The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3 – 7 June 2024, Lisboa, Portugal

A PRACTICAL DATA GENERATION FRAMEWORK FOR 4D AUTOMOTIVE MIMO RADAR: ENABLING DEEP LEARNING AND RADAR PERFORMANCE ANALYSIS

MANUEL JÜRGENSEN [1,2], MARTIN VOSSIEK [2], AND J. C. FUENTES MICHEL [1]

[1] BMW Group, Munich, Germany, manuel.juergensen@bmw.de, www.bmwgroup.com

[2] Institute of Microwaves and Photonics FAU Erlangen-Nürnberg, Erlangen, Germany, www.lhft.eei.fau.de

Keywords: Radar, Deep Learning, Data-driven Methods, Physics-based Methods

Summary. The application of machine learning in automotive radar systems presents severe challenges. In particular, the availability of raw radar data tailored to specific radar configurations and annotated datasets is limited. In this paper, we propose a novel method to reduce the lack of available datasets and bridge the gap between recorded datasets (which fit one type of radar) and sophisticated ray tracing simulations (which are computationally expensive and based on assumptions). Our approach utilizes recorded features from existing radar systems currently used in production. We specifically accumulate radar detections (as defined in ISO 23150:2023) over multiple cycles to increase the resolution of a given radar and minimize fluctuation in the features obtained. To achieve more representative results, we include clutter in the simulation. The accumulated features, which we will refer to as pseudo-scatter points, are then treated like scatter centres to calculate raw spectra for virtual radar systems with arbitrary antenna arrangements. The following processing steps extract range, velocity and angular information. We compare simulated radar data from a digital twin with measured radar data in an identical scenario to validate our method. Our framework provides a practical and versatile solution for radar data generation, facilitating the development of deep neural networks and the predevelopment of radar systems. Although it may not achieve the level of fidelity provided by ray tracing simulations, we demonstrate that our approach offers a cost-effective method for generating high-quality radar data for advanced driver assistance systems (ADAS) applications.

1 INTRODUCTION

Radar sensors play a major role in the environmental perception of ADAS [1]. Their robustness in adverse weather conditions particularly differentiates the radar from other sensors used in ADAS, such as camera and lidar [2]. Furthermore, radar sensors are cheaper than lidar sensors, making them feasible for a wide range of applications in vehicles. Due to the physical working principle, radar sensors can achieve accurate measurements in range and velocity. However, in ADAS applications, not only range and velocity measurements but also measurement of the angular position in both the azimuth and elevation domains are crucial. In future applications, radars ought to be able to resolve fine details in the environment, such as

vulnerable road users in different situations. Increasing the resolution of radar applications is a field of current research in the ADAS field [3].

Multiple papers have shown that deep learning has great potential in the domain of radar signal processing. In particular, in the case of angular processing, deep neural networks (DNNs) show promising results [4], [5]. DNNs can increase the resolution of different radar sensors using the same hardware and antenna layouts [4]. However, training DNNs relies on labelled data availability. Current datasets on radar data cover only some signal processing steps and sometimes lack annotations [6].

Furthermore, generic radar datasets that can be applicable to different radar systems cannot be retrieved through measurements. This is due to the nature of the radar cube and the information stored within it. Zhou et al. [6] reviewed different radar datasets for diverse types of radar. From an industrial perspective, developing a DNN for detection purposes is reasonable only if the DNN fits the sensor in use. This creates a problem with data collection and labelling. For every new sensor in development, we need to create a new dataset and a new set of labels, which is resource-intensive and time-consuming. A solution to this problem is proposed with a ray tracing simulation based on synthetic environments, in which Schuessler et al. [7] created realistic results in their radar data simulations. However, this approach requires creating artificial scenarios or using pregenerated digital worlds, each presenting its challenges related to realism. Ensuring realistic scenarios and a broad mix of object and scenario types is critical, as is addressing the challenges of simulating radar cross-section (RCS) [6].

In this paper, we present a method for generating datasets that are partly synthetic and partly data-driven. We use recorded detections from a production high-resolution (HR) radar sensor and accumulate them over consecutive radar cycles to increase the resolution. For our raw radar spectrum simulation, we treat each accumulated detection as a scatter centre and generate a sine wave in the intermediate frequency (IF) domain. Our signal processing framework then processes the raw spectrum, computes range and velocity fast Fourier transform (FFT), performs beamforming, and determines the local maxima through a constant false alarm rate (CFAR) algorithm. In this way, we generate every step of the radar data cube that is of interest for radar performance analysis. Our framework's advantages lie in the practicability and versatility of the approach. According to our findings, our approach has proven to be economical in the predevelopment of radar and in generating data for possible use in deep learning applications.

2 4D MIMO RADAR DATA GENERATION FRAMEWORK

The availability of raw radar data is an issue regarding DNN development in the field of automotive radar [6]. To solve this issue, we propose an adaptive framework for radar data generation. Our goal is to build a framework that is as flexible as possible while keeping costs low. Parameters such as antenna configuration, transmitting power, gain and analogue digital converter (ADC) sampling rate should be definable. The frequency modulated continuous wave (FMCW) waveform should be adjustable with carrier frequency, bandwidth, chirp period and slope parameters.

Furthermore, signal processing steps, such as windowing or CFAR techniques, should be configurable. Modularity is also the goal of our data-generation framework. Every step in signal processing should be made available as an output. To approach the problem, we use already

collected radar detections from our test vehicles with an HR radar as input for the simulation. By detection, we refer to the thresholded local maxima of the processed radar spectrum. Detections are defined as points where the radar system identifies targets in the environment. Our approach treats the recorded and accumulated detections as scatter points from the environment, which we call pseudo-scatter points. This method includes an accumulation step to enhance resolution, minimize environmental clutter and isolate dominant scatter points. We accumulate these detections over three radar cycles. To compensate for the car's and the environment's movements, we need to correct the positions of the detections between the radar frames. We create raw data streams for each antenna with the collected pseudo-scatter centres. Next, a signal processing chain follows, which consists of range, velocity and angle measurement. The focus lies on replicating the radar cube. Our approach includes azimuth and elevation measurements.

2.1 Detection Accumulation and Correction

With the collected HR radar detections, we build the pseudo-scatter point environments. The input to our simulation needs to resemble the physical scatter points, which is why we accumulate detections from our production HR radar over three frames to better extract scattering behaviour. Our method is inspired by Abadpour et al. [8], who measured the backscattering behaviour of vulnerable road users from all possible incident angles. Our approach relies on changing incident angles through changes in the scenario.

Note that our approach will only work for short timesteps < 1 s because we only measure the backscattering behaviour in a short period of time. Longer periods will change the backscattering behaviour due to the movements of the ego vehicle and its surroundings.

Due to the movement of both the environment and the test vehicle, correcting these effects during data accumulation is crucial. We achieve this by calculating the displacement by integrating the velocities. We found that for static targets, it works best to use the vehicle's velocity for correction. For dynamic targets, the relative velocity of the detections is more suitable. Hence, we introduce a static velocity threshold, which can be fine-tuned to achieve the best results. To correct the displacement, we first need to convert the relative velocities from polar coordinates into a cartesian. Figure 1 shows the correction of moving detections in the global coordinate system and the correction of static detections in the local coordinate system.

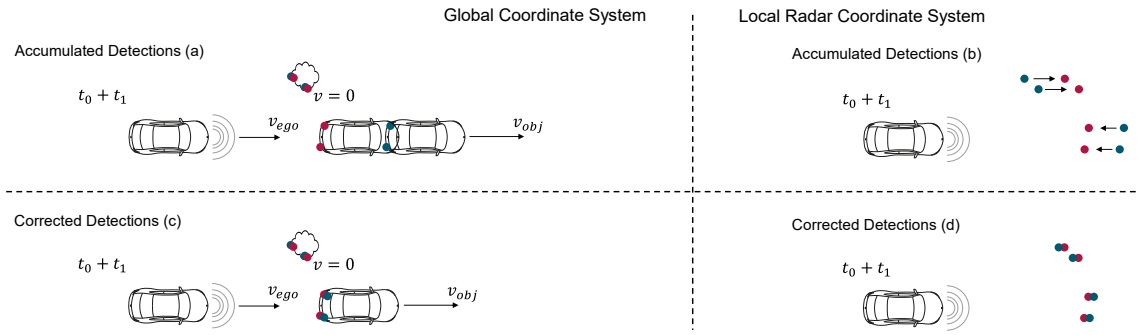

Figure 1: Detection accumulation example scenario for two timesteps t_0 and t_1. (a) and (b) show the accumulated detections for the two timesteps in different coordinate systems. Note that due to ego velocity and object velocity, the accumulated detections do not match. (c) and (d) show the correction step, where the positions of the detections are corrected in both the global and local coordinate systems.

As shown in Figure 1, we retrieve an error if we accumulate the detections over time. Our approach to correct the displacement is integrating the given velocity from the car, ego velocity v_{ego}, for static detections. We correct the displacement observed in dynamic detections by integrating the radial velocities obtained from the measurements. For short timesteps, we assume that the change in z is negligible. To obtain the yaw angle ψ, we integrate the yaw rate over time. The displacements are calculated as follows:

$$x_{n+1} = v_{ego} \cdot \sin(\psi) \cdot \Delta t_n + x_n \tag{1}$$

$$y_{n+1} = v_{ego} \cdot \cos(\psi) \cdot \Delta t_n + x_n \tag{2}$$

Testing shows that it is useful to distinguish between moving and stationary detections. We introduce a threshold to incorporate this in our correction step. To compensate for moving detections, we use the following equation:

$$\overline{x_{corr}}(t) = \left(\bar{x}(t) - \overline{x_{det}}(t)\right) \cdot \frac{1}{|\bar{x}(t) - \overline{x_{det}}(t)|} \cdot \overline{v_{rel}}(t) \cdot \Delta t, \tag{3}$$

where x_{corr} denotes the correction vector, x denotes the radar position and x_{det} is the detection position. Essentially, we build the direction vector between the radar and detection positions and normalize it. We then multiply with the relative velocity v_{rel} and perform a discrete integration over the frame period Δt. After the correction step, we again apply the velocities to each scatter point to animate the scenario. For the spectra calculation, we calculate a new position for every radar chirp from the current position of the scatter point, which changes with every chirp due to the relative velocity.

2.2 Modelling the Environment Clutter

The detection building step in radar signal processing distinguishes between environmental noise and clutter and the present signal in the spectrum. A thresholding step removes the clutter and noise from the signal. This is why our method does not include all clutter and noise in the environmental measurements. To address this issue, we model clutter in spectrum generation. We use randomly distributed scatter points to model clutter in our simulation [7], [9]. We represent the clutter using a two-dimensional (2D) map and model the average occurrence of clutter targets by employing a Poisson distribution with a lambda (λ) value of 0.62, as referenced in [9]. The position distribution is modelled by a uniform distribution [9]. To statistically model the RCS value, we use the Rayleigh distribution with an estimated sigma (σ) from our measurements of the value -40 dBm [10]. We assume that clutter appears in the field of view and with a position between z = ±10 cm Gaussian distributed. Modelling clutter can be arbitrarily complex, depending on the scenario and the desired realism [10]. We chose this approach for our simulation as a good trade-off between complexity and accuracy.

2.3 FMCW Radar Model

FMCW radar relies on the time-of-flight principle. A transmitter emits an electromagnetic wave that travels through the environment. Small differences in time-of-flight need to be measured to extract information such as range, velocity or angle of arrival from the received signal. This is achieved by encoding the electromagnetic wave with a frequency-modulated signal [10]. Figure 2 depicts the basic principle of a radar system. It consists of the most relevant

hardware and signal-processing components for the scope of our work. For our FMCW radar signal model, we treat the multiple input multiple output (MIMO) system as a virtual antenna array consisting of N receive antennas and one transmit antenna.

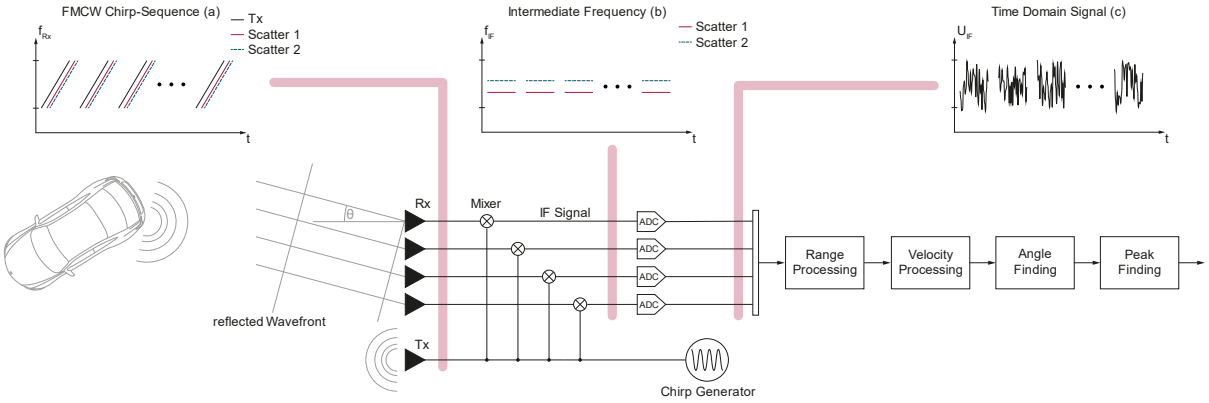

Figure 2: FMCW radar principle with processing chain: The transmitted frequency chirp travels through the environment and is reflected by scatter centres. (a) depicts the transmitted waveform and the received reflections. After the mixer (b), the constant IF signal is sampled by the ADCs. The signal to process is a voltage in the time domain (c). The subsequent signal processing extracts the range, velocity and angular information of the environment. (Figure adapted from [3])

To create the raw spectrum for each antenna from our environment, we model the physical behaviour of the reflected wavefront by modelling the reflected signal after the mixer. With the radar range in Equation (4), we model the relation between emitted power P_t and received power P_r. This is a function of the distance R between the antenna and the scattering centre [10]. The RCS value σ incorporates the reflective behaviour of the target. To reduce complexity in our framework, we use a simplified radar range equation from [10]:

$$P_r = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 R^4}, \tag{4}$$

where λ denotes the wavelength and G is the radar gain. Our measured detections from our test vehicles already include the RCS, eliminating the need to physically simulate the parameter. The basic principle of the signal model is illustrated in Figure 2. An FMCW radar transmits a frequency chirp with a base frequency and bandwidth. The electromagnetic wave is reflected and received at the receiver element. Due to the time of flight, the received frequency chirp shifted slightly in time. For our simulation, we assume that the scatter points are static for each chirp. The mixer multiplies the transmitted signal with the received signal. Figure 2 (b) shows that after the mixer, each scatter point appears as a single constant frequency called the IF signal. Note that the multiplication of sine waves is a subtraction and addition of the present frequencies. A low-pass filter filters out the high-frequency part. For simplicity, we discard the high-frequency part in our simulation [10]. The IF signal is calculated by summating the sinewaves from each scatter point (see Equation (5)). The distance between the transmitting antenna, scatter point and the respective receive antenna must be calculated to build each sinewave. The IF signal of each scatter point can be calculated using the slope of the chirp, which is the quotient of bandwidth and chirp duration, as well as the time-of-flight calculated with lightspeed c. To calculate the received phase $\theta_{t,n}$ at the antenna, a multiple of the

wavelength is used, and a modulo operation is performed. Using the IF signal and phase, we can calculate the spectrum with:

$$\bar{y}(t) = \sum_{n=1}^{N} P_{r,n}(R) \cdot e^{j[2\pi F_{t,n}(t-t_0)+\theta_{t,n}(t-t_0)]} \tag{5}$$

The frequency of the IF signal for every scatter point n is denoted with $F_{t,n}$ calculated from the time of flight and the chirp's slope. The difference between the current time t and the transmit time t_0 calculates the time of flight. With a power $P_{r,n}$ depending on the distance R, the spectrum is calculated for each receiving element and chirp, incorporating every scatter point with properties such as RCS, position and velocity. We build the radar cube if we order the complex samples in a fast, slow and spatial dimension. With so-arranged samples, we can further perform signal processing.

2.4 MIMO Radar Signal Processing

Signal processing aims to extract information, such as range, velocity and angle of arrival, from the raw signals. Figure 3 shows a typical signal processing chain from the sampled IF signal to the detections after thresholding. The different steps serve different purposes and extract different types of information. For simplification, we calculate the received signals directly for each virtual antenna. It is outside the scope of this work to simulate the MIMO demodulation step. As stated by Scheer and Melvin [11], this simplification is possible. To build the virtual receiver array in a 2D plane, we add both the x and z positions of the real transmitting and receiving antenna for each antenna pair [11]. The resulting virtual array consists of m x n virtual receiving antennas and one transmitting antenna. This simplification holds only if the far-field assumption is valid [11]. In the next step, an ADC converts the received signal into a digital and discrete form that can be processed further. First, the range FFT converts the received signal into interpretable range information by extracting all present frequencies in the IF signal f_{IF} with an FFT. The IF signal after the mixer for every scatter centre is approximately continuous over one chirp. With the continuous characteristic, the frequencies are transformed into ranges using the following formula:

$$f_{IF} = \frac{2\,S\,d}{c}, \tag{6}$$

where S is the slope of the chirp and c represents the lightspeed. The range or distance d is the value of interest. A Doppler FFT over a batch of chirps extracts the different relative velocities present in the environment by analysing the differences in the phase of the signal for each range bin. The resolution can be determined depending on the length of the window. The cycle of one chirp determines the maximum unambiguous velocity that can be obtained. If the period of the chirps is long, the maximum unambiguous velocity is small. This means that higher velocities appear as alias frequencies. The combination of range and velocity information is called a range-Doppler map. It is depicted in Figure 6. Range is displayed on one axis, and relative velocity on the other axis. The z value or the colour intensity depicts the magnitude of the signal for a certain range and Doppler bin. With the range-Doppler map, we perform the first thresholding to extract targets. Peaks in magnitude in the range-Doppler map can be interpreted as reflections from the environment. One characteristic of the range-Doppler map is that it has a line at the negative ego velocity for all static objects.

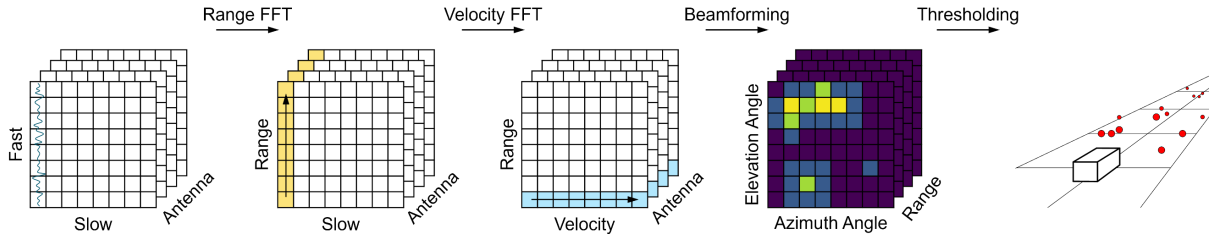

Figure 3: Radar signal processing chain: The raw spectrum contains range, velocity and angular information. The goal of signal processing is to efficiently extract this information. From left to right, the relevant information is extracted using FFTs and beamforming and compressed using CFAR and maximum thresholding.

To further process the range-Doppler map, we average the received spectra for all antennas to increase the signal-to-noise ratio. A cell averaging CFAR (CA-CFAR) extracts peaks in the signal. CA-CFAR compares a cell of interest with the average of the cells around. If the cells around the cell of interest exceed a certain threshold, the cell is marked as a peak. The CA-CFAR algorithm chooses the range bins to further process in the angular dimension [10]. To do so, 2D beamforming for each range bin resolves the incident angle in the elevation and azimuth directions. This results in a four-dimensional (4D) representation of the received spectra in range, velocity, azimuth and elevation angles. Beamforming is a matrix multiplication between a vector of values of interest extracted from the range-Doppler map and each angle of interest – in our case, the complete field of view and all antenna channels [11]. To further reduce the signal size for more efficient computing, a state-of-the-art approach is to reduce the spectrum to detections, which is the last step of our proposed framework. We perform maximum thresholding for every azimuth-elevation map to extract the dominant reflectors.

2.5 Data Collection

From our fleet, we collect data from different scenarios and weather conditions. Our goal is to have a well-distributed collection of real driving scenarios, including, for example, highways and rural roads. We focus on data collection in challenging scenarios for radar sensors. In our case, this includes bridges or tunnels. In future work, we will focus on quantifying the distribution of different objects and scenario types.

3 EVALUATION

To evaluate our raw radar data generation method, we compare a simulated radar data cube from a digital twin of a radar evaluation board with the measured radar data cube. We are particularly interested in the range and velocity signal, which is the native strength of the radar. We performed the same scenario twice to collect comparable measurement data for our series production HR radar and the evaluation radar. This approach ensures that the production radar does not interfere with the evaluation radar.

3.1 Comparison of Synthetic and Real Radar Cube

To validate our method, we compare a radar data cube from measurement with a data cube from our simulation for the same scenario. To reduce the complexity of validation, we focus on a simple scenario with a moving and a static target. Figure 4 depicts the test scenario (a) and

the measurement setup (b, c). For our validation, we ensured that our evaluation radar was in the same position as the HR radar. To rule out interference, we recorded the scenario two times: once with the evaluation radar and once with our production sensor. The target vehicle drove at the same speed and position in both scenarios. The corner reflector and the measurement vehicle were stationary. At the time of the evaluation, the vehicle was in the range of approximately 12 m from the radar, with a speed of 10 kph. The corner reflector is at a range of 8 m.

(a) (b) (c)

Figure 4: Scenario (a) and measurement setup (b, c): The target vehicle drives at 10 kph at a distance of 12 m during the measurement. The corner reflector is positioned at a distance of 8 m in front of the measurement vehicle. Note that the evaluation sensor is positioned in the same position as our HR radar.

We created a digital twin with the same specifications as the evaluation board for the evaluation. We model parameters such as antenna layout, bandwidth, slope and carrier frequency. The simulation then calculates the spectrum from the pseudo-scatter centres and performs a range FFT and a Doppler FFT. We observe the same frame for measurement and simulation for comparison purposes.

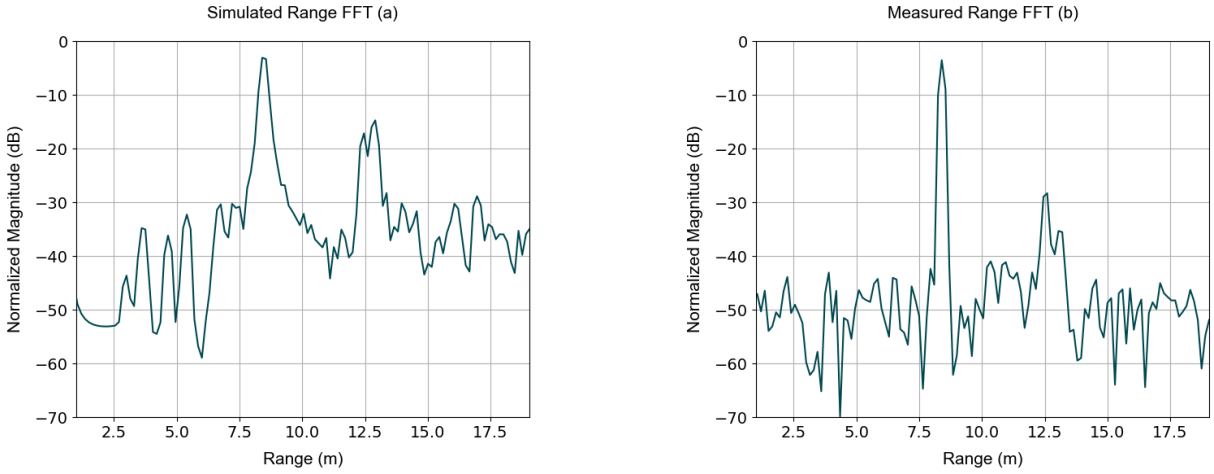

Figure 5: Comparison of simulation (a) and measurement (b) from the test scenario in the range domain. For the simulation, we created a digital twin of our evaluation radar. Note that the simulation resembles the measurement very well, except for the magnitude of the signal.

Figure 5 depicts the range-magnitude plot after the range FFT of our simulation (a) and the measurement (b) from the same scenario. We observe that our simulation is similar to the measurement. The range peak at the position of the corner reflector and the car are distinguishable from the clutter. However, the graphs show that the ratio between the magnitude of the corner reflector and the car is not reproduced accurately. Section 2.3 states that we use a simplified radar equation that does not incorporate atmospheric loss that is range dependent [10]. Adding atmospheric loss to the radar equation could lead to better results.

Furthermore, the graphs show that clutter can be modelled using the method described by [9]. The graphs in Figure 6 show the range-Doppler map after the Doppler FFT where the targets appear in high magnitude at certain ranges and velocities. Note that our evaluation sensor could not generate fast enough chirps, leading to limited maximum velocities. This is why the 10 kph of the car appears at a velocity of -1 m/s as an alias frequency. This shows that our framework can correctly simulate aliases in velocity. In addition, one can observe that the range-Doppler map also shows a difference in magnitude between the simulation and the measurement. We will need to investigate this behaviour further to reduce the gap between our simulation and the measurement.

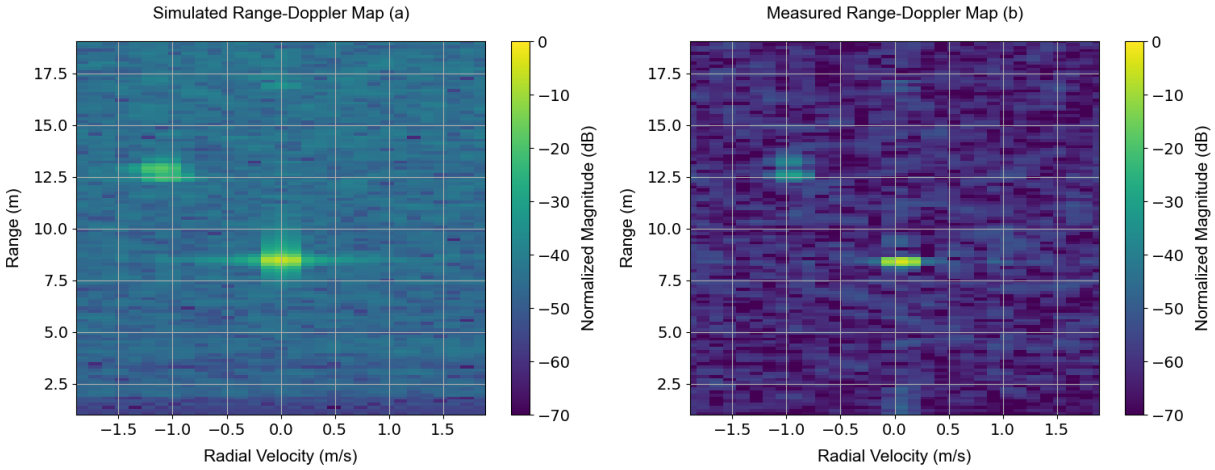

Figure 6: Comparison of simulation (a) and measurement (b) of the test scenario with range Doppler plots. For comparison, we use a digital twin of the evaluation radar. It can be observed that the simulation depicts the measurement very well, except for the magnitude of the signal and the clutter.

Our data show that we still need more precision in the detection step, which will require further investigation. Our current approach with a CA-CFAR in the range-Doppler domain and maximum thresholding in the azimuth-elevation domain does not yield sufficient results. Although the CA-CFAR in the range-Doppler domain works well, we need to see whether enough targets are detected in this step. For angle finding, we need to enhance our maximum finding further.

4 RESULTS

Our research shows that generating raw radar data for different radar setups from pre-recorded scenarios is possible. Figure 7 illustrates the processing results based on a recorded scenario and pseudo-scatter points. Our approach treats detections from our HR production

radar as pseudo-scatter centres and includes environment clutter to create a more realistic representation of the environment. Due to the modular approach, every step in the signal processing chain, from the ADC measurement of the IF signal up to the detection level, can be chosen for the dataset. Every parameter of an FMCW radar system can be changed and adapted with our approach. Our framework is designed so that every 2D antenna layout can be computed. However, our method has a few limitations. Due to the measurement step's limited resolution, the method works for radar systems with a similar resolution or lower resolution compared to our HR production radar. We use a simplified model for the clutter simulation, which could be extended further. This needs to be kept in mind when assessing the performance of radar sensors or training neural networks using our method. Our method applies particularly to automotive 4D MIMO radar systems. Other systems might require other approaches. Our approach directly calculates the signals received in virtual channels. With this, we neglect the MIMO demodulation, leading to errors. In the current implementation, the magnitude of the signal and the clutter do not fit the measured magnitude. Further investigation is needed to implement this correctly. The detections are measured with a 77 GHz radar system, which limits the application to similar virtual systems – it can be said that the further the carrier frequency varies, the greater the error. Further investigation is needed to quantify this error. Our method aims at regression learning for DNNs; we do not include labels such as pedestrians or vehicles.

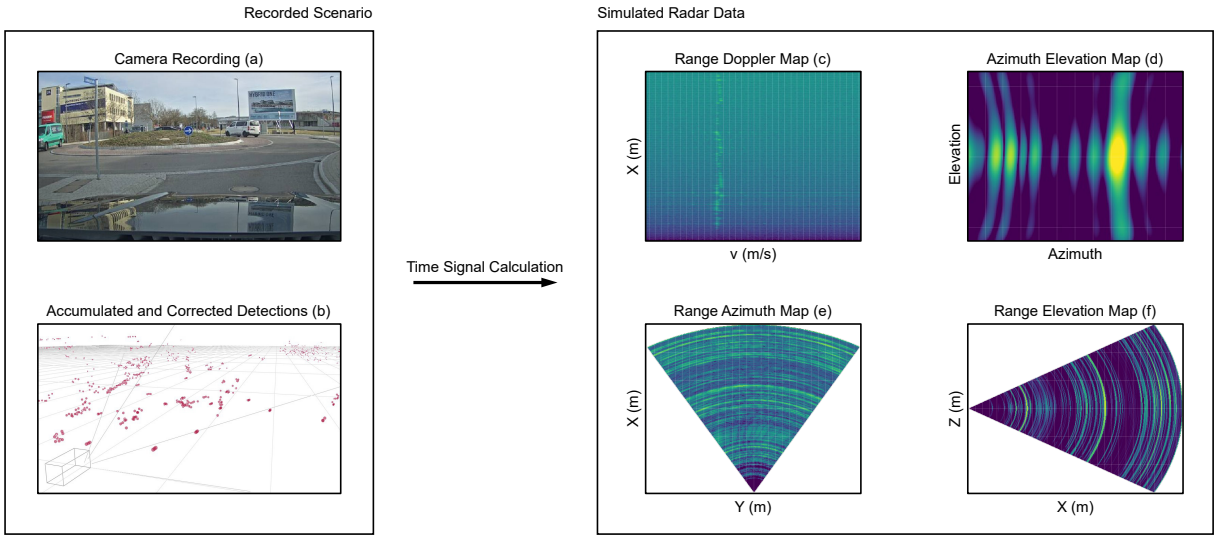

Figure 7: Sample data from our method: (a) and (b) show a recorded scenario from which we simulate the radar data. The accumulated and corrected detections for this scenario are depicted in (b). With our method, it is possible to simulate radar cubes in different situations with arbitrary antenna configurations. Different processing steps can be used for further processing.

With the use of graphics processing unit (GPU) accelerated computing, we achieve a speed of 1 min calculation per frame for a reference radar with 192 virtual antennas and a radar cube size of 2,000 × 256 × 192 in range, velocity and angle, respectively. A completely new dataset can be generated in a week, consisting of 10,000 frames. For our research, we use the NVIDIA® GeForce RTX™ 4090 and the Cupy library for Python [12]. Currently, we use GPU acceleration only for IF signal generation. We are confident that there could be more performance gains in applying GPU acceleration to the subsequent signal processing steps,

which would further increase performance.

Our approach fills the gap between predefined datasets from measurements that fit only one radar specification and sophisticated ray tracing frameworks. It is a fast and cheap approach to the data problem radar systems face. For radar performance assessment, different mid-resolution radars can be compared in different edge scenarios to see whether the radar of choice can handle certain relevant cases in the early stage of development. Our method shows that it is possible to produce a realistic radar cube with a practical approach.

5 CONCLUSIONS

The scarcity of raw radar data tailored to various radar configurations motivated this research. Our objective was to develop a practical and realistic solution to that problem. The proposed research demonstrates the feasibility of creating radar data cubes from recorded scenarios in a cost-effective way. This approach closes the gap between solely recorded datasets and simulation-based approaches, such as ray tracing. Future research will aim to improve the accuracy and realism of synthetic data. One potential enhancement could be to add a more sophisticated noise and clutter model.

Our validation shows that improvements are needed in the signal's magnitude. We aim to incorporate this through a more accurate implementation of the radar equation and a better representation of the radar in the digital domain. To further validate our framework, we plan to compare different signal processing steps from a digital twin and the measurements from a radar, which we aim to reproduce. Another focus area is the validation of the simulation's detections against those from the production radar. We will use this framework to assess radar sensors across different scenarios of interest. We further plan to assess the feasibility of the framework in other basebands and quantify the associated errors. Additionally, we will investigate the feasibility of using our framework to simulate HR imaging radars.

We plan to use our framework for DNN training to assess its capabilities in this domain. We will test whether our method can increase the number of training samples or be used for network pretraining. More applications for our framework include augmenting current labelled datasets with raw data and evaluating different radar sensors.

Further work is required to ensure a balanced distribution of scenarios, weather conditions and classes in our dataset. We also plan to explore radar edge cases that have proven challenging in the past and include them in the dataset.

Overall, we aim to refine our framework's capabilities to enhance its value for the automotive radar community. By providing radar raw data across various scenarios and radar architectures with ground truth, our framework can be used for benchmarking signal processing techniques, radar sensors and DNNs. Expanding our framework to include other waveforms could further enhance its value. Implementing a distribution metric to vary the occurrences of attributes such as bridges, pedestrians or road types could be beneficial for different applications.

REFERENCES

[1] C. Waldschmidt, J. Hasch, and W. Menzel, 'Automotive Radar — From First Efforts to Future Systems', *IEEE J. Microw.*, vol. 1, no. 1, pp. 135–148, Jan. 2021, doi: 10.1109/JMW.2020.3033616.

[2] S. Sun, A. P. Petropulu, and H. V. Poor, 'MIMO Radar for Advanced Driver-Assistance Systems and Autonomous Driving: Advantages and Challenges', *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 98–117, Jul. 2020, doi: 10.1109/MSP.2020.2978507.
[3] J. Fuchs, M. Gardill, M. Lubke, A. Dubey, and F. Lurz, 'A Machine Learning Perspective on Automotive Radar Direction of Arrival Estimation', *IEEE Access*, vol. 10, pp. 6775–6797, 2022, doi: 10.1109/ACCESS.2022.3141587.
[4] C. Schuessler, M. Hoffmann, and M. Vossiek, 'Super-Resolution Radar Imaging with Sparse Arrays Using a Deep Neural Network Trained with Enhanced Virtual Data', *IEEE J. Microw.*, vol. 3, no. 3, pp. 980–993, Jul. 2023, doi: 10.1109/JMW.2023.3285610.
[5] Y. Wang, Z. Jiang, Y. Li, J.-N. Hwang, G. Xing, and H. Liu, 'RODNet: A Real-Time Radar Object Detection Network Cross-Supervised by Camera-Radar Fused Object 3D Localization', *IEEE J. Sel. Top. Signal Process.*, vol. 15, no. 4, pp. 954–967, Jun. 2021, doi: 10.1109/JSTSP.2021.3058895.
[6] Y. Zhou, L. Liu, H. Zhao, M. López-Benítez, L. Yu, and Y. Yue, 'Towards Deep Radar Perception for Autonomous Driving: Datasets, Methods, and Challenges', *Sensors*, vol. 22, no. 11, p. 4208, May 2022, doi: 10.3390/s22114208.
[7] C. Schuessler, M. Hoffmann, J. Braeunig, 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, doi: 10.1109/JMW.2021.3104722.
[8] S. Abadpour, A. Diewald, S. Marahrens, M. Pauli, and T. Zwick, 'Back Scattering of Traffic Participants Based on an Automotive Radar Measurement', in *2020 17th European Radar Conference (EuRAD)*, Utrecht, Netherlands: IEEE, Jan. 2021, pp. 99–102. doi: 10.1109/EuRAD48048.2021.00036.
[9] M. Bühren and B. Yang, 'Simulation of Automotive Radar Target Lists considering Clutter and Limited Resolution', 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:36750367
[10] M. A. Richards, *Fundamentals of Radar Signal Processing*, Second edition. New York: McGraw-Hill Education, 2014.
[11] J. Scheer and W. L. Melvin, Eds., *Principles of Modern Radar - Advanced Techniques*. Raleigh, NC: SciTech Pub, 2010.
[12] R. Okuta, Y. Unno, D. Nishino, S. Hido, and C. Loomis, 'CuPy: A NumPy-Compatible Library for NVIDIA GPU Calculations', in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017. [Online]. Available: http://learningsys.org/nips17/assets/papers/paper_16.pdf