





# A Realistic Radar Ray Tracing Simulator for Large MIMO-Arrays in Automotive Environments

CHRISTIAN SCHÜBLER <sup>1</sup> (Graduate Student Member, IEEE),  
MARCEL HOFFMANN <sup>1</sup> (Graduate Student Member, IEEE), JOHANNA BRÄUNIG<sup>1</sup>, INGRID ULLMANN <sup>1</sup>,  
RANDOLF EBELT<sup>2</sup>, AND MARTIN VOSSIEK <sup>1</sup> (Fellow, IEEE)

(Regular Paper)

<sup>1</sup>Institute of Microwaves and Photonics, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen 91058, Germany

<sup>2</sup>BMW AG, München 80809, Germany

CORRESPONDING AUTHOR: Christian Schüßler (e-mail: christian.schuessler@fau.de).

**ABSTRACT** With the rise of high resolution multiple input multiple output (MIMO) systems, radar became an important sensor in the development of Advanced Driver Assistance Systems (ADAS) and autonomous driving applications. Autonomous driving will rely strongly on artificial intelligence. Since most modern classification algorithms are based on neural networks, they require huge amounts of data to perform well, especially in unexpected traffic situations. Radar sensor simulation can potentially produce a great variety of training data for machine learning algorithms, which makes it an important cornerstone in the development of ADAS. Furthermore, with radar simulators, different antenna configurations and various edge cases can be simulated. In this work, a versatile ray tracing toolchain based on the shoot and bouncing rays (SBR) approach is presented. The program is able to simulate complex urban environments including realistic clutter, by utilizing simplistic reflection models. The program does not only produce realistic radar images, but also generates camera-like images using the same materials. Furthermore, this work deals with the adaption of the SBR method to radar sensors with an arbitrary number of transmit (TX)- and receive (RX) antennas, which enables the simulation of large MIMO arrays. A novel performance optimization approach is proposed for large numbers of TX antennas, which reduces the runtime dramatically. The quality of the simulation is verified by measuring a complex and realistic scenario with a high resolution automotive MIMO radar. Also, a study of the effect on quality and runtime is being investigated for various optimization approaches, including the proposed method.

**INDEX TERMS** MIMO radar, radar simulation, ray tracing, shoot and bouncing rays.

## I. INTRODUCTION

Radar sensors offer various advantages compared to Lidar or camera sensors, such as being comparably cheap and also working in the absent of light and under harsh weather conditions. Latest developments in high resolution MIMO radars may also alleviate the disadvantages of a low angular resolution, compared to expensive lidar sensors [1], [2]. For these reasons, radar sensors are widely used in ADAS solutions and autonomous driving applications.

Simulating radar data brings various benefits, such as testing various antenna configurations optimized only for a specific array factor [3] under more realistic scenarios before creating a physical prototype. Also, testing novel algorithms under user defined scenarios is often easier to accomplish with simulated data compared to conducting real measurements. Radar simulations can also be used to train machine learning algorithms [4], to get more balanced data for corner cases, such as unlikely traffic situations or to enlarge the training data set in general.

There exist a number of different approaches to simulate radar sensor data. The authors in [5] differentiate between physical models, scattering centers models and data-driven models. The physical model used most often in automotive radar simulations is the shooting and bouncing rays (SBR) approach, first introduced in [6]. This technique is used in [4] and [7]–[12]. A good introduction to raytracing approaches for electromagnetic wave propagation can be found in [13].

In the SBR method, rays are transmitted from the position of the TX antenna and are reflected by the objects in the simulation environment. The simulation environment is typically modeled by triangle meshes. RX antennas are modeled as spheres, which, when reached by a ray, store the complete ray path length from TX antenna to the respective RX antenna's center point. Eventually, an intermediate frequency signal (IF signal) can be generated using the stored path lengths.

The authors in [17] claim that ray tracing methods offer realistic simulations in respect of multipath and occlusion. The drawback of these approaches is that they are computationally expensive, sophisticated to implement and require a detailed model of the environment.

Scattering centers approaches are considered to be much less computationally expensive and are able create simulation data in real-time. This approach is for example used in [5] and [14]. In this method complex objects are modeled by a number of different point targets (scattering centers) with an angle dependent RCS. Scattering centers can achieve accurate radar data, but the scatter model has to be created for each object separately. One approach is to use accurate but more time consuming raytracing simulations to generate various scattering center models, which can be used in real-time simulations afterwards [23]. Compared to triangle meshes, scattering centers only model point targets and hence not a closed surface, therefore problems with occlusion and multipath remain. In automotive scenarios, occlusion is pervasive in almost all critical scenarios, for example when a pedestrian enters the lane from behind a car. Also, multipath effects and ghost detections have to be considered, especially in dense urban environments.

Another major advantage of ray tracing methods is that they are based on 3D triangle meshes, which are pervasive and well studied in the computer graphics community. Consequently, suitable meshes can be taken from many sources including CAD models or computer games. They can also easily be created by 3D creation programs such as the open-source 3D computer graphics software “Blender”. This constitutes a great advantage of ray tracing methods, as the aforementioned resources are mainly unavailable for methods based on scattering centers.

Data-driven models generate synthetic data sets from a pre-trained neural network [15], [16]. This relatively new research field shows promising results. However, the output data can only mimic data sets stemming from real measurements or simulated data. The main goal of our simulator is to generate data sets for radar sensors with an arbitrary antenna and modulation scheme.

For these reasons the SBR method fits our needs best, especially as simulations in real-time are not required for the training of neural networks and development purposes in general. It only has to be sufficiently fast to produce *enough* data in a reasonable amount of time. The runtime can even further be reduced by introducing specialized raytracing hardware into graphics processing units (GPU) [24].

Implementations of the SBR method can also be found in commercial EM simulation tools such as FEKO commonly named Geometrical Optics (GO). They are often used to compute the radar cross section (RCS) of objects, see for example [20], [21] or [22]. The simulated scattering behavior can also be used for SAR-Imaging as shown in [18]. In [28] FEKO is used to place an antenna array in front of a 3D car model to simulate power and phase of each antenna, respectively. The simulated signals are used to evaluate different angle estimation algorithms afterwards. Since we are interested in the range, Doppler velocity and angle of each object, power and phase of the carrier signal is not suitable for our purposes. Instead we require the full IF signal.

In [19], FEKO was used to simulate a full chirp sequence FMCW signal by sampling the carrier signal at multiple frequency points during a chirp. However, this approach is significantly less efficiently compared to simulate the IF signal directly. Further, in all of the examples the 3D meshes were substantial less complex compared to our simulated scene, which consists of more than 24 000 triangles.

For us it was also not possible to simulate this scene in FEKO, since it seem to require some constraints on the input triangle mesh, such as that triangles are not allowed to overlap or should have a specific minimum or maximum size.

By implementing the SBR method by our own we can simulate the IF signal directly, are unconstrained in our 3D scene and are able to make further optimizations.

The contributions of this work are the following:

- 1) The implementation of a simplistic material model adapted from visual ray tracing approaches, which are not only able to produce realistic radar data, but also camera-like images, which makes prototyping of radar and camera fusion algorithms possible.
- 2) As far as the authors know, current publications only show realistic simulations of range-Doppler data [8], [9], or single-channel simulations [11]. We show that with the proposed material model realistic automotive range-angle data for modern high resolution MIMO sensors can be simulated. Implementation details on how an arbitrary number of TX- and RX antennas can be simulated efficiently in a modern ray tracing engine are given.
- 3) A novel optimization technique is introduced, which can dramatically improve the simulation performance for large MIMO array configurations.
- 4) An investigation on how different optimization approaches affect the quality and runtime of the simulation is conducted. For example, the size of the receive

antenna may cause phase errors, but the influence especially on millimeter waves has not been investigated yet [26].

The paper is structured as follows: In Section II, we give an overview of the simulation tool chain and a brief description of the simulator's software architecture. The implemented material models are described in Section III. Afterwards, a detailed explanation of our modifications for MIMO arrays can be found. We also describe the ray generation process, which differs from current work [8], [11], but is beneficial for our material models. Different optimization techniques are described at the end of the section. Results comparing simulated with measured data are given in Section III, in which optimization techniques are also compared in respect to image quality and runtime performance.

## II. SIMULATION TOOL CHAIN

This section gives the reader an overview of the simulator before a more detailed description follows in the next sections. It shows, how a radar signal is generated from the environment representation to an IF signal. Further the software architecture enabling a flexible, fast, and user-friendly simulator design, is briefly described.

### A. SIGNAL GENERATION

The simulation program consists of two main steps. The first step generates ray path lengths out from the environment described as triangle meshes and the TX- and RX antenna configurations. This process is described in Section III in more detail. Additional to path lengths, the ray direction can be computed, so that antenna patterns can be applied in the IF signal generation. The complete simulation workflow is summarized in Fig. 1.

Since the material models are adapted from a common visual raytracer (see section III), camera-like images can be generated with the same ray tracing procedure, by simply adding color information to the traced rays. This is not only helpful for debugging purposes, but also for testing and implementing data fusion algorithms between camera and radar data.

In the second step, the IF signal for each receive antenna is generated. The IF signal only depends on signal modulation parameters and the signal delay  $\tau$ , which again depends on the path length  $d$  between TX antenna, reflected object, and RX antenna as denoted by

$$\tau = \frac{d}{c}, \quad (1)$$

whereby  $c$  is the speed of light in vacuum. The most common modulation scheme of automotive radar is the frequency modulated continuous wave (FMCW) modulation. The radar signal is often transmitted in several chirps to measure the target's Doppler velocity [27]. The FMCW IF signal for a single reflection point can be described as below [28]

$$s_{\text{IF}_s}(t) = A(\alpha, \gamma) \exp(2\pi j(\mu t \tau + f_c \tau)). \quad (2)$$

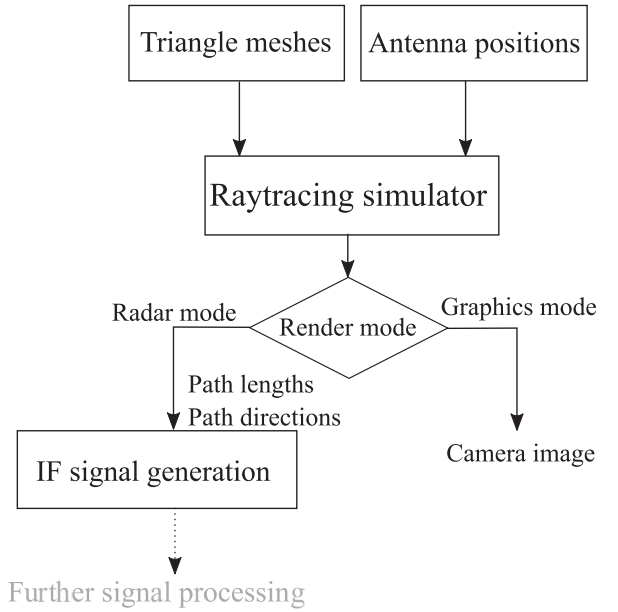


FIGURE 1. Overview of the complete simulation workflow.

The signal amplitude  $A$  depends on the antenna pattern, which is parameterized by the spherical angles  $\alpha$  (azimuth) and  $\gamma$  (elevation). The frequency slope  $\mu$  is defined by the signal's bandwidth  $B$  and chirp duration  $T$

$$\mu = \frac{B}{T}. \quad (3)$$

The symbol  $f_c$  denotes the carrier frequency. Furthermore, the signal power also depends on the path length and the radar cross section  $\sigma$  of the reflected object, as described by the radar equation [29].

The path length dependent behavior is reflected by the simulation approach directly, because the emitted power  $P_e$  can be regarded as the number of rays  $N_{\text{TX}}$  transmit from the TX antenna i.e.

$$P_e \sim N_{\text{TX}}. \quad (4)$$

As described in section III in more detail, the rays are shoot uniformly through a sphere surface. Therefore the ray density  $S_{\text{TX}}$  can be described by

$$S_{\text{TX}} = \frac{N_{\text{TX}}}{4\pi R^2}, \quad (5)$$

whereby  $R$  is the ray distance from the TX antenna. This equals again the emitted signal power density of the TX antenna as derived for the radar equation:

$$S_E = \frac{P_e}{4\pi R^2}. \quad (6)$$

For multiple reflections, the complete IF signal for a single RX antenna can be approximated by coherently summing up

each reflected signal

$$s_{IF}(t) = \sum_{i=0}^N A(\alpha, \gamma) \exp(2\pi j(\mu t \tau + f_c \tau)), \quad (7)$$

whereby  $N$  denotes the number of path lengths,  $i$  the path length index, and  $t$  the time index.

In this paper we only simulate MIMO antenna configurations, which lead to virtual uniform linear arrays (ULA). Therefore, the angular image can be calculated using a Fourier transform along the virtual channels [29].

In an FMCW signal the range is encoded in the IF signal frequency and can therefore also be computed by a Fourier transform. Computing an unambiguous range-angle radar image can be obtained by a two dimensional Fourier transform.

To reduce sidelobes, the IF signal is multiplied by a two dimensional hanning window  $w$  and zero padded beforehand. Since only discrete signals can be obtained in real world, we apply a discrete Fourier transform.

The complete reconstruction formula is given in the equation below.

$$\begin{aligned} S_{IF}(i_r, i_\alpha) &= \text{DFT}_{2d}(w(k, t) s_{if, \text{pad}}(k, t)) \\ &= \sum_{k=0}^{K-1} \sum_{t=0}^{T-1} \exp\left(-2\pi j \left(\frac{i_r t}{T} + \frac{i_\alpha k}{K}\right) w(k, t) s_{if, \text{pad}}(k, t)\right), \end{aligned} \quad (8)$$

whereby  $s_{if, \text{pad}}$  denotes the zero padded IF signal and  $k$  the virtual antenna channel index. The indices  $i_r$  and  $i_\alpha$  describe the range and angular index, respectively.

Since the simulation program also supports object animation, such as walking pedestrians or cyclist, it is also possible to simulate Doppler radar data. Simulating additional Doppler information differ only slightly from simulating range-angle data. Instead of a single simulation run, the simulation program has to be executed for each chirp separately. After each chirp the animation has to be set to the instant of time for the selected chirp  $t_s$ .

The reconstruction formula has to be extended by one dimension for the window function and the Fourier transform, as shown below:

$$S_{IF}(i_r, i_\alpha, i_v) = \text{DFT}_{3d}(w(k, t, t_s) s_{if, \text{pad}}(k, t, t_s)), \quad (9)$$

whereby the index for each Doppler bin is denoted by  $i_v$  and the time index across chirps is given by  $t_s$ .

According to [9], the scene has to be sampled with more rays compared to RCS simulations for accurate Doppler signals, because phase errors across chirps distort the signal. This is also true in our case, where range angle data is simulated.

The simulator model can later be extended to include Doppler information. However, this is a very extensive update to this novel approach. For this reason, we focus only on range angle data simulations in this work.

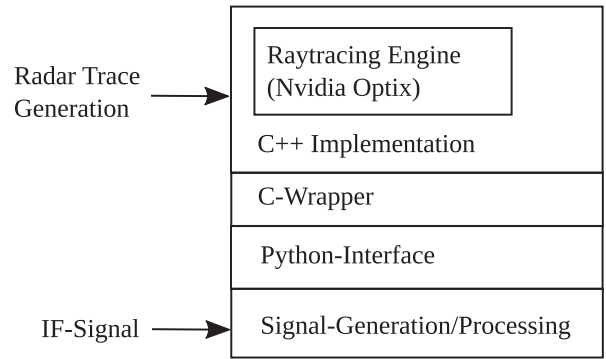


FIGURE 2. Overview of the complete software architecture.

## B. SOFTWARE ARCHITECTURE

The core of the simulator, and the most critical part with respect to runtime, is the path length computation, which is done by the ray tracing module. Since implementing a computationally efficient ray tracer is a difficult task [25], the Nvidia OptiX [30] ray tracing engine is used in many existing SBR implementations [8], [11], [31]. For this reason, it is also deployed here.

A scheme of the software architecture can be seen in Fig. 2. The ray tracing engine itself must be parametrized using the comparatively low level language C++. The complete ray tracing interface can be accessed through this language. Since Python is more accessible and common in the scientific community the C++ code is wrapped in a Python interface through an intermediate C-Wrapper.

The IF signal generation is then independent of the path length computation and different modulation schemes can be used for the same path length data.

## III. RAY TRACING TECHNIQUES

In the following we explain the material models and the ray generation approach. After that, the adaptation for MIMO arrays is explained and a description of various runtime optimization techniques is given.

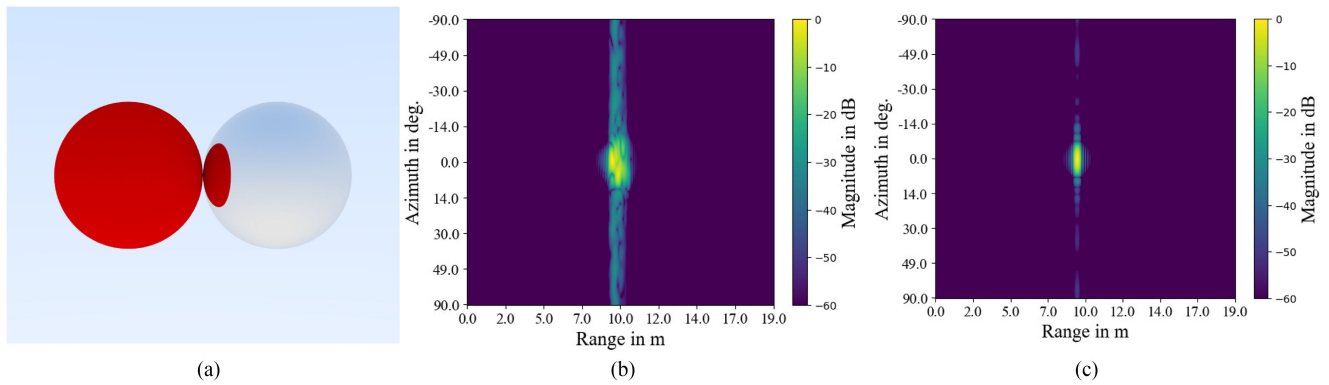
### A. MATERIAL MODELS

The implemented material model is a combination of the two models proposed in [32]. The first material simulates diffuse material by implementing Lambertian scattering. The outgoing ray direction of the diffuse scattering  $\vec{t}_{o,d}$  is computed as follows:

$$\vec{t}_{o,d} = \vec{n} + \vec{r}, \quad (10)$$

whereby  $\vec{n}$  is the object's surface normal vector and  $\vec{r}$  is a random point on a unit sphere, which is computed as explained in Section III-B.

The second material describes metal, or more specific a perfect reflecting mirror, where the incident ray  $\vec{t}_i$  angle equals



**FIGURE 3.** Image (a), red Sphere modeling diffuse material and reflecting sphere modeling metal material. Image (b), two dimensional FFT of a single diffuse sphere. Image (c), two dimensional FFT of a single metal sphere. Each receive antenna perceived around 260 rays.

the outgoing ray  $\vec{t}_{o,m}$  angle.

$$\vec{t}_{o,m} = -2\vec{t}_i \cdot \vec{n}. \quad (11)$$

Fig. 3 shows the graphical and radar response of the two types of material. In the original work, the metal material was extended to less perfect reflecting metal. Even by using the extended metal material, we were not able to produce satisfactory simulation results for all objects.

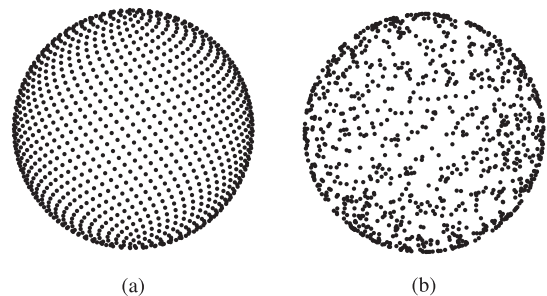
However, with a combination of these two material models we get better results than using the single materials alone. Also we could simulate our complete environment with one single parametric material. The formula of the outgoing ray  $\vec{t}_{o,c}$  is

$$\vec{t}_{o,c} = \alpha \frac{\vec{t}_{o,d}}{|\vec{t}_{o,d}|} + (1 - \alpha)\vec{t}_{o,m}, \quad (12)$$

assuming that  $\vec{t}_{o,m}$  is already normalized. The factor  $\alpha$  decides whenever the material behaves diffuse or more metal-like.

## B. RAY GENERATION

Since a ray is reflected in a random manner, more than one single reflection is required to compute a representative object reflection behavior. This approach is common in graphical ray tracing [25] to compute a noise free or at least less noisy image. However, for SBR methods not based on a probabilistic material models, the rays are often sent in equidistantly distances [8], [11] using a spherical Fibonacci lattice [33]. Shooting rays in equidistant distances, often combined with correction techniques, are required to avoid double counts [35], [36], [37], [38]. A double count occurs if two rays representing the same wavefront are perceived by a RX antenna, but only a single hit is desired. In other ray generation setups it may also happen that one ray is not perceived by any receive antenna, which is also not desired. However, in the automotive application in [11] double counts did not deteriorate the result noticeable.



**FIGURE 4.** Image (a) each point has the same distance to its neighbor. Image (b), points are distributed randomly, but uniformly on the sphere's surfaces, which is computationally more efficient and suitable for our approach. For both images 1000 points were distributed.

Since we use probabilistic material models, which reflect rays in a random manner, more than one ray per facet or object is required. Distributing the launched rays uniformly will lead to unbiased counting errors, which should decrease with the number of rays.

Shooting large numbers of rays is computationally expensive when using the Fibonacci lattice method since it requires trigonometric functions. However, we do not require the distance between rays to be equal. Instead, it is sufficient to create rays, which are distributed uniformly on a sphere's surface.

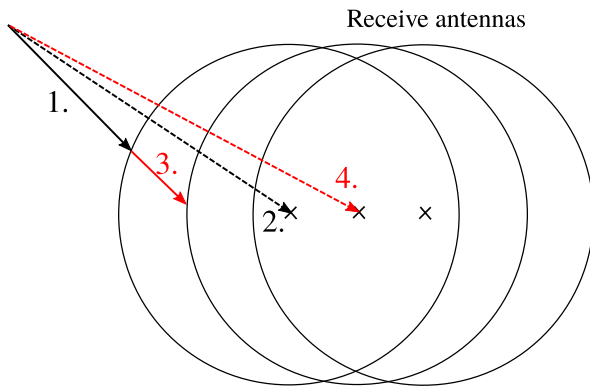
This can be done efficiently by drawing each coordinate of the ray direction  $\vec{a}$  from a normal distribution  $\mathcal{N}$  and dividing each by the length of the resulting vector ([34], p. 135), as described in the equations below:

$$\vec{a} = \frac{1}{|\vec{a}|}(a_x, a_y, a_z), \quad (13)$$

with

$$a_{x/y/z} \sim \mathcal{N}(0, 1). \quad (14)$$

A comparison between the Fibonacci lattice method and the uniform generation is shown Fig. 4.



**FIGURE 5.** Top view of three RX antennas shown as circles. The solid lines show rays and the dashed lines show calculated path lengths. When an RX antenna is hit by a ray (step 1), the distance between the starting point of the ray and the center position of the RX antenna is calculated and added to the total ray's path length (step 2), afterwards the current ray is terminated. Next, a new ray is started slightly inside the hit RX antenna having the same direction as the ray before (step 3). As soon as this ray hits an antenna again, the path length is computed as before (step 4). This procedure is repeated until all antennas are processed and the final ray is transmitted into the simulation environment again.

### C. SIMULATING MULTIPLE RECEIVE ANTENNAS

The single elements in antenna arrays are commonly spaced closely. On the other hand, the reception radius of the antenna is simulated much greater than the distance between two antennas. For example, in a uniform linear array, the distance between each element must be less than half a wavelength for an unambiguous direction of arrival (DOA) estimation, which is only 1.9 mm for a typical automotive radar at 77 GHz. In contrast, the perception sphere is often simulated with a size of around 1 m in our simulation scenarios. By using this configuration, the perception spheres of each receive antenna overlap dramatically as illustrated in Fig. 5. In this case, the outer antennas occlude the inner antennas and these are therefore unreachable for incident rays.

In the following we describe our solution for this problem. The position  $\vec{s}_0(k)$  of a ray perceived by the first non occluded perception sphere can be described by

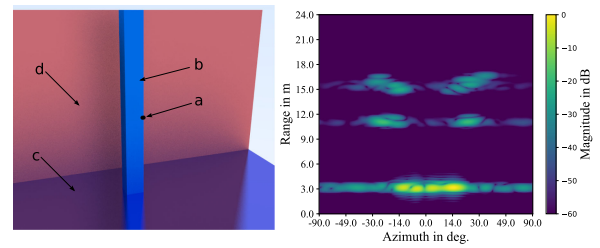
$$\vec{s}_0(k) = k\vec{r} + \vec{d}, \quad (15)$$

whereby  $k$  describing a scalar,  $\vec{r}$  the ray direction and  $\vec{d}$  the ray origin, in this case the reflection point.

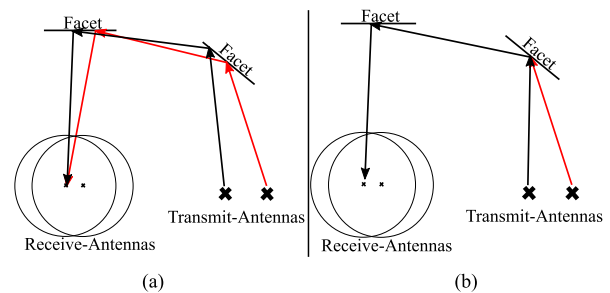
Before starting a new ray, the antenna stores the complete ray path length, which persists of the distance  $l$  traveled from the shooting TX antenna to the last reflection point and the distance between the last reflection point to the center of the hit RX antenna. The computation of the complete path length  $l^*$ , required for the IF signal generation can then be summarized in the following equation

$$l^* = l + |\vec{x}_{RX} - \vec{s}_0(0)| = l + |\vec{x}_{RX} - \vec{d}|, \quad (16)$$

with  $\vec{x}_{RX}$  denoting the center position of the selected RX antenna.



**FIGURE 6.** Artificial example showing occlusion and multipath. The camera-like image is shown in the left picture. The radar signal is sent out from the antenna array (a) and is reflected from the blue pillar (b), from the red background wall (d) and over the reflecting ground (c) again from the red wall. In the right image the range angle radar image is shown. The blue pillar occlude the red wall partially, furthermore the radar signal reflected from the red wall appears twice. Once at the correct position and once again behind the correct position as effect of multipath.



**FIGURE 7.** Image (a), each TX antenna shoots rays in another random direction, leading to phase errors at the RX antenna. Image (b), only one TX antenna shoots in a random direction and all other send their rays to the same first hit position. The random generators are initialized identically for each transmitted ray, so that each ray follows the same path. With this approach, phase errors in the RX channels can be avoided.

Afterwards, a new ray is generated, which starts at the intersection point between the initial ray and the perception sphere. The new starting point is realized with a scalar offset  $h_1$ . The new ray has the same origin and direction as the initial ray, as shown in the equation below

$$\vec{s}_1(k) = (k + h_1 + \epsilon)\vec{r} + \vec{d}. \quad (17)$$

A tiny offset  $\epsilon$  is required to avoid a double hit with the initial hit antenna. The new ray also stores the variable  $l$  of the previous ray in order to determine the correct path length of the next RX antenna hit. This process is repeated until all RX antennas are passed, see Fig. 5.

### D. SIMULATING MULTIPLE TRANSMIT ANTENNAS

Transmitting rays in different directions and hitting different objects, which are again reflecting randomly, would lead to different path lengths for each ray of every transmit antenna. Since different path lengths lead to different phase information, as can be seen in (2), this behavior would distort the phase information perceived by the RX antennas. Since DOA estimation is based on the phase information between all antennas [39], this simulation approach is unsuitable, as illustrated in Fig. 7.

**Algorithm 1:** Rays From mMultiple TX Antennas.

---

**Input:** Number Tx-Antennas  $N_A$ , number rays  $N_R$

```

1: for  $i_{ray} = 0$  to  $N_R - 1$  do
2:    $i_{tx} \leftarrow \text{rand\_uniform}(N_A)$  ▷Select random antenna
3:   ray_pos  $\leftarrow$  tx_antennas[ $i_{tx}$ ]
4:   ray_dir  $\leftarrow$  random_direction()
5:   s  $\leftarrow$  generate_seed() ▷create seed variable s
6:   hit_pos  $\leftarrow$  shoot_ray(ray_pos, ray_dir, s) ▷Get hit pos of first facet
7:   for  $j_{tx} = 0$  to  $N_A - 1$  do
8:     if  $j_{tx} \neq i_{tx}$  then
9:       ray_pos  $\leftarrow$  tx_antennas[ $j_{tx}$ ]
10:      ray_dir  $\leftarrow$  ray_pos - hit_pos
11:      shoot_ray(ray_pos, ray_dir, s)
12:     end if
13:   end for
14: end for

```

---

To solve this problem, a ray is shot by a single arbitrary TX antenna and the intersection position with the first hit triangle is stored. Afterwards, all other TX antennas transmit rays to the stored position. To avoid different reflection behaviors, each ray initializes the random number generator with the same seed. The complete ray shooting process is described in Algorithm 1. This shooting procedure is named *burst* in the following.

As can be noticed, the angle-dependent reflection property of the material model is not affected by this approach. This is in that sense critical, because most objects in automotive applications have an angle dependent radar cross section (RCS) [40], [41].

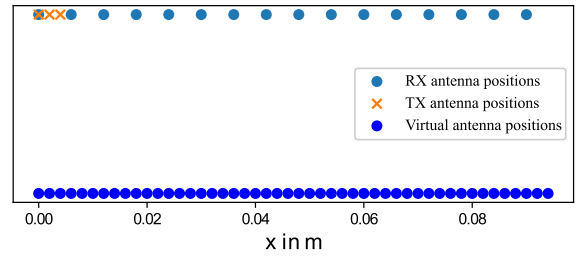
Shooting rays with these techniques enable simulations that do not only take different reflection behaviours into account, but also consider occlusion and multipath effects. This is depicted in Fig. 6 for an artificial example.

### E. OPTIMIZATION TECHNIQUES

The runtime performance mainly depends on the number of rays shot. To be able to shoot a smaller number of rays, more of the transmitted rays must be perceived by the corresponding RX antennas. Each transmitted ray which is shot, but not perceived by an antenna does not contribute to the generated signal and therefore only wastes computation time.

The simplest method to increase the number of perceived rays is to enlarge the size of the RX antenna's perception sphere.

Another widely used approach is to simulate the rays as cone beams, by making the antenna's perception sphere radius depending on the ray path length [36]. With this approach objects which are closer to the RX antenna are simulated with smaller antenna sizes and objects further away are detected with larger perception spheres.



**FIGURE 8.** The antenna configuration consists of 3 transmit- and 16 receive antennas operating in MIMO mode. The resulting array contains 36 virtual antennas with a distance of 2 mm.

Simulating radar data with larger sphere radii also affects the simulation quality, as investigated in the results section.

In addition to these two optimization techniques, the proposed simulation program implements another novel optimization approach suited for multiple collocated TX antennas. As already mentioned in Section III-D, for one ray *burst* the reflection behavior only depends on the incident angle of the ray. In most cases, the objects are far away from the transmit antennas and the antennas itself are very close to each other in typical MIMO scenarios. Therefore, this angle dependency can often be neglected.

Applying this simplification, it is sufficient to shoot only one ray for each *burst*. The calculated path length can then be corrected for every other transmit antenna. Selecting the antenna for ray shooting randomly should further alleviate the error caused by this approach. The equation below shows how to correct the path length  $l_i$  measured by the TX antenna at position  $\vec{x}_{TX,i}$  for all other antennas at position  $\vec{x}_{TX,j}$ .

$$l_j = l_i - |\vec{z}_i - \vec{x}_{TX,i}| + |\vec{z}_i - \vec{x}_{TX,j}|, \quad (18)$$

whereby  $\vec{z}_i$  is the hit position of the  $i$ th TX antenna, as described in Section III-D.

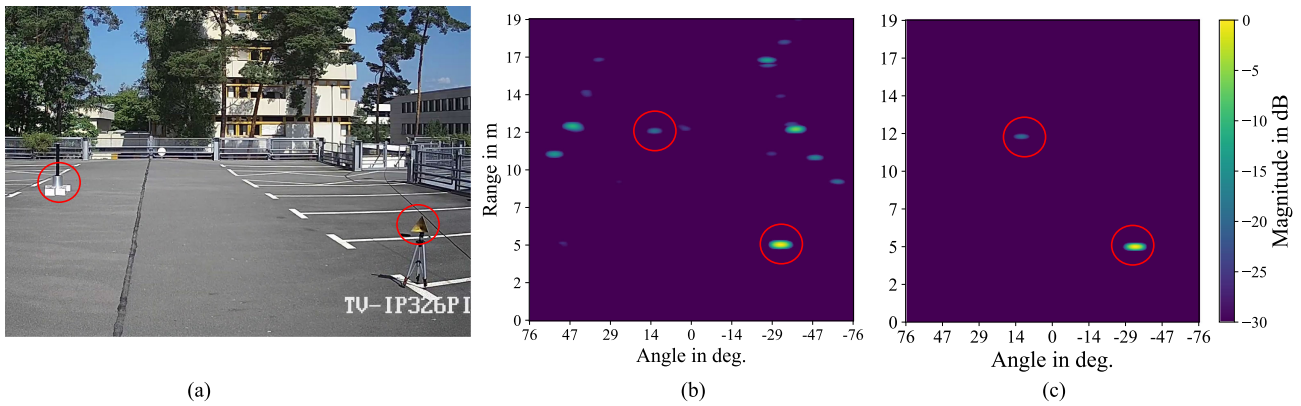
## IV. RESULTS

This section is divided into two subsections. In the first subsection we describe the measurement setups and also the 3D model of the simulated scenes. In the second subsection the simulation results are shown and different optimization techniques are assessed.

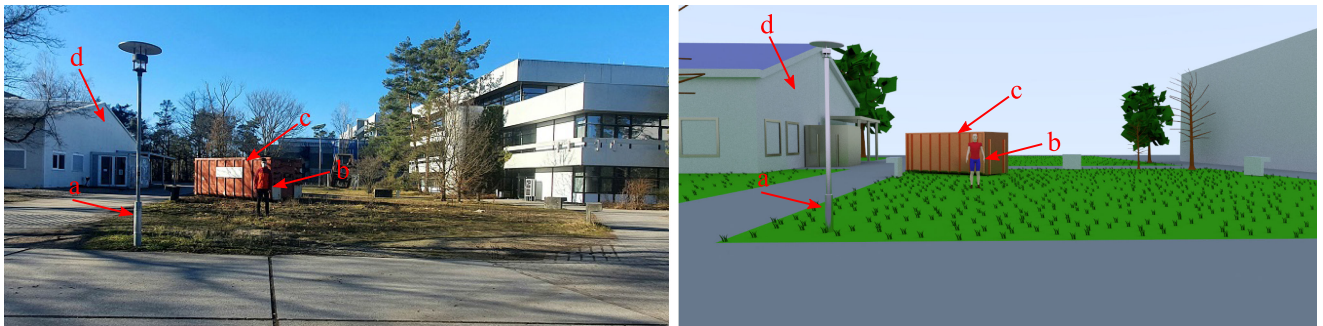
### A. MEASUREMENT SETUP

The antenna configuration of the used radar sensor is shown in Fig. 8 and the parameters of the radar sensor can be found in Table 1.

To verify the proposed simulator, we simulated two different scenes. The first simplistic scenario depicted in Fig. 9 only consists of a single metallic cylinder and a corner reflector, which were both simulated with perfectly reflecting ( $\alpha = 0$ ) material. The purpose of a simplistic scene is to eliminate the influence of derivations of the created 3D environment from the real world. With this approach, we can directly evaluate



**FIGURE 9.** The left image (a) shows a camera image of the measured scene. The metallic cylinder in the background has a diameter of 24 cm and the inner edges of the corner reflector in the foreground have a length of 14 cm. The center image (b) shows the measured scene and the right image (b) shows the radar simulation of both objects. The fence, which is visible in the measured radar data was not simulated.



**FIGURE 10.** The left image shows the measured scene from a position near the radar unit. The right image shows the created 3D simulation scene, which is rendered by the graphical render path of the implemented simulation program, using the same material models as for the radar simulation. The red arrows points to some objects of interest, which can also be found in the simulated radar data shown in other figures.

**TABLE 1.** Parameters of Radar Sensor

Symbol	Parameter	Value
$f_c$	RF carrier frequency	77 GHz
$B$	RF bandwidth	1 GHz
$f_s$	IF sampling frequency	533 MHz
$T_{chirp}$	Chirp duration	80.6 $\mu s$
$N_{chirp}$	Number of chirps	1
$M$	Number of RX antennas	16
$N$	Number of TX antennas	3

**TABLE 2.** Simulation Object Materials

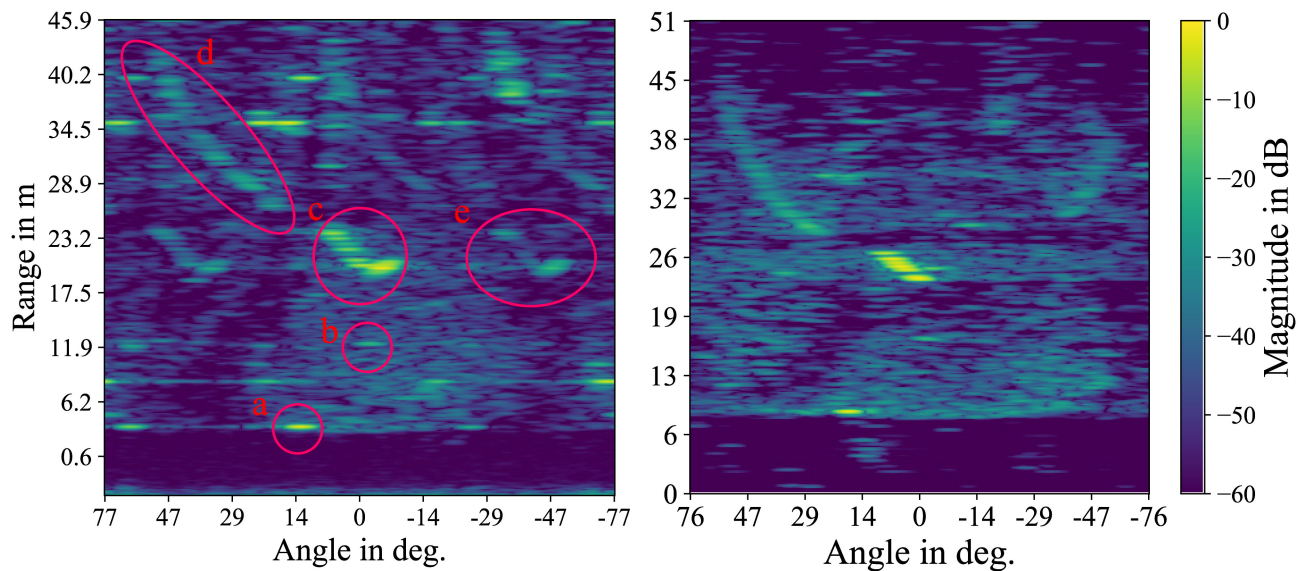
Object	Material parameter value
Red container	$\alpha = 0.15$
Left lantern	$\alpha = 0.15$
Door left building	$\alpha = 0.1$
Pillars left building	$\alpha = 0.1$
Window Frames left building	$\alpha = 0.2$
Buildings	$\alpha = 0.7$
Lawn	$\alpha = 0.7$
Pavement	$\alpha = 0.7$
Pedestrian	$\alpha = 0.5$

the simulation algorithm itself and detect shortcomings or algorithmic errors.

The second rather complex scene aims to mimic a challenging urban scenario. It contains one pedestrian, a red strongly reflecting container and some buildings and trees in the background. Fig. 10 shows a photo of the measured scene in the perspective of the radar unit and the modeled 3D scene. The tree and the pedestrian model were taken from the

Open-Source Autonomous-Driving-Simulator CARLA [42]. All chosen material models of the 3D scene are listed in Table 2.





**FIGURE 11.** The left image shows the measured range angle image, obtained by the described reconstruction algorithm. The right image shows the simulation for the modeled 3D scene. The objects in the red circle (a) is the radar response of the left front lantern, the circle (b) shows the pedestrian, and (c) is the response of the red metal container. The big building is indicated by ellipse (d). Radar responses in the red ellipses with label (e) are caused by antenna calibration errors, which are not part of the simulation and are therefore not present in the right image.

The simulations run on a workstation with 64 GB memory, an Intel Core i7-9700 CPU @ 3 GHz and a Nvidia RTX 3070 GPU with 8 GB video memory.

### B. SIMULATION RESULTS

In the first simplistic setup in Fig. 9, it can be observed that the reflected power of the cylinder in both simulation and measurement is around 20 dB below the power of the corner reflector. Apart from the surrounding fences that were not simulated, both images are almost identical. This gives hint that the implemented algorithms work as expected.

In the next simulation case we investigate the complex scenario as described in the section before. Here, we not only focus on the radar image quality in general, but also on the runtime performance for different optimization approaches and their influence on the simulation result.

In the initial simulation no performance optimization was applied. The rays were simulated as tubes, not as cones and the RX antenna radius was chosen to 10 cm, which is very small compared to the objects in the simulated scene. Since in this work, we mainly focus on the quality of the range-angle radar images, only a single chirp was simulated. Therefore, no Doppler information is available.

The measurement is compared to the simulation in Fig. 11. As can be seen, the simulation approximates the real measurement well, especially the clutter originating from the grass could be simulated well by placing low-poly grass chunks randomly on the lawn. It also can be seen that in contrast to the red metallic container and the lantern, the pedestrian is hardly visible through the clutter. For this initial simulation 983 million rays were shot from each TX antenna, but only around 10 000 rays were perceived by each RX antenna. The

**TABLE 3.** RMSE and Stdev for Different Sphere Sizes

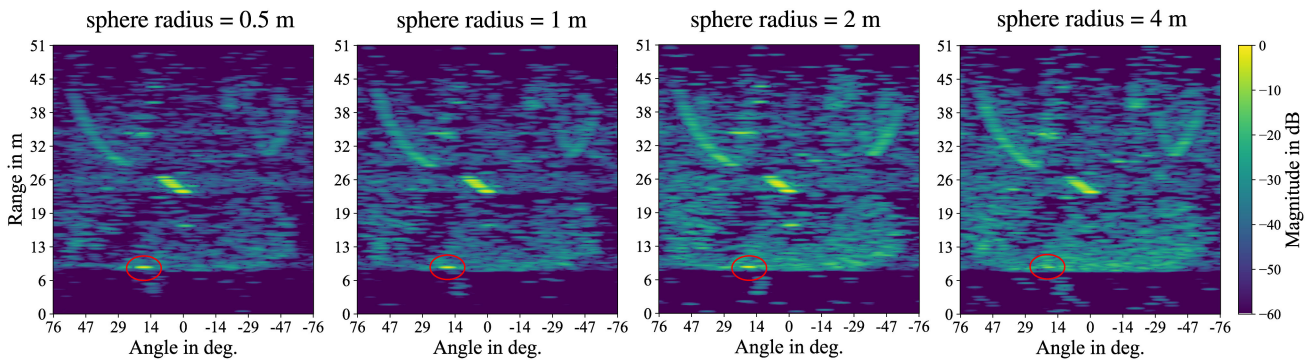
Sphere size	RMSE	Stdev
0.5 m	-48.8 dB	-40.6 dB
1.0 m	-47.7 dB	-40.2 dB
2.0 m	-42.0 dB	-35.5 dB
4.0 m	-41.1 dB	-34.3 dB

simulation (excluding IF signal generation) took 4 minutes and 44 seconds.

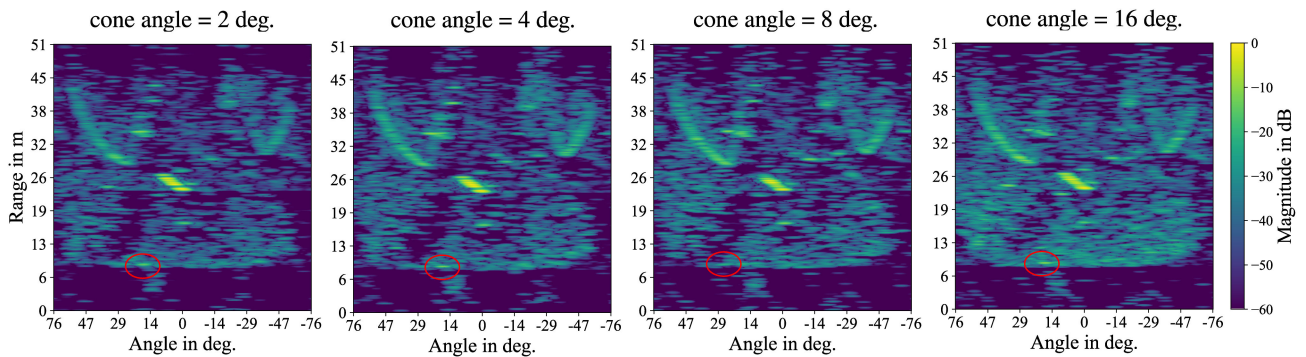
However, this simulation time is not only far from a real time performance, it is also too slow for simulations that include Doppler information. For radar data including Doppler information, several chirps have to be transmitted. This would multiply the simulation time by the number of chirps. For this reason we investigate how the proposed optimization techniques scale to the quality and the runtime of the simulation.

First of all we vary the radius of the perception sphere from 0.5 m to 4.0 m while trying to keep the number of received rays constant by simultaneously reducing the number of transmitted rays. The simulated radar images for different RX antenna sizes are shown Fig. 12. In order to compare the results quantitatively the root mean square error (RMSE) and standard deviation (Stdev) of the pixel wise difference of the radar images was computed.

Table 3 shows RMSE and the Stdev between the optimized simulations and the accurate simulation with sphere radius of 0.1 m shown at the beginning.



**FIGURE 12.** Resulting radar image for different sphere sizes. At a sphere radius of 4 m the result image deteriorates noticeable. Especially near the antennas the clutter seems to be magnified so that the left lantern is almost not distinguishable from the background clutter. The red circle shows the image area of the left lantern.

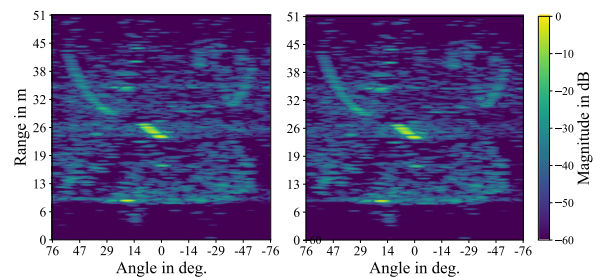


**FIGURE 13.** Resulting radar image for different cone angles. The right image shows a similar effect as for large sphere radii, the clutter cover the signal of the left lantern almost completely. The red circle shows the image area of the left lantern.

As can be seen, the RMSE between 0.5 m and 1.0 m is almost equal. At a radius of 2.0 m and 4.0 m the RMSE increases by almost 6 dB and 7 dB, respectively. In the simulation results, the lantern then becomes indistinguishable from the background clutter. This is because, metal like objects reflect incident rays under the same angle. Therefore small RX antennas only sense rays from the lantern, which are reflected under a very specific incoming angle. For very large antennas this does not hold anymore, since the perception sphere is hit by rays from a wide angular range. Therefore, even metallic objects become diffuse for larger perception spheres.

However, with larger sphere radii, the runtime decreases dramatically and also the novel optimization approach, shooting only one transmit ray and adjust the path lengths for all other antennas brings a considerable runtime benefit, as can be seen in Fig. 16. In Fig. 14, the computed radar image for a sphere radius of 1 m is shown, with and without the optimization for TX antennas.

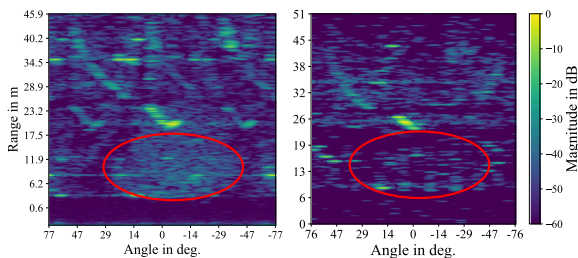
Often rays are simulated as cones to increase the probability of perceiving rays from distant objects. Fig. 13 and Table 4 show the resulting images and error values for different cone angles, respectively. For greater cone angles similar effects as for greater sphere sizes occur i.e. metal-like objects seems



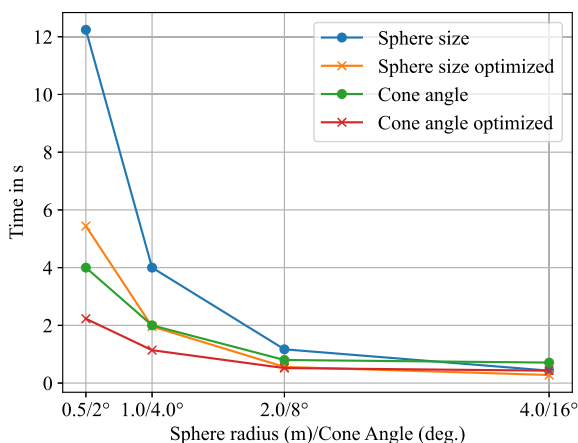
**FIGURE 14.** Resulting radar image without (left image) and with (right image) the optimization for more TX antennas. No remarkable difference between the two image can be observed. The measured RMSE between the two radar images was less than one dB.

**TABLE 4.** RMSE and Stdev for Different Cone Angles

Cone angle	RMSE	Stdev
2 deg	-46.8 dB	-37.61 dB
4 deg	-44.6 dB	-36.9 dB
8 deg	-43.2 dB	-34.1 dB
16 deg	-42.1 dB	-35.4 dB



**FIGURE 15.** Measured data (left) compared to simulated radar image using a fibonacci lattice with 10 million sample points to compute ray directions (right). The overall runtime did not differ considerable, however the simulation results deviate in some parts significantly (clutter region in the red circle). This method performs worse than methods computing ray directions randomly.



**FIGURE 16.** Runtime for different sphere radii and different cone angles. Optimized graphs are measured using the TX antenna optimization approach proposed in Section III.

to reflect rays diffuse. The runtime performance for different cone angles is also shown in Fig. 16.

We further compared the image quality and runtime performance of rays generated using the Fibonacci lattice method and as proposed by generating ray directions randomly. Even though the operations used in the Fibonacci lattice method are more complicated, the overall runtime of the simulation was not affected noticeable. However, the image quality of the simulation results presented in Fig. 15 differs significantly compared with the proposed method. This is especially true for areas where the clutter signal dominates. For the simulation of the Fibonacci lattice method, the number of sample points on the sphere was increased up to 10 million. However, even with this large number of samples, there is still a small constant gap between each ray. This consequently explains, why ground clutter cannot be simulated correctly. Further increasing the number of sample points to 100 million lead to flawed images when using single precision floating point values, as they are not accurate enough to represent the smaller gaps between the launched rays. Using double precision data types solves this problem but consequently increases

the runtime significantly, because most state-of-the-art GPUs are optimized for single precision computations.

## V. CONCLUSION

In this work we proposed a radar ray tracing simulator based on a novel simplistic material model. Problems occurring for the simulation with multiple antennas have been solved and the results show that the simulator can reproduce measurement data realistically. Since runtime is crucial to produce enough simulation data, for example for machine learning algorithms, different optimization approaches have been evaluated in respect of runtime and quality. It turned out that even with large RX antenna sizes (around 2 m) the simulations do not lose their accuracy. Furthermore, the novel optimization approaches for several TX antennas do not influence the simulation quality, but can instead improve the runtime performance significantly. For larger antenna arrays, the proposed technique accelerates the simulation process even more, since it scales with the number of TX antennas. For example, we were able to simulate a MIMO antenna array with 64 TX- and 64 RX antennas in around 15 seconds. Therefore the implemented simulator satisfies our needs to produce high resolution radar images in an acceptable amount of time. This simulator can be used in future applications to train or at least pre-train neural networks for classification or radar image reconstruction. Since the simulator also allows to simulate camera images, it can also be used to develop data fusion algorithms between camera and radar sensors. However, the runtime performance and image quality for Doppler data has not been investigated and the authors intend to focus on these issues in their future research. Especially how the simulation can be optimized over several chirps is of great interest.

## ACKNOWLEDGMENT

The authors would like to thank Jannis Groh and Analog Devices, Inc. for the measurement setup.

## REFERENCES

- [1] C. Waldschmidt, J. Hasch and W. Menzel, "Automotive radar - from first efforts to future systems," *IEEE J. Microwaves*, vol. 1, no. 1, pp. 135–148, Jan. 2021, doi:[10.1109/JMW.2020.3033616](https://doi.org/10.1109/JMW.2020.3033616).
- [2] I. Bilik, O. Longman, S. Villeval and J. Tabrikian, "The rise of radar for autonomous vehicles: Signal processing solutions and future research directions," *IEEE Signal Process. Mag.*, vol. 36, no. 5, pp. 20–31, Sep. 2019, doi:[10.1109/MSP.2019.2926573](https://doi.org/10.1109/MSP.2019.2926573).
- [3] D. Mateos-Núñez, M. A. González-Huici, R. Simoni, F. B. Khalid, M. Eschbaumer and A. Roger, "Sparse array design for automotive MIMO radar," in *Proc. 16th Eur. Radar Conf.*, Paris, France, 2019, pp. 249–252.
- [4] A. P. Sligar, "Machine learning-based radar perception for autonomous vehicles using full physics simulation," *IEEE Access*, vol. 8, pp. 51470–51476, 2020, doi:[10.1109/ACCESS.2020.2977922](https://doi.org/10.1109/ACCESS.2020.2977922).
- [5] M. Jasiński, "A generic validation scheme for real-time capable automotive radar sensor models integrated into an autonomous driving simulator," in *Proc. 24th Int. Conf. Methods Models Autom. Robot.*, Miedzyzdroje, Poland, 2019, pp. 612–617, doi:[10.1109/MMAR.2019.8864669](https://doi.org/10.1109/MMAR.2019.8864669).
- [6] H. Ling, R. Chou and S. Lee, "Shooting and bouncing rays: Calculating the RCS of an arbitrarily shaped cavity," *IEEE Trans. Antennas Propag.*, vol. 37, no. 2, pp. 194–205, Feb. 1989, doi:[10.1109/8.18706](https://doi.org/10.1109/8.18706).

- [7] M. Maier, V. P. Makkapati and M. Horn, "Adapting phong into a simulation for stimulation of automotive radar sensors," in *Proc. IEEE MTT-S Int. Conf. Microw. Intell. Mobility*, Munich, Germany, 2018, pp. 1–4, doi:10.1109/ICMIM.2018.8443493.
- [8] 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, doi:10.1109/JSEN.2020.3026416.
- [9] S. O. Wald and F. Weimann, "Ray tracing for range-doppler simulation of 77 GHz automotive scenarios," in *Proc. 13th Eur. Conf. Antennas Propag.*, Krakow, Poland, 2019, pp. 1–4.
- [10] A. Schmitz and L. Kobbelt, "Efficient and accurate urban outdoor radio wave propagation," in *Proc. Int. Conf. Electromagn. Adv. Appl.*, Turin, Italy, 2011, pp. 323–326, doi:10.1109/ICEAA.2011.6046362.
- [11] N. Hirsenkorn *et al.*, "A ray launching approach for modeling an FMCW radar system," in *Proc. 18th Int. Radar Symp.*, Prague, Czech Republic, 2017, pp. 1–10, doi:10.23919/IRS.2017.8008120.
- [12] L. Reichardt, J. Maurer, T. Fugen and T. Zwick, "Virtual drive: A complete V2X communication and radar system simulator for optimization of multiple antenna systems," *Proc. IEEE*, vol. 99, no. 7, pp. 1295–1310, Jul. 2011, doi:10.1109/JPROC.2011.2124430.
- [13] Z. Yun and M. F. Iskander, "Ray tracing for radio propagation modeling: Principles and applications," *IEEE Access*, vol. 3, pp. 1089–1100, 2015, doi:10.1109/ACCESS.2015.2453991.
- [14] H. Buddendick and T. F. Eibert, "Radio channel simulations using multiple scattering center models," in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, North Charleston, SC, USA, 2009, pp. 1–4, doi:10.1109/APS.2009.5172257.
- [15] T. A. Wheeler, M. Holder, H. Winner and M. J. Kochenderfer, "Deep stochastic radar models," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Los Angeles, CA, USA, 2017, pp. 47–53, doi:10.1109/IVS.2017.7995697.
- [16] T. Truong and S. Yanushkevich, "Generative adversarial network for radar signal synthesis," in *Proc. Int. Joint Conf. Neural Netw.*, Budapest, Hungary, 2019, pp. 1–7, doi:10.1109/IJCNN.2019.8851887.
- [17] M. Holder *et al.*, "Measurements revealing challenges in radar sensor modeling for virtual validation of autonomous driving," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, Maui, HI, USA, 2018, pp. 2616–2622, doi:10.1109/ITSC.2018.8569423.
- [18] A. Brandewie and R. J. Burkholder, "FEKO simulation of radar scattering from objects in low earth orbit for ISAR imaging," in *Proc. Int. Appl. Comput. Electromagn. Soc. Symp.*, 2020, pp. 1–2, doi:10.23919/ACES49320.2020.9196137.
- [19] S. Yoo, H. Kim, G. Byun, and H. Choo, "Estimation of detection performance for vehicle FMCW radars using EM simulations," *J. Electromagn. Eng. Sci.*, vol. 19, no. 1, pp. 13–19, Jan. 2019, doi:10.26866/JEES.2019.19.1.13.
- [20] G. Zhang, W. Huo, W. Jiang, Y. Huang, J. Yang and J. Wu, "RCS model of ship based on FEKO," in *Proc. Int. Conf. Comput. Problem-Solving*, 2013, pp. 419–422, doi:10.1109/ICCP.2013.6893565.
- [21] E. Bel Kamel, A. Peden and P. Pajusco, "RCS modeling and measurements for automotive radar applications in the W. band," in *Proc. 11th Eur. Conf. Antennas Propag.*, 2017, pp. 2445–2449, doi:10.23919/EuCAP.2017.7928266.
- [22] X. Wang, C. Wang and Y. Liu, "RCS computation and analysis of target using FEKO," in *Proc. 3rd Asia-Pacific Conf. Antennas Propag.*, 2014, pp. 822–825, doi:10.1109/APCAP.2014.6992625.
- [23] R. Bhalla and H. Ling, "Three-dimensional scattering center extraction using the shooting and bouncing ray technique," *IEEE Trans. Antennas Propag.*, vol. 44, no. 11, pp. 1445–1453, Nov. 1996, doi:10.1109/8.542068.
- [24] A. R. Willis, M. S. Hossain, and J. Godwin, "Hardware-accelerated SAR simulation with NVIDIA-RTX technology," in *Proc. SPIE 11393, Algorithms for Synthetic Aperture Radar Imagery XXVII*, 2020, doi:10.1117/12.2558091
- [25] S. Marschner and P. Shirley, *Fundamentals of Computer Graphics*. Boca Raton, FL, USA: CRC-Press, 2018
- [26] M. M. Taygur and T. F. Eibert, "A ray-tracing algorithm based on the computation of (exact) ray paths with bidirectional ray-tracing," *IEEE Trans. Antennas Propag.*, vol. 68, no. 8, pp. 6277–6286, Aug. 2020, doi:10.1109/TAP.2020.2983775.
- [27] F. Roos, J. Bechter, C. Knill, B. Schweizer and C. Waldschmidt, "Radar sensors for autonomous driving: Modulation schemes and interference mitigation," *IEEE Microw. Mag.*, vol. 20, no. 9, pp. 58–72, Sep. 2019, doi:10.1109/MMM.2019.2922120.
- [28] S. M. Patole, M. Torlak, D. Wang and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Process. Mag.*, vol. 34, no. 2, pp. 22–35, Mar. 2017, doi:10.1109/MSP.2016.2628914.
- [29] M. Richards, *Fundamentals of Radar Signal Processing*. New York, NY, USA: McGraw-Hill, 2014
- [30] S. G. Parker *et al.*, "OptiX: A general purpose ray tracing engine," in *ACM SIGGRAPH 2010 Papers*, vol. 29, no. 4, pp. 1–13, Jul. 2010, doi:10.1145/1778765.1778803.
- [31] R. Felbecker, L. Raschkowski, W. Keusgen and M. Peter, "Electromagnetic wave propagation in the millimeter wave band using the NVIDIA OptiX GPU ray tracing engine," in *Proc. 6th Eur. Conf. Antennas Propag.*, Prague, Czech Republic, 2012, pp. 488–492, doi:10.1109/EuCAP.2012.6206198.
- [32] P. Shirley, "Ray tracing in one weekend," 2018. [Online]. Available: <https://github.com/petershirley/raytracinginoneweekend>
- [33] A. González, "Measurement of areas on a sphere using fibonacci and latitude-longitude lattices," *Math. Geosci.*, vol. 42, no. 1, pp. 49–64, 2010.
- [34] D. E. Knuth, *The Art of Computer Programming: Seminumerical Algorithms*, vol. 2, 3rd ed. Reading, MA, USA: Addison-Wesley, 1998
- [35] N. Noori, A. A. Shishegar and E. Jedari, "A new double counting cancellation technique for three-dimensional ray launching method," in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, Albuquerque, NM, USA, 2006, pp. 2185–2188, doi:10.1109/APS.2006.1711020.
- [36] S. Kasdorf, B. Troksa, C. Key, J. Harmon and B. M. Notaroš, "Advancing accuracy of shooting and bouncing rays method for ray-tracing propagation modeling based on novel approaches to ray cone angle calculation," *IEEE Trans. Antennas Propag.*, vol. 69, no. 8, pp. 4808–4815, Aug. 2021, doi:10.1109/TAP.2021.3060051.
- [37] Z. Yun, M. F. Iskander and Z. Zhang, "Development of a new shooting-and-bouncing ray (SBR) tracing method that avoids ray double counting," in *IEEE Antennas Propag. Soc. Int. Symp. Dig. Held Conjunction: USNC/URSI National Radio Sci. Meeting (Cat. No. O1CH37229)*, Boston, MA, USA, 2001, vol. 1, pp. 464–467, doi:10.1109/APS.2001.958892.
- [38] V. Mohtashami and A. A. Shishegar, "A new double-counting cancellation technique for ray tracing using separation angle distribution," in *Proc. IEEE Int. RF Microw. Conf.*, Kuala Lumpur, Malaysia, 2008, pp. 306–310, doi:10.1109/RFM.2008.4897435.
- [39] J. Li and P. Stoica, "MIMO radar with colocated antennas," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 106–114, Sep. 2007, doi:10.1109/MSP.2007.904812.
- [40] S. B. Jayapal Gowdu, A. Schwind, R. Stephan, and M. A. Hein, "Monostatic RCS measurements of a passenger car mock-up at 77 GHz frequency in virtual environment," in *Proc. 49th Eur. Microw. Conf.*, 2019, pp. 996–999, doi:10.23919/EuMC.2019.8910719.
- [41] M. Chen and C. Chen, "RCS patterns of pedestrians at 76–77 GHz," *IEEE Antennas Propag. Mag.*, vol. 56, no. 4, pp. 252–263, Aug. 2014, doi:10.1109/MAP.2014.6931711.
- [42] A. Dosovitskiy, G. Ros, F. Codevilla, A. López, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.



**CHRISTIAN SCHÜBLER** (Graduate Student Member, IEEE) was born in Fürth, Germany, in 1991. He received the B.Eng. and M.Eng. degrees in electrical engineering and information technology from the Technische Hochschule Nürnberg Georg-Simon-Ohm, Nuremberg, Germany, in 2014 and 2015, respectively. Since 2019, he has been working toward the Ph.D. degree with the Institute of Microwaves and Photonics. From 2015 to 2019, he was a Software Engineer with the Algorithmic Group, Fraunhofer Center for X-Ray Development in the field of computed tomography. His research interests include simulation, signal processing, and data fusion, especially for large MIMO arrays in automotive applications.



**MARCEL HOFFMANN** (Graduate Student Member, IEEE) was born in Witten, Germany, in 1994. He received the B.Sc. and M.Sc. degrees in electrical engineering from the Friedrich-Alexander University Erlangen-Nürnberg, Erlangen, Germany, in 2016 and 2018, respectively. He is currently working toward the Ph.D. degree with Friedrich-Alexander University Erlangen-Nürnberg. In 2018, he joined the Institute of Microwaves and Photonics, Friedrich-Alexander University Erlangen-Nürnberg. His research interests include radar signal processing and automotive radar applications including new SAR and SLAM approaches.



**JOHANNA BRÄUNIG** was born in Bamberg, Germany, in 1993. She received the M.Sc. degree in electrical engineering from the Friedrich-Alexander University of Erlangen-Nuremberg, Erlangen, Germany, in 2020. Since 2020, she has been working toward the Ph.D. degree with the Institute of Microwaves and Photonics. Her research interests include radar imaging and radar applications in medical technology.



**INGRID ULLMANN** is currently working toward the Ph.D. degree with the Institute of Microwaves and Photonics, Friedrich-Alexander University Erlangen-Nuremberg, Erlangen, Germany. Her research interests include radar imaging and radar signal processing for nondestructive testing, security, medical, and automotive applications. She is the Group Leader of the research group Radar and Imaging Systems, Institute of Microwaves and Photonics and is a Reviewer for the European Radar Conference and various journals in the field

of microwaves. She was the recipient of the Argus Science Award (sponsored by Airbus Defense and Space, now Hensoldt) for the master thesis in 2016 and the Best Paper Award of the European Radar Conference in 2019.



**RANDOLF EBELT** received the Ph.D. degree from the Institute of Microwaves and Photonics, University of Erlangen-Nuremberg, Erlangen, Germany, in 2014. He is currently a Sensor Perception Specialist with BMW Group, where he develops radar based driver assistance systems and autonomous driving. He is the author of various publications on wireless positioning and radar signal processing.



**MARTIN VOSSIEK** (Fellow, IEEE) received the Ph.D. degree from Ruhr-Universität Bochum, Bochum, Germany, in 1996. In 1996, he joined Siemens Corporate Technology, Munich, Germany, where he was the Head of the Microwave Systems Group from 2000 to 2003. Since 2003, he has been a Full Professor with Clausthal University, Clausthal-Zellerfeld, Germany. Since 2011, he has been the Chair of the Institute of Microwaves and Photonics, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany.

He has authored or coauthored more than 250 articles. His research has led to more than 90 granted patents. His current research interests include radar, transponder, RF identification, communication, and locating systems. He is a Member of organizing committees and technical program committees for many international conferences. He is a Member of the German IEEE Microwave Theory and Techniques (MTT)/Antennas and Propagation Chapter Executive Board and the IEEE MTT-S Technical Coordinating Committees MTT-24, MTT-27, and MTT 29. He was the Founding Chair of the MTT IEEE Technical Coordinating Subcommittee MTT-27 Wireless-Enabled Automotive and Vehicular Application. He was the recipient of several international awards. Recently, he was awarded the 2019 Microwave Application Award from the IEEE MTT Society (MTT-S). He was on the Review Boards for numerous technical journals. From 2013 to 2019, he was an Associate Editor for IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES.