Achieving Efficient and Realistic Full-Radar Simulations and Automatic Data Annotation by exploiting Ray Meta Data of a Radar Ray Tracing Simulator

Christian Schüßler[#1*], Marcel Hoffmann[#*], Vanessa Wirth[§*], Björn Eskofier[$*], Tim Weyrich[†*], Marc Stamminger[§*], Martin Vossiek[#*]

[#]Institute of Microwaves and Photonics
[§]Chair of Visual Computing
[†]Professor of Digital Reality
[$]Chair of Machine Learning and Data Analytics
[*]Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
[1]christian.schuessler@fau.de

***Abstract*— In this work a novel radar simulation concept is introduced that allows to simulate realistic radar data for Range, Doppler, and for arbitrary antenna positions in an efficient way. Further, it makes it possible to automatically annotate the simulated radar signal by allowing to decompose it into different parts. This approach makes not only almost perfect annotations possible, but also allows the annotation for exotic effects, such as multi-path effects or to label signal parts originating from different parts of an object. This is possible by adapting the computation process of a monte carlo shooting and bouncing rays (SBR) simulator. By considering the hits of each simulated ray, various meta data can be stored such as hit position, mesh pointer, object IDs, and many more. This collected meta data can then be utilized to predict the change of path lengths introduced by object motion to obtain Doppler information or to apply specific ray filter rules to obtain radar signals that only fulfil specific conditions, such as multiple bounces or containing specific object IDs. Using this approach perfect and otherwise almost impossible annotations schemes can be realized.**

***Keywords*— Automotive Radar, Radar Simulation, Data Annotation**

I. Introduction

Radar sensors have become one of the most important type of sensors when it comes to automated or autonomous driving applications. Compared to lidar sensors, they are cheap, robust and can also operate under various weather conditions such as rain and fog [1]. Furthermore, radar sensors can directly measure the radial speed of objects by utilizing the Doppler effect. Since radar sensors are especially unobtrusive and compact they can even be employed in domains such as human activity recognition [2], [3] or for medical tasks [4].

In all of these domains, detection and classification for various cases plays a major role. For example, entities in automotive scenarios such as pedestrian, cyclist, and cars have to be detected and classified reliably [5]–[7]. For activity recognition, various tasks such as hand gesture recognition [8] or breathing and fall detection exist [9]–[11]. Also the detection of ghost targets caused by multi-path reflections is important in all of these applications, since it can lead to false detections and classifications. Various machine learning approaches exists to alleviate this effect, see [12]–[14].

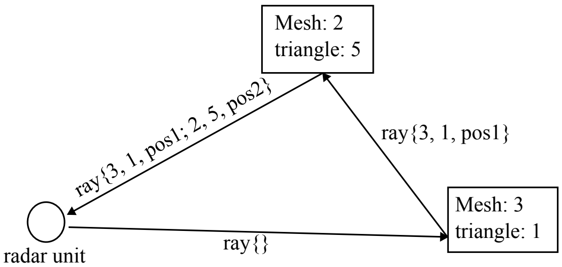

Fig. 1. A launched ray collects meta data in the simulation process, which can be used to generate Doppler simulations and efficiently annotate radar images in the post processing step. In this illustration, the meta data consists of a a mesh id, a triangle index of the mesh, and the ray hit position.

In order to train potential classifiers, commonly the data has to be annotated beforehand. This can either happen manually or self-supervised using reference sensors, such as lidar or camera sensors [15], [16]. However, conducting real measurements and annotating data manually is expensive, time consuming, and often error-prone. Compared to natural images, radar data is very unfamiliar for the the human eye and expert knowledge is required for manual annotation. Even with the help of reference sensors, the problems of generating enough corner cases remains. Further, neither lidar nor camera sensors share the same physical and data processing principles making self-supervised learning not completely reliable.

To overcome this problem, a digital twin of the real world and the sensor itself can be created [17]. This digital reality can serve as ground truth and allows for an improved annotation. Even the car manufacturer Tesla, Inc. created a digital resemblance of San Francisco to train its autonomous driving algorithms[1], which underlines the importance of this topic.

For radar data, several simulation approaches exist, starting from point-scattering models [18], generative machine learning approaches [19] to very accurate ray tracing based physical models [20]–[23]. Especially physical models can obtain very

[1] https://www.teslaacessories.com/blogs/news/virtual-san-francisco-tesla-tests-autopilot-with-simulation-from-large-game-engine-unreal

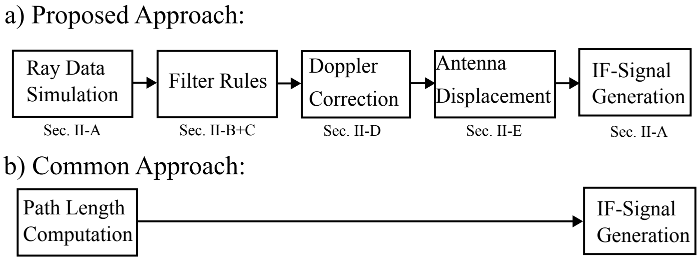

Fig. 2. The contribution of this work this work is shown in the upper part of the image (a). Compared to convential approaches (b) path lengths and ray data is only simulated once and are adjusted afterwards for different chirps and antenna positions. Below each box the respective section is referenced, which explains the processing step in more detail.

versatile and accurate simulations and are successfully used in classification tasks [24]. However, even in simulations it is hard to determine which part of the signal belongs to each specific object. This is because the reflectance at an object also causes sidelobes or creates multi-path targets beside the correct signal part. In the examplary case of a walking pedestrian obscured by a car, a ghost target can be created in the radar signal due to multi-path effects. For safety considerations it is essential to detect that this signal actually originates from a human and at the same time that it is caused by multi-path effects. Another simpler example is that even in simulations an object may be occluded by another one and therefore is not visible in the signal at all. The simulator should therefore be able to detect this effect in order to not annotate this part in the radar image wrongly.

In this contribution, a concept is proposed that solves theses issues by collecting ray meta data during the simulation process with a radar ray tracing simulator. These meta data include pointers to the meshes and triangles that the rays hit on their way from transmitter to receiver. More over, the meta data can be augmented by even more attributes such as normal vectors or material information. This process is illustrated in Fig. 1.

With this meta data, it is possible to efficiently simulate Doppler characteristics as well as data for large antenna arrays. It is further possible to automatically generate (almost) perfect and precise labels for an annotation of the simulated radar signal.

II. Concept

In this section, the complete simulation workflow is explained from the radar signal model, to the utilization of ray meta data up to the efficient simulation for different antenna positions, as also depicted in Fig. 2.

A. Radar Simulation And Signal Model

The contribution of this paper is based on the simulator we presented in [23]. It features an implementation of the shooting and bouncing rays (SBR) approach and uses a probabilistic material model to account for specular and more diffuse reflection types. In order to account for realistic multi-path simulations and to achieve realistic material behavior, a quite large amount of rays is shot into the simulation environment. This is computational expensive but tolerable with modern hardware and standard for visual ray tracing approaches [25] [26].

Conventionally, each ray with index i sums up its path length d_i from the transmit (TX) antenna, to one or more objects and finally the receiving (RX) antenna. This process is also illustrated in Fig. 1. The delay τ_i for each ray is defined by

$$\tau_i = \frac{d_i(j)}{c}, \tag{1}$$

with c is being speed of light. Due to the large amount of rays, this information is sufficient to compute a good representation of the beat or intermediate frequency (IF) signal of a frequency modulated continuous wave (FMCW) radar signal as shown in the equation below

$$s_{\mathrm{IF}}(t,j) = \sum_{i=0}^{N} A(\alpha,\beta)\exp(2\pi j(\mu t\tau_i(j) + f_c\tau_i(j))). \tag{2}$$

With $A(\alpha,\beta)$ modelling the direction-depending antenna radiation pattern and f_c representing the carrier frequency of the signal. The frequency slope μ is defined by the ratio of the bandwidth B and the chirp duration T_c, see equation below

$$\mu = \frac{B}{T_c}. \tag{3}$$

A more detailed description of FMCW radar signal processing can for example be found in [27].

In order to simulate multiple Doppler shifts, so called *chirps*, have to be simulated. The time between two chirps is given by T_d and is often close to T_c but not necessarily equal. Each chirp simulation is indicated by the index j as used in (1) and (2). If an object is in motion during the radar measurement, the path length and therefore also τ will slightly change during each chirp. The time variable t of a single chirp is commonly named *fast time* and the time variable across all chirps with index j is called *slow time*. A complete measurement consisting of several chirps is called *chirp sequence*.

Ordering the chirp sequence into a two dimensional array and computing a discrete Fourier transform results in a Range-Doppler spectrum

$$S_{\mathrm{IF}}(f_r, f_d) = \mathrm{DFT}_{\mathrm{2d}}(s_{\mathrm{IF}}(t,j)). \tag{4}$$

Zero padding and windowing is omitted in this short description. The resulting frequency axes represent the range frequency f_r and the Doppler frequency f_d.

B. Ray Meta Data

In this subsection, the extension of the aforementioned simulation approach is described. The simulator will be adapted, so that it not only stores a ray's path length during the simulation, but also relevant additional information, every time it hits a triangle.

In general, the type and amount of meta data is arbitrary and only limited by the computing resources e.g. video memory. However, in this work the meta data consists of a list of tuples. Each list entry represents a single ray hit and each tuple consists of the following elements:

- mesh identfier a_k
- triangle index b_k
- barycentric triangle coordinates u_k and v_k

Whereby, a mesh represents an object consisting of multiple triangles and each triangle consists of three vertices, see Fig. 3.

The index *k* determines the entry position in the list. Since the complete scene consists of several meshes, which again are made up of several triangles, identifiers for both have to be stored. To obtain the exact hit position after the ray tracing process, the barycentric coordinates of the hit position of the triangle have to be stored as well.

It is also possible to store the Cartesian position of the hit position directly, but by using this approach, it simplifies the computation of Doppler information, as shown later in section II-D. Having a unique mesh identifier is also required to decompose or to automatically label objects in the processed radar signal, as described in the next section.

C. Radar Signal Decomposition

Having stored all rays including their meta data, it is now possible to decompose the IF-Signal in the following way:

$$s_{\text{IF}}(t, j) = \sum_{h=0}^{h=M} s_{\text{IF}}^{h}(t, j, \tau_h). \tag{5}$$

Here, h depicts a ray span or region, that consists of several rays generating an IF-Signal part s_{IF}^{h}. Each ray span may be generated by arbitrary filter rules, such as:

- Any list entry that includes the mesh ID of a pedestrian
- Has more than one list entry to account for ghost-targets
- Has only one list entry and a specific mesh ID to account for line of sight detections of a specific object
- ...

With this technique, radar images including only specific properties can be created and labels can be automatically generated by applying very simple binary segmentation techniques in the processed radar image.

D. Efficient Doppler Simulation

Assuming that the motion across the slow time is small enough so that each ray would hit the same triangle at each chirp snapshot, one simulation run is sufficient as long as the initial ray hit positions are stored. Since the position of each animated triangle is known at every specific point in time, the stored hit positions can be updated and new path lengths can be calculated. This is much more efficient than running a complete simulation for each snapshot, because ray tracing requires a lot of expensive collision checks and most traced rays never hit any object or reach the receiver. This idea has already been utilized in a similar way by [28] by considering the objects velocity and computing the Doppler shift directly. An extension of this was proposed in [29] for a image-based ray tracing simulation for multiple bounces. However, this is implemented for an SBR approach and also supports non-linear motion in general. In fact, it still performs virtual snapshots, but since the initial rays were already sampled, the remaining computation time turned out to be negligible.

This principle can be described as follows: after an initial ray sampling, the hit positions for each ray are known as barycentric coordinates u and v. After the time T_d, the positions of each vertex of each animated object are updated. Since the hit positions should be moved by the same transformation as the vertices while still being placed on the triangle, the updated hit position can then the computed by

$$\vec{p}_i = (1 - u - v) \cdot \vec{v}_1 + u \cdot \vec{v}_2 + v \cdot \vec{v}_3, \tag{6}$$

where the vectors $\vec{v}_1$, $\vec{v}_2$, and $\vec{v}_3$ representing the three vertices of the stored triangle and $\vec{p}_i$ is the updated hit position in Cartesian coordinates after T_d. Updating each ray for the selected chirp, new path lengths and a new IF-Signal can be computed for each chirp.

E. Simulating Different Antenna Positions

By assuming that the objects are sufficiently far away and the array aperture is small enough, path lengths for other antennas can be computed directly, as already done in a similar way in our initial work [23] for the displacement of TX antennas. There, it was directly implemented in the ray tracing process, which is less efficient and consumes a significant higher amount of video memory.

Since the ray hit positions include the first and the last hit point, the path lengths caused by antenna displacement can easily be adjusted by the following formula

$$l_i = l_i - |\vec{p}_i^{f} - \vec{x}_{tx}| + |\vec{p}_i^{f} - \vec{x}'_{tx}| - |\vec{p}_i^{l} - \vec{x}_{rx}| + |\vec{p}_i^{l} - \vec{x}'_{rx}|, \tag{7}$$

with $\vec{p}_i^{f}$ being the first hit position and $\vec{p}_i^{l}$ being the last hit position for the simulated TX and RX antenna pair with positions $\vec{x}_{tx}$ and $\vec{x}_{rx}$. The positions for the displaced antennas are denoted as $\vec{x}'_{tx}$ and $\vec{x}'_{rx}$, respectively.

The idea is that, by subtracting the path lengths from the TX antenna to the first hit position and the path length from the RX antenna to the last hit position, only the path lengths originated through the environment remain. Assuming that the antenna array is small enough and objects are sufficiently far away, the reflection behavior of the objects do not change and the path lengths only has to be adjusted for the antenna displacement. This process is illustrated in Fig. 4. This approximation typically holds for automotive applications in the millimeter wave regime.

III. Results

In the first experiment, we simulated a typical Range-Doppler map of a walking pedestrian and compared it with our measurement data. The measured and simulated range-Doppler image as well as a camera image can be seen in Fig. 6. Clearly, a micro-Doppler signature of the pedestrian

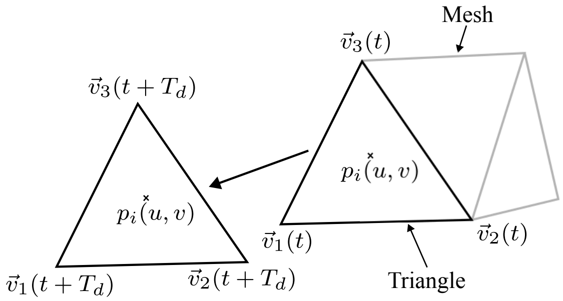

Fig. 3. This scheme shows an example of how a mesh with its assigned triangles moves through time. After T_d a new snapshot has to be taken and the hit position has to be updated. Since hit position experience the same transformation as the hit triangle, a single simulation run is sufficient to compute the barycentric coordinates $p_i(u,v)$ of the hit position. For all subsequent chirps, only the vertices have to be updated to compute the correct hit position.

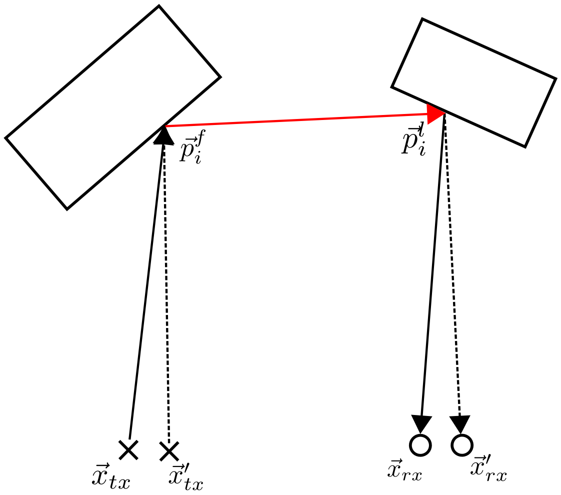

Fig. 4. The process of computing path lengths of multiple antennas without re-simulating the scene is depicted. Only the inner path length (red) remains equal, path lengths directly connected to the antennas can be adjusted afterwards. This depiction is not to scale, in typical real measurement scenarios the antennas are much closer and the objects are placed much farer away from the antennas.

can be seen, as highlighted by red ellipses. There also exist multi-path effects leading to ghost targets, which can also quite accurately reproduced by the simulator. However, there are still some differences between simulation and measurement, which might originate from an insufficient 3D description and from a too simple noise simulation.

Since we can manipulate the ray meta data, it is possible to create a simulated range-Doppler image that only shows the ghost targets. As already mentioned, this is especially helpful for artificial intelligence to learn to differentiate between real and ghost targets for automotive applications. This technique is demonstrated in Fig. 7 for different types of multi-path effects. As can be seen in image (c) the Doppler signal from the legs, assuming having the largest Doppler extent is mainly visible indirectly through reflections by the floor, while ghost-targets far away are generated by reflections from the side-walls.

As mentioned before, it is also possible to decompose the signal into different parts of an object. For example, in hand gesture recognition tasks only the signals reflected by hands or arms should be used for further processing. In Fig. 5, it is demonstrated how to separate the signal originated from the arms from the rest of the body. Here, the same pedestrian object and 3D environment was used as before. As can be seen, the Doppler signal that only stems from the arms does overlap with the signal from the rest of the body, but has significantly lower extend in Doppler direction, which is otherwise assumed to be caused by legs.

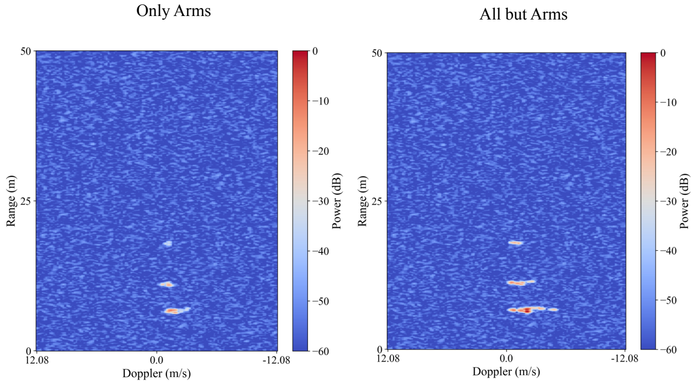

Fig. 5. In the left image, the complete scene was simulated as in Fig. 6, but afterwards only rays that hit the arms of the pedestrian were used for the signal generation. In the right image, it is vice versa everything but the arms were kept during signal generation.

IV. Discussion and Future Work

In this work, a radar simulation approach was presented that is based on a single TX and RX-antenna pair. This is sufficient, to generate simulated data for larger arrays and Doppler information. Consequently, a complete radar cube can be simulated. For objects which are close to the antennas some assumptions may not hold since the reflection behavior of the surfaces is angle-dependent. In future work, also the normal vector of these surfaces shall be stored in the ray meta data, so that the received power for each antenna may be adjusted accordingly. Further, it was shown how the meta data of the simulated rays can be used to decompose the signal according to predefined filter rules. With this technique, simulation data can be automatically labeled in a way that would be mostly impossible by common simulation approaches and also extremely challenging for real measurement data.

Future work might improve the simulation process by creating more diverse 3D worlds and compare the simulation with more complex measurement environments. It should also be further evaluated to which extent simulation data can help to train machine learning algorithms.

Acknowledgment

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Grant SFB 1483–Project-ID 44241933.

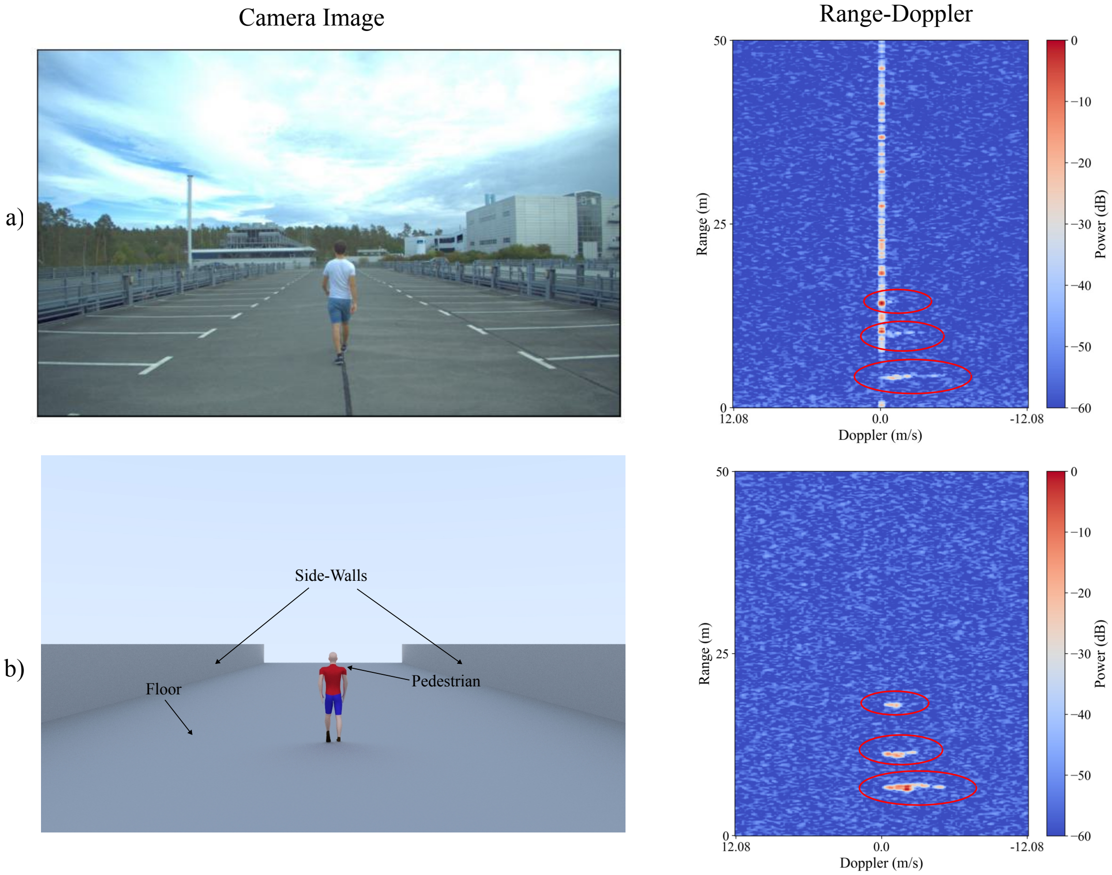

Fig. 6. The real measurement from a walking pedestrian is shown in the upper part of the figure (a) with its simulated counterpart in the lower part of the figure (b). As can be seen, both range-Doppler maps show the same geometrical behavior and two ghost targets originating from multi-path effects are visible. Since the radar signal hits the walls in the simulation environment in a sharp angle, the signal is reflected away from the RX antennas making them invisible in the simulated radar image. The walls in the real measurement are consisting of metal poles and bars, which can directly reflect the signal and are therefore visible. Since we focus mainly on multi-path effects, we did not take the effort to create such a detailed simulation environment.

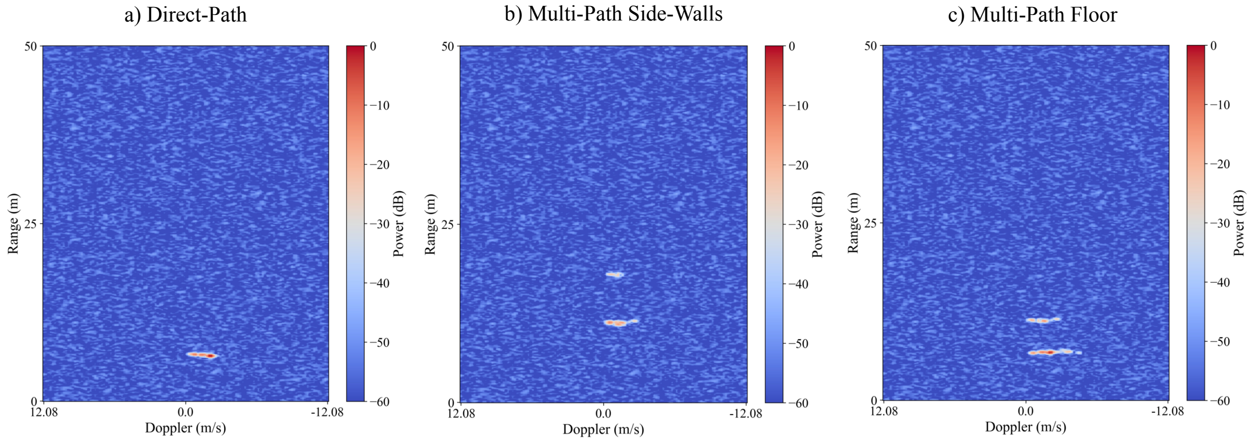

Fig. 7. Ray meta data was filtered in three different ways. In the left image (a) only direct rays without multiple bounces were considered during the signal generation. In the center image (b) rays with at least one bounce hitting the side-walls were used and in the right image (b) multi-path effects including the floor were considered.

The authors would like to thank the Symeo team from indie Semiconductor (Jannis Groh, Javier Martinez, and Mark Christmann) for their support with the radar system used for tests.

References

[1] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, "A systematic review of perception system and simulators for autonomous vehicles research," *Sensors*, vol. 19, no. 3, p. 648, 2019.

[2] S. Z. Gurbuz and M. G. Amin, "Radar-based human-motion recognition with deep learning: Promising applications for indoor monitoring," *IEEE Signal Processing Magazine*, vol. 36, no. 4, pp. 16–28, 2019.

[3] X. Li, Y. He, and X. Jing, "A survey of deep learning-based human activity recognition in radar," *Remote Sensing*, vol. 11, no. 9, p. 1068, 2019.

[4] Y. Liu, G. Zhang, C. G. Tarolli, R. Hristov, S. Jensen-Roberts, E. M. Waddell, T. L. Myers, M. E. Pawlik, J. M. Soto, R. M. Wilson *et al.*, "Monitoring gait at home with radio waves in parkinson's disease: A marker of severity, progression, and medication response," *Science Translational Medicine*, vol. 14, no. 663, p. eadc9669, 2022.

[5] P. Sorowka and H. Rohling, "Pedestrian classification with 24 ghz chirp sequence radar," in *2015 16th International Radar Symposium (IRS)*. IEEE, 2015, pp. 167–173.

[6] A. Angelov, A. Robertson, R. Murray-Smith, and F. Fioranelli, "Practical classification of different moving targets using automotive radar and deep neural networks," *IET Radar, Sonar & Navigation*, vol. 12, no. 10, pp. 1082–1089, 2018.

[7] R. Prophet, M. Hoffmann, M. Vossiek, C. Sturm, A. Ossowska, W. Malik, and U. Lübbert, "Pedestrian classification with a 79 ghz automotive radar sensor," in *2018 19th International Radar Symposium (IRS)*. IEEE, 2018, pp. 1–6.

[8] S. Skaria, A. Al-Hourani, M. Lech, and R. J. Evans, "Hand-gesture recognition using two-antenna doppler radar with deep convolutional neural networks," *IEEE Sensors Journal*, vol. 19, no. 8, pp. 3041–3048, 2019.

[9] F. Adib, H. Mao, Z. Kabelac, D. Katabi, and R. C. Miller, "Smart homes that monitor breathing and heart rate," in *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, 2015, pp. 837–846.

[10] A. Bhattacharya and R. Vaughan, "Deep learning radar design for breathing and fall detection," *IEEE Sensors Journal*, vol. 20, no. 9, pp. 5072–5085, 2020.

[11] M. G. Amin, Y. D. Zhang, F. Ahmad, and K. D. Ho, "Radar signal processing for elderly fall detection: The future for in-home monitoring," *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 71–80, 2016.

[12] Y. Jin, R. Prophet, A. Deligiannis, I. Weber, J.-C. Fuentes-Michel, and M. Vossiek, "Comparison of different approaches for identification of radar ghost detections in automotive scenarios," in *2021 IEEE Radar Conference (RadarConf21)*. IEEE, 2021, pp. 1–6.

[13] L. Wang, S. Giebenhain, C. Anklam, and B. Goldluecke, "Radar ghost target detection via multimodal transformers," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7758–7765, 2021.

[14] M. Chamseddine, J. Rambach, D. Stricker, and O. Wasenmuller, "Ghost target detection in 3d radar data using point cloud based deep neural network," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 10 398–10 403.

[15] B. Major, D. Fontijne, A. Ansari, R. Teja Sukhavasi, R. Gowaikar, M. Hamilton, S. Lee, S. Grzechnik, and S. Subramanian, "Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[16] 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 Journal of Selected Topics in Signal Processing*, vol. 15, no. 4, pp. 954–967, 2021.

[17] T. Dahmen, P. Trampert, F. Boughorbel, J. Sprenger, M. Klusch, K. Fischer, C. Kübel, and P. Slusallek, "Digital reality: a model-based approach to supervised learning from synthetic data," *AI Perspectives*, vol. 1, no. 1, pp. 1–12, 2019.

[18] H. Buddendick and T. F. Eibert, "Radio channel simulations using multiple scattering center models," in *2009 IEEE Antennas and Propagation Society International Symposium*. IEEE, 2009, pp. 1–4.

[19] T. A. Wheeler, M. Holder, H. Winner, and M. J. Kochenderfer, "Deep stochastic radar models," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 47–53.

[20] U. Chipengo, A. Sligar, and S. Carpenter, "High fidelity physics simulation of 128 channel mimo sensor for 77ghz automotive radar," *IEEE Access*, vol. 8, pp. 160 643–160 652, 2020.

[21] J. Thieling, S. Frese, and J. Roßmann, "Scalable and physical radar sensor simulation for interacting digital twins," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3184–3192, 2020.

[22] N. Hirsenkorn, P. Subkowski, T. Hanke, A. Schaermann, A. Rauch, R. Rasshofer, and E. Biebl, "A ray launching approach for modeling an fmcw radar system," in *2017 18th International Radar Symposium (IRS)*. IEEE, 2017, pp. 1–10.

[23] 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 Journal of Microwaves*, vol. 1, no. 4, pp. 962–974, 2021.

[24] U. Chipengo, A. P. Sligar, S. M. Canta, M. Goldgruber, H. Leibovich, and S. Carpenter, "High fidelity physics simulation-based convolutional neural network for automotive radar target classification using micro-doppler," *IEEE Access*, vol. 9, pp. 82 597–82 617, 2021.

[25] P. Shirley, "Ray tracing in one weekend," *Amazon Digital Services LLC*, vol. 1, 2018.

[26] M. Pharr, W. Jakob, and G. Humphreys, *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

[27] X. Li, X. Wang, Q. Yang, and S. Fu, "Signal processing for tdm mimo fmcw millimeter-wave radar sensors," *IEEE Access*, vol. 9, pp. 167 959–167 971, 2021.

[28] L. Azpilicueta, C. Vargas-Rosales, and F. Falcone, "Intelligent vehicle communication: Deterministic propagation prediction in transportation systems," *IEEE Vehicular Technology Magazine*, vol. 11, no. 3, pp. 29–37, 2016.

[29] D. Bilibashi, E. Vitucci, and V. Degli-Esposti, "Dynamic ray tracing: Introduction and concept," in *2020 14th European Conference on Antennas and Propagation (EuCAP)*. IEEE, 2020, pp. 1–5.