), models different characteristics of the real world scene (e).

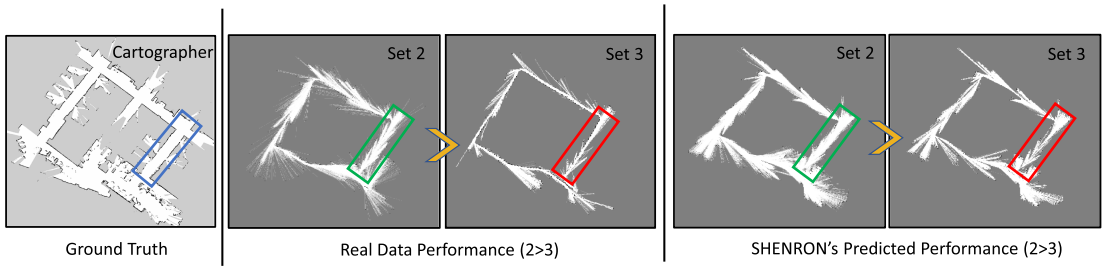


Fig. 8. Sim2Real Predictivity. The figure displays occupancy maps for parameter sets 2 and 3 using both real radar data and simulated data. A highlighted box indicates a corridor mapped differently under these parameter sets. Notably, SHENRON accurately predicts the relative performance of these sets, indicating that set 2 yields superior mapping results.

SHENRON. The scene includes a car in front of the radar, with trees and concrete structures in the background (Fig. 4). In the no material case, the simulator cannot differentiate the power of reflections based on materials, resulting in similar power returns for all points. In the specular only case, the specular reflection peak is accurately captured, but the power distribution across the car's body and the background objects (trees and concrete structures) is not represented. In the scattering only case, the power distribution is properly estimated for both the car and background objects, but the strong specular peak is missing. Finally, SHENRON accurately models both specular and scattering power returns, producing simulations that closely resemble real-world data for both the car and background objects.

B. Predicting Object Detection Performance

We assess the effectiveness of SHENRON in a real-world automotive task, specifically object detection, which is crucial for the perception stack in autonomous driving. Here, we consider a case of a general automotive scenario, i.e., a busy intersection.

TABLE II
AVERAGE PRECISION (AP) PERFORMANCE: THE RESULTS SHOW THAT OBJECT DETECTION TRAINED ON SIMULATED DATA WORKS ALMOST AS GOOD AS ONE TRAINED ON ACTUAL REAL DATA

Dataset →	Real	Simulated	Noise	RadarSimPy
Average Precision	57.5	57.8	15.0	16.38

We collect data for multiple vehicles passing through the intersection in all directions. Fig. 4(e) shows the environment where we collect data. We divide the dataset into an 8:2 ratio for training and testing. The train and test data comes from different recordings at the intersection.

Experiment Details and Results: We train three networks, one each on the real and simulated data and one on Gaussian noise. For the noise experiment, the test set is from real data. We use U-NET [22] backbone with detection heads for object detection. We report average precision (AP) metric with IOU threshold of 0.5. Table II shows the result of this experiment. The network trained only on simulated data achieves a very close performance to the network trained on the real data.

TABLE III
OBJECT DETECTION PERFORMANCE USING RADATRON [23]: THE NETWORK TRAINED BY AUGMENTING THE DATASET USING SIMULATED DATA GIVES BETTER PERFORMANCE ON REAL DATA

Training Data	Test Data	Average Precision
Real	Real	31.78
Real + Simulated	Real	47.033

Only by accurately modeling the environment and vehicle radar reflections SHENRON creates data that can be used to test deep learning networks for deployment in the real world.The marginally higher value in simulated data can be explained by the fact that both the ground truth (GT) boxes and simulated data are generated using lidar data and hence there is no misalignment between GT and simulated radar data, which is not the case for real data. Further, we see that the network trained solely on Gaussian noise performs significantly worse than the one trained on simulated or real data. This underscores the importance of accurate object representations in the input data for effective model training, validating that the simulations successfully mimic the real-world data. Notably, our approach SHENRON, enables the direct use of real-world lidar data for simulations, whereas using existing MIMO simulator *RadarSimPy* https://github.com/radarsimx/radarsimpy[Link] with sparse lidar data converted to mesh using poisson surface reconstruction predicts performance close to that of noise (Table II)

C. Case Study - Dataset Augmentation Object Detection

In this case study, we demonstrate the effectiveness of using SHENRON for data generation to enhance the available training data. We employ a pre-existing radar-based object detection network called Radatron [23] for our task. To conduct our experiments, we utilize all the data collected in scene 1-3 for the controlled experiment as described previously (Fig. 4(a) and (d)). The collected data is divided equally into training and testing datasets. The performance of Radatron on this data is presented in Table III.

Subsequently, we leverage the lidar data from the training set to generate simulated radar data. This simulated data is then incorporated into the training process. We evaluate the trained model on the original test set from real data. The outcomes reveal a notable 15.023 average precision (IoU 0.5) improvement in performance. This indicates the successful application of SHENRON in augmenting radar datasets, leading to better generalization and enhanced performance.

D. Case Study - Radar Parameter Tuning

We perform a case study to show how SHENRON can be used to solve the "cold-start" problem in radars. The aim is to understand if SHENRON can predict the relative performance while using radar with different parameters. We consider the crucial application of indoor mapping using radar that can aid in navigation during low visibility scenarios such as fire. To perform this study, we collect data for an office space of around 50 m x 50 m, using a system of lidar and camera sensors mounted on a turtle bot, as illustrated in Fig. 4. We use the cartographer algorithm [24] to obtain a ground truth map which will be used to assess the quality of map generated using radar data. We

TABLE IV
PARAMETER SETS USED FOR INDOOR STUDY. BW: BANDWIDTH (MHZ); N: SAMPLES PER CHIRP; F_s: SAMPLING RATE (MHZ); N_{chirps}: CHIRPS PER FRAME; T_c: CHIRP REPETITION INTERVAL (MS); R: RANGE; D: DOPPLER; res: RESOLUTION; max: MAXIMUM

Set No.	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6
BW	250	250	250	125	125	250
N	256	256	256	256	128	64
F_s	2.00	2.00	2.00	2.00	1.00	0.50
N_{chirps}	128	128	32	128	64	128
T_c	0.250	0.750	3.000	0.750	0.750	0.750
R_{res}	0.60	0.60	0.60	1.20	1.20	0.60
R_{max}	153.60	153.60	153.60	307.20	153.60	38.40
D_{res}	0.194	0.065	0.065	0.065	0.129	0.065
D_{max}	12.44	4.15	1.04	4.16	4.16	4.15
IoU_{sim}	**0.64**	**0.65**	0.47	**0.65**	0.58	0.62
IoU_{real}	**0.57**	**0.51**	0.27	**0.50**	0.37	0.48

collected approximately 1500 frames at a rate of 10 frames per second.

We first use lidar and camera data as input to SHENRON to generate simulated radar point clouds. We performed simulations for different parameter sets as listed in Table IV. Specifically, we changed the bandwidth used (which determines range resolution), sampling rate and number of samples per chirp (determining maximum range), time between chirps (determining maximum doppler) and number of chirps per frame (determining doppler resolution). We convert the simulated radar data into an occupancy map using the matlab's *buildMap()* function. We then compare the generated map against the ground truth map created by cartographer, using the Intersection over Union (IoU) metric. As expected, different parameter sets generate varying quality of maps, shown by the IoU values obtained in Table IV. We find that the parameter sets 1, 2 and 4 provide the most optimal values.

Now, to validate the predicted performance of different parameters by SHENRON, we also collect data using a real radar mounted on the same turtle bot. We perform 6 different data collection runs in the same office space, one for each parameter set from Table IV. Remarkably, the relative performance between different parameter sets follows the same pattern as that predicted by SHENRON (Table IV). It is clear from this evaluation that parameter set 1, 2 and 4 are the optimal sets for this application compared to sets 3, 5 and 6. Note that the purpose of this evaluation is to show how SHENRON can predict the performance of different parameter sets before even collecting real data. A more exhaustive parameter search would actually reveal the most optimal parameter sets for use. We also calculate the Sim-vs-Real Correlation Coefficient (SRCC) which was proposed by Kadian et al. [4] to measure the predictive ability of a simulation. The SRCC of this study comes out to be **0.93443**, demonstrating the effectiveness of using SHENRON in predicting the relative performance while using radar with different parameters, in a quick and efficient manner.The difference in IoU values between real and simulated data is primarily due to IoU's sensitivity to map alignment with lidar ground truth data. Since SHENRON utilizes lidar as input, it achieves higher IoU values because of perfect alignment. Fig. 8 shows the qualitative output for set 2 and 3. The total time spent in going through all the parameter sets in simulations was around 30 minutes, while the same real world test took an entire working day. This is only

TABLE V
TIME AND RAM USAGE

	Memory Usage	Time per simulation
Wavefarer	≈10GB	3146s
SHENRON	≈0.5GB	<3s

for 6 parameter sets, scaling this for 100+ sets will clearly be infeasible for real world tests. Hence, this case study clearly shows how SHENRON can significantly help in the ubiquitous deployment of radars for several indoor/outdoor applications, which otherwise would be a daunting task.

E. Time and RAM Usage

We also compare the compute resource utilization of SHENRON against commercial solvers. We run all the simulations in a Windows PC with 32 GB RAM and 3.19 GHz processor. For SHENRON, we run a multi-antenna multi-chirp simulation with the entire scene given by lidar point cloud. For Wavefarer, we run a simulation for single channel radar and a car mesh with all the material properties. The total memory used in the simulation and the time taken for the simulation is provided in Table V. SHENRON uses **20x** time less memory and **1000x** times less time for each simulation. Hence, by using lidar point clouds as input, SHENRON can provide highly accurate results in a more efficient and scalable manner.

VI. LIMITATION AND FUTURE WORKS

The radar data generated by SHENRON can be used for various applications, such as object detection and radar parameter tuning, as shown in the paper. However, it should be noted that the simulator's reliance on lidar and camera data means that it cannot use datasets with distorted lidar or camera data due to poor weather conditions. The primary objective of SHENRON is to provide researchers with access to vast amounts of radar data to facilitate the development and testing of radar algorithms. Once developed, these algorithms can simply be employed in challenging weather conditions, as radar data is impervious to weather-related interference. A potential avenue for future research is to expand the range of materials considered by the system using the method proposed. This could enable the subdivision of different parts of the same semantic object into sub-classes, such as car windows and tires, or address material interaction challenges, particularly for materials like thin cardboard that allow mmwave to pass but yield different results for lidar signals. Secondly, SHENRON only considers direct reflections of radar signals. In the future, we can use splat-based techniques to perform ray-tracing directly on lidar point clouds and even capture multipath reflections.

REFERENCES

[1] S. Vishwakarma, W. Li, C. Tang, K. Woodbridge, R. Adve, and K. Chetty, "SimHumalator: An open-source end-to-end radar simulator for human activity recognition," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 37, no. 3, pp. 6–22, Mar. 2022.

[2] U. Chipengo, P. M. Krenz, and S. Carpenter, "From antenna design to high fidelity, full physics automotive radar sensor corner case simulation," *Modelling Simul. Eng.*, vol. 2018, 2018, Art no. 4239725.

[3] G. Skidmore, T. Chawla, and G. Bedrosian, "Combining physical optics and method of equivalent currents to create unique near-fieldpropagation and scattering technique for automotive radar applications," in *Proc. IEEE Int. Conf. Microw. Antennas Commun. Electron. Syst.*, 2019, pp. 1–6.

[4] A. Kadian et al., "Sim2Real predictivity: Does evaluation in simulation predict real-world performance?," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6670–6677, Oct. 2020.

[5] A. P. Sligar, "Machine learning-based radar perception for autonomous vehicles using full physics simulation," *IEEE Access*, vol. 8, pp. 5'1470–51476, 2020.

[6] R. Degen, H. Ott, F. Overath, C. Schyr, M. Leijon, and M. Ruschitzka, "Methodical approach to the development of a radar sensor model for the detection of urban traffic participants using a virtualreality engine," *J. Transp. Technol.*, vol. 11, pp. 179–195, 2011.

[7] V. Degli-Esposti, F. Fuschini, E. M. Vitucci, and G. Falciasecca, "Measurement and modelling of scattering from buildings," *IEEE Trans. Antennas Propag.*, vol. 55, no. 1, pp. 143–153, Jan. 2007.

[8] S. O. Wald and F. Weinmann, "Ray tracing for range-doppler simulation of 77 GHz automotive scenarios," in *Proc. IEEE 13th Eur. Conf. Antennas Propag.*, 2019, pp. 1–4.

[9] N. Hirsenkorn et al., "A ray launching approach for modeling an FMCW radar system," in *Proc. IEEE 18th Int. Radar Symp.*, 2017, pp. 1–10.

[10] J. Thieling, S. Frese, and J. Roßmann, "Scalable and physical radar sensor simulation for interacting digital twins," *IEEE Sensors J.*, vol. 21, no. 3, pp. 3184–3192, Feb. 2021.

[11] X. Li et al., "Research on millimeter wave radar simulation model for intelligent vehicle," *Int. J. Automot. Technol.*, vol. 21, no. 2, pp. 275–284, 2020.

[12] T. Machida and T. Owaki, "Rapid and precise millimeter-wave radar simulation for ADAS virtual assessment," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 431–436.

[13] C. Schüßler, M. Hoffmann, J. Bräunig, I. Ullmann, R. Ebelt, and M. Vossiek, "A realistic radar ray tracing simulator for large MIMO-arrays in automotive environments," *IEEE J. Microw.*, vol. 1, no. 4, pp. 962–974, Oct. 2021.

[14] L. Wang, B. Goldluecke, and C. Anklam, "L2R GAN: Lidar-to-radar translation," in *Proc. Asian Conf. Comput. Vis.*, 2020. [Online]. Available: https://openaccess.thecvf.com/content/ACCV2020/html/Wang_L2R_GAN_LiDAR-to-Radar_Translation_ACCV_2020_paper.html

[15] R. Weston, O. P. Jones, and I. Posner, "There and back again: Learning to simulate radar data for real-world applications," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 12809–12816.

[16] K. Bansal, K. Rungta, S. Zhu, and D. Bharadia, "Pointillism: Accurate 3D bounding box estimation with multi-radars," in *Proc. 18th Conf. Embedded Netw. Sensor Syst.*, 2020, pp. 340–353.

[17] V. Degli-Esposti and H. L. Bertoni, "Evaluation of the role of diffuse scattering in urban microcellular propagation," in *Proc. IEEE VTS 50th Veh. Technol. Conf.*, 1999, pp. 1392–1396.

[18] H. Rohling, "Radar CFAR thresholding in clutter and multiple target situations," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-19, no. 4, pp. 608–621, Jul. 1983.

[19] C. Sakaridis, D. Dai, and L. Van Gool, "ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10765–10775.

[20] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[21] B. Langen, G. Lober, and W. Herzig, "Reflection and transmission behaviour of building materials at 60 GHz," in *Proc. IEEE 5th Int. Symp. Pers Indoor Mobile Radio Commun. Wireless Netw. - Catching Mobile Future*, 1994, pp. 505–509.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, Munich, Germany: Springer, 2015, pp. 234–241.

[23] S. Madani, J. Guan, W. Ahmed, S. Gupta, and H. Hassanieh, "Radatron: Accurate detection using multi-resolution cascaded MIMO radar," in *Proc. 17th Eur. Conf. Comput. Vis.*, Tel Aviv, Israel, 2022, pp. 160–178.

[24] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 1271–1278.