IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 19, 2021, accepted March 1, 2021, date of publication March 8, 2021, date of current version March 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3064396

The Open-Source Framework for 3D Synthetic Aperture Radar Simulation

JEDRZEJ DROZDOWICZ, (Graduate Student Member, IEEE)
Department of Electronics and Information Technology, Institute of Electronic Systems, Warsaw University of Technology, 00-665 Warsaw, Poland
e-mail: j.drozdowicz@elka.pw.edu.pl

This work was supported by the National Science Centre Poland "Three-dimensional synthetic aperture radar imaging using optimized, complex trajectory of the radar carrier movement" under Grant 2015/19/N/ST7/01506.

ABSTRACT Synthetic Aperture Radar (SAR) simulation is widely used for system design, processing techniques development, and mission planning. However, there is no readily available and free framework for SAR raw signal simulation, and many teams and organizations struggle with developing their own simulator from scratch, repeating work that others have done earlier. This work's purpose is to create a simple, open-source SAR simulation framework that can be used for many purposes and is available for free to everyone. It is a ready tool for SAR processing techniques verification, and thanks to its open-source nature, it fosters collaboration between scientists, both on the simulation and the tool development. Moreover, as the simulation's underlying mathematics is described in detail in this article, it may serve as a handbook for the radar's simulation. The simulator was prepared as a stand-alone, multiplatform software working in the Matlab/Octave environment. It is available in an online software repository that allows others to contribute to the original code or to create forks of it. The simulator supports monostatic, bistatic, and multistatic configurations with shadowing. The surface model includes roughness using the modified Phong model, transparency, and complex reflectivity. The antenna pattern is defined with a general two-dimensional model. The article is concluded with the simulator's demonstrations, including antenna pattern, Doppler shift, and various surface parameters. It is expected that this simulator will be widely adopted and improved by many collaborators.

INDEX TERMS Computer simulation, open-source software, radar imaging, radar signal processing, synthetic aperture radar.

I. INTRODUCTION

Synthetic Aperture Radar technology is a crucial imaging and remote sensing method in various fields, and new SAR systems are continually being developed. SAR simulation tools are used at every step of system development and deployment, such as system design and verification, algorithm testing (target recognition, GMTI (Ground Moving Target Indication), autofocusing), and mission co-simulation. Simulated SAR data are essential for developing algorithms using machine learning to train and test neural networks.

Although there are many SAR simulators, most of them are closed-source, used only by a single research group or a company. This is acceptable for SAR simulators deployed as standalone commercial products; however, most of the existing simulators are part of larger projects, with many not being under development anymore. There are two open-source SAR image simulators – RaySAR [1], [2] and PolSARproSim [3]. They are under constant development and have been used for years in numerous projects; however, they are very complex tools and do not produce raw radar data.

This article presents 3dsar-sim, a simple, open-source SAR simulator that aims to be developed, modified, and expanded by anyone. It is not meant to be a single solution for every possible simulation problem but rather a common base to build upon. Such an approach will reduce development time and foster collaboration between various teams worldwide.

It is worth noting that the open-source simulator mentioned before – RaySAR – has been used by numerous research groups all over the world [4]–[8].

The article is organized as follows: Section II introduces the initial assumptions and requirements for a SAR simulator, Section III provides a comprehensive review of existing SAR simulators, and Section IV provides the details of

The associate editor coordinating the review of this manuscript and approving it for publication was Chengpeng Hao.

simulator design and implementation. The results are provided in Section V. Additionally, the open-source license considerations are presented in Section VI. The article is concluded in Section VII.

II. ASSUMPTIONS AND REQUIREMENTS

A. SIMULATOR TYPE

There are two types of SAR simulators: raw signal simulators and image simulators [53]. The former create raw signals based on the radar and trajectory parameters, and the latter produce ready images [54]. This article describes a raw signal simulator that can be used for the design and evaluation of a real SAR system. It has to be noted that ''*a SAR raw signal simulator is not a surrogate of the actual SAR data campaign but rather a complementary tool in order to better understand and exploit the geophysical information contained in the SAR imagery*'' [53]. With this in mind, it becomes evident that a universal SAR simulator, however desirable, is impossible to create, as the simulator must be particularly suited for its intended use. On the other hand, a simple basis for a SAR simulator common for many different applications can be created. This research's purpose is not to create a universal simulator but instead to provide an ecosystem of modular tools that can be easily used, developed, and adapted by radar researchers and engineers worldwide.

B. SAR SIMULATOR OR A RADAR SIMULATOR

The main difference between a SAR simulator and a radar simulator is the Doppler effect's omission in the former one, as it does not play an important role in SAR imaging [55]. However, it cannot be neglected in very high-resolution imaging and some GMTI techniques. Thus, the simulator should include the Doppler effect. It must be noted, however, that although such a simulator can be used as a radar simulator, its primary purpose is SAR simulation, and the scattering model, scene definition, and signal generation are all SAR-oriented.

C. CODE COMPLEXITY

Based on the experience of many open-source projects, the author has decided to put the code readability over the performance, which often comes with complexity. Thanks to Moore's law, the simulation time can be reduced using more computing power. Optimizations from contributors that do not clutter the code are welcome; however, modifications that reduce the readability should be developed as separate versions. The simulator in its original form is dedicated to a generic single- or multi-core computer able to run Matlab or Octave. The simulator does not use Graphics Processing Units (GPUs); however, a version that does can be created by contributors.

D. PLATFORM COMPATIBILITY

As the simulator is open-source itself, it is meant to run not only in the Matlab environment but also in open-source Octave. Therefore, Matlab-specific functions are not present in the code. Again, contributors are welcome to create their own derivatives that are either hardware- or software-specific.

E. MODULARITY AND SCALABILITY

One of the principles of the simulator is modularity and scalability. It is straightforward to prepare a simple single-point simulation with an omnidirectional antenna, but it is also possible to simulate a complex scene with shadowing, different surface models, and multiple independently moving antennas with individual beam patterns forming a MIMO array.

F. MULTISTATIC/MIMO CAPABILITY

The simulator in its current version is bistatic, which means it has one transmitting and one receiving antenna that can have the same or different locations. The multistatic/MIMO capability can be achieved by running a simulation of the same scene with different antenna trajectories. Such an approach makes the simulator easier to use for monostatic and bistatic scenarios.

III. OVERVIEW OF EXISTING SIMULATORS

Tables 1 and 2 show the lists of the existing image and raw signal simulators, respectively, mentioned in publications. Both tables show a vast spectrum of simulator applications and implementation details. It needs to be stressed that only two of these simulators are fully open-source [2], [3], and there are many very similar simulators on the list. Again, it is impossible to replace all of them with a single solution, but a certain amount of work can be avoided by introducing a common basis. Even the development of the hardware-based simulators [42], [47] can be simplified by using a software-based one as a reference.

Although the presented simulator is very simple and does not represent a technological breakthrough in the field, apart from being open-source, there are several advantages that are not often found in other solutions. The ability to create a scene composed of both single points and planes is important, as is the ability to model the full pattern of the antenna and include its rotation around the axis. An additional advantage is that the planes have only four parameters (transparency, roughness, magnitude, phase), so it is possible to create and manipulate scenes in standard CAD software, where the planes also have four parameters (red, green, blue, transparency).

IV. THE SIMULATOR

This chapter describes the developed simulator in detail. It includes the dataflow and underlying models of each step. The simulation algorithm is depicted in Figure 1 and consists of the following steps:

Define sar scene with at least the transmitting and receiving antennas' positions. Objects can be specified as faces or omnidirectional scattering points.

Convert faces to points so the objects are converted to scatterers.

TABLE 1. A List of SAR Image Simulators.

Name/source	Main purpose	Details	Year
[9]	urban areas		1994
[10]	realistic multisensor simulation	CAD model, commercial product	2002
[11]	real-time simulation	GPU-based	2006
PolSARproSim [3]	forest areas, polarimetry	open-source, active project	2006
RaySAR [1], [2]	spaceborne, urban areas	open-source	2008
MOCEM [12]	general-purpose	fast, non-commercial, for collaborative work and government, CAD models	2008
PIRDIS [13]	GMTI		2008
[14]	general-purpose	double bounce, GPU, real-time	2009
[15]	geometric simulator of CAD data		2009
[16]	urban scenes, large infrastructure		2011
[17]	urban structures, image interpretation		2011
CohRaSS [18]	large urban scenes		2012
[19]	urban structures	double reflection	2012
[20]	speckle simulator	physical approach	2013
[21]	canopy simulator		2014
[22]	urban structures		2016

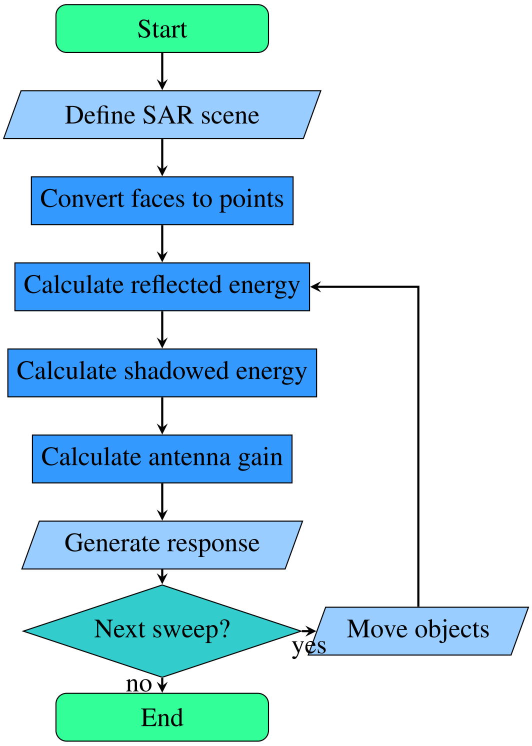

FIGURE 1. Simulation algorithm steps.

Calculate the reflected energy for each point, taking the position and orientation of the antennas, points, and faces into account.

Calculate energy shadowed by each of the faces with respect to each point and both antennas.

Calculate antenna gain taking the position, rotation, and beam pattern into account.

Generate a complex response in the form of amplitude versus range.

The steps from the calculation of the reflected energy to the response generation must be repeated for every sweep, for every transmitter-receiver antenna pair. The objects (points, faces, and antennas) can be freely moved between the sweeps.

A. COORDINATE SYSTEM

For an open-source simulator intended to be used by many experts, it is essential to assume a well-defined coordinate system. A truly three-dimensional simulator must define 6 degrees of freedom: three-dimensional position and three-dimensional rotation. For the position, a right-handed (positive) Cartesian coordinate system is used: $\mathbf{p} = [p_x, p_y, p_z]^T$. Although there are no restrictions regarding the ground plane's position, the most popular convention is to define it as the $x - y$ plane. There are several conventions present for orientation, such as the Euler angles, the orientation matrix, or the orientation vector. Because in geometrical scene definition and processing the normal vector is often needed for calculation, the orientation vector $\mathbf{d_r}$ and the rotation angle θ_r were specified. Such an approach makes the definition most comprehensive and straightforward, as usually only the vector $\mathbf{d_r}$ will be used for most of the objects in the SAR scene. The rotation angle θ_r is needed only for the antenna with a defined two-dimensional pattern and for future implementations with polarimetry support. The omnidirectional objects (such as the omnidirectional scattering points and the omnidirectional antennas) have only three degrees of freedom, and for these objects both the orientation vector

TABLE 2. **A List of SAR Raw Signal Simulators.**

Name/source	Main purpose	Details	Year
SARAS [23], [24]	facet approximation of the 3D scene	2D-FFT based	1991
SiSAR [25]	system simulator, development aid	modular	1995
SARIS [26]	detailed instrument simulator with hardware imperfections	MATLAB implementation	2001
[27]	general-purpose	inverse algorithm creating raw data from SAR image	2002
GRECOSAR [28]	polarimetric SAR/ISAR simulator	processing	2006
[29]	clutter simulation for GMTI	java implementation, multiplatform	2006
SBRAS [30]	spaceborne SAR, formation flying	simulation and processing, GUI, DEM generation, performance analysis	2007
SARSIM [31]	GMTI		2007
[32]	image quality evaluation	parameter-based	2007
[33]	scanline simulator, X-band SAR		2008
[34]	fast simulator of extended scenes		2008
[35]	multilevel interferometric simulator	processing	2009
DIONISOS [36], [37]	all-in-one SAR simulator	processing	2010
[38]	bistatic simulator		2011
[39]	multi-mode simulator	processing	2012
[40]	point target simulator, calibration		2012
[41]	general-purpose	3D, combining geometrical optics and full-wave electromagnetic methods	2014
[42]	closed-loop testing of UAV-based radar	FPGA-enabled hardware in the loop	2014
[43]	spaceborne spotlight SAR focusing		2014
Wavemill End-to-End Simulator (WE2ES) [44]	ocean currents estimation	multi-beam capabilities	2014
cSAR [45]	multichannel SAR system simulator		2014
[46]	bistatic simulator of extended scenes		2014
[47]	general-purpose	FPGA-enabled hardware in the loop	2015
Synthetic aperture radar imaging simulator [48]	pulse envelope evaluation, spotlight SAR	quality metrics	2017
[49]	GMTI		2017
[50]	high performance, multi-mode, trajectory deviations		2017
Pol-SARAS [51]	extended soil surfaces	fully polarimetric	2017
CLUSIM [52]	clutter simulator for planetary exploration		2017

and the rotation angle are ignored. The orientation vector is a unit vector $\|\mathbf{d_r}\| = 1$, and no rotation corresponds to $\mathbf{d_r} = \mathbf{i_z} \triangleq [0, 0, 1]^T$. The coordinate system is depicted in Figure 2. It presents a simple rectilinear stripmap/spotlight trajectory; however other trajectory shapes can be defined, such as circular SAR or more complex non-rectilinear trajectories.

B. SCATTERING MODEL

Among the raw signal simulators, two types can be distinguished: the point simulator and the extended scene simulator [53]. However desirable, the latter requires extended models for electromagnetic scattering, and the models are not always available. Moreover, the models are often more complicated than the simulator itself. However, the simulator is designed in a way that allows for the implementation of extended models.

There are three types of scattering (point scattering, surface scattering, volumetric scattering). The real scenes are a combination of these three types; however, the implementation of the volumetric scattering would make the simulator overly complicated, as it would require defining the scene as a set of three-dimensional shapes and implementing a complex radar response [56], [57]. On the other hand, the surface

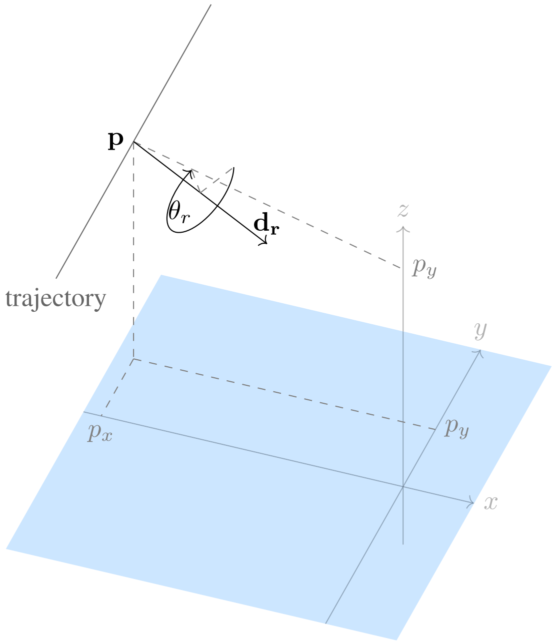

FIGURE 2. **Coordinate system.**

scattering can be easily approximated by point scattering if the surfaces are converted to points and the orientation vector is maintained. As a result, the volumetric scattering is not implemented.

Assuming the radar signal is a single LFM (linear frequency modulation) pulse [58]:

$$x(t) = A\,\mathrm{rect}\left(\frac{t}{T}\right)\cos\left(2\pi f_0 t + \pi\alpha t^2\right) \tag{1}$$

where f_c is the carrier (center) frequency, $\alpha = \frac{B}{T}$ is the LFM slope, B is the bandwidth, T is the pulse duration, and A is the signal amplitude.

A response of a single-point target in the time domain after pulse compression and matched filtering is [58] as follows:

$$\begin{aligned} E_1(t) = A_1 & \left(1 - \left|\frac{t-t_1}{T}\right|\right) \\ & \mathrm{sinc}\left(\left(f_d + \alpha\left(t - t_1\right)\right) T \left(1 - \left|\frac{t-t_1}{T}\right|\right)\right) \\ & \exp\left(j2\pi f_c \left(t - t_1\right)\right) \exp\left(j\pi f_d \left(t - t_1\right)\right) \\ & \exp\left(j\phi_1\right) \end{aligned} \tag{2}$$

where A_1 is the scatterer amplitude, ϕ_1 is the scatterer phase, $f_d = -\frac{2v_1}{c} f_c$ is the target Doppler frequency, where v_1 is the target velocity, and $t_1 = \frac{r_1}{c}$, where r_1 is the target distance from the antenna. For a bistatic configuration, r_1 is the bistatic range: $r_1 = \frac{r_{1_{TX}} + r_{1_{RX}}}{2}$, where $r_{1_{TX}}$ and $r_{1_{RX}}$ are the target distances from the transmitting and receiving antennas, respectively. The bistatic velocity $v_1 = v_{1_{TX}} + v_{1_{RX}}$, where $v_{1_{TX}}$ and $v_{1_{RX}}$ are the target radial velocities relative to the transmitting and receiving antennas, respectively. This response form is valid for large time-bandwidth products $BT \gtrsim 100$ [59]. It must be noted that the inclusion of the target velocity may lead to a range-Doppler ambiguity in the obtained radar response.

If the radar scene consists of more than one point, the response is the sum of responses from each scatterer [60]:

$$\begin{aligned} E(r) = \sum_{n=1}^{N} A_n & \left(1 - \left|\frac{t-t_n}{T}\right|\right) \\ & \mathrm{sinc}\left(\left(f_d + \alpha\left(t - t_n\right)\right) T \left(1 - \left|\frac{t-t_n}{T}\right|\right)\right) \\ & \exp\left(j2\pi f_c \left(t - t_n\right)\right) \exp\left(j\pi f_d \left(t - t_n\right)\right) \\ & \exp\left(j\phi_n\right) \end{aligned} \tag{3}$$

where $t_n = \frac{r_n}{c}$, where $r_n = r_1 \ldots r_N$ is the bistatic distance of the nth of all N targets.

C. 3D SCENE DEFINITION

The way a SAR scene is defined has a significant influence on the simulator's behavior. There are objects that can be represented as a combination of single-point scatterers and objects better represented by three-dimensional solids. For 3D urban area simulation, detailed building models are needed [61]. Such models are useful not only for the airborne SAR but also for the spaceborne SAR simulation. Buildings with flat walls are not detailed enough for very high-resolution SAR, so windows, balconies, and other facade irregularities must be included as well for the simulation to match real radar signals [61].

The scene formation and manipulation are not included in the simulator, but the input format is designed to make scene formation and manipulation easy to perform with other software.

D. CONVERTING FACES TO POINTS

As described in Section IV-B, the scene objects represented as solids must be converted to point scatterers. This is accomplished by converting faces to points while maintaining the parameters responsible for directionality and shadowing. The dataflow of the process is presented in Figure 3. One important issue to consider is the distribution of the points on the surfaces. Evenly distributed points would result in periodicity in the radar response and thus may cause the appearance of false grating lobes related neither to the simulated radar parameters nor to the scene geometry. To avoid this, random sampling is used. To obtain a pseudo-uniform distribution of points on the plane in 3D, the triangle is rotated so that it lies on the X-Y plane. The smallest rectangle containing this triangle is then determined, and the generation of points with a uniform distribution inside this rectangle begins. Points not contained within the triangle are discarded, and point generation stops when the number of points n_{pts} satisfies the condition:

$$n_{pts} = \lceil \frac{A}{d^2} \rceil, \tag{4}$$

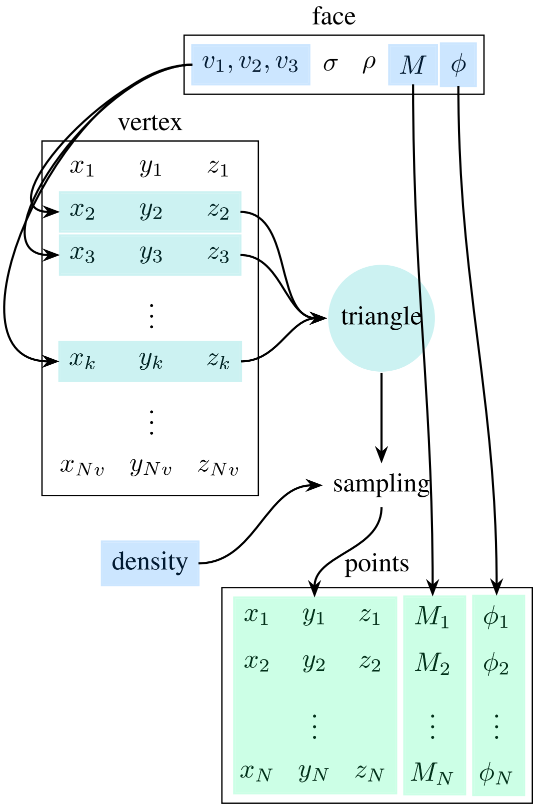
face
v_1, v_2, v_3 σ ρ M ϕ
vertex
x_1 y_1 z_1
x_2 y_2 z_2
x_3 y_3 z_3
$\vdots$
x_k y_k z_k
$\vdots$
x_{Nv} y_{Nv} z_{Nv}
triangle
sampling
density
points
x_1 y_1 z_1 M_1 ϕ_1
x_2 y_2 z_2 M_2 ϕ_2
$\vdots$ $\vdots$ $\vdots$
x_N y_N z_N M_N ϕ_N

FIGURE 3. The dataflow of converting faces to point scatterers.

where d is the density and A is the area of the triangle. The points are then rotated back to the initial position of the triangle in 3D.

It must be noted that converting faces to point scatterers is a simplification and may lead to improper or inaccurate representation, especially if the sampling density is too low. Another important note is that when using pseudorandom sampling, the Random Number Generator (RNG) seed must always be the same in order to achieve repeatability of the simulation.

E. REFLECTIVITY

There are various reflectivity models, and they can be divided into two groups: physical and empirical. The former are more complicated and require detailed knowledge of the material parameters, while the latter are less accurate but can be easily adapted for different materials. The reflectivity models were defined based on physical properties for typical simple shapes, such as a sphere [62] or a cylinder [63]. For an ideal plane, the model is trivial, as it is for an ideal point scatterer. The problem is that real scenes consist of more complex shapes that cannot be easily approximated by basic ones. Research shows that model parameters change depending on the wavelength and object's shape, size, and material [64]. Surfaces, especially natural ones, are not actually flat and cannot be accurately represented by a plane. A fractal model has been used successfully, but it is too complex and requires knowledge of the real object's parameters [65]. Another approach is to use a three-dimensional model and volumetric scattering, for example, through a multilayer model [66]. All of the previously mentioned approaches have in common the high complexity and the need for a thorough investigation of the simulated surfaces. For general-purpose simulations, particularly when the scene is extensive and consists of both natural and man-made surfaces, it may be reasonable to use a more straightforward approach. Optical models are much simpler because they do not account for volumetric scattering, but they do allow certain surface features, such as roughness, to be rendered [67].

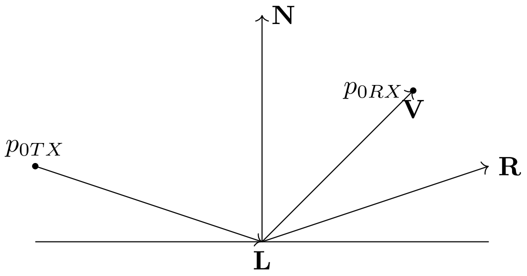

FIGURE 4. Reflectivity model.

For this simulator, the Phong model [68] was modified and used, as presented in Figure 4. A face (surface) has, apart from its position and orientation, three parameters associated with reflection:

- Magnitude M in range $0-1$ corresponding to the portion of the energy reflected from the surface;
- Phase shift ϕ induced by the surface;
- Roughness ρ in range $0 - 1$, where $\rho = 0$ corresponds to a flat, mirror-like surface, $\rho = 1$ to a rough surface with diffuse reflection, and $0 < \rho < 1$ corresponds to a surface with specular reflection, as presented in Figure 5.

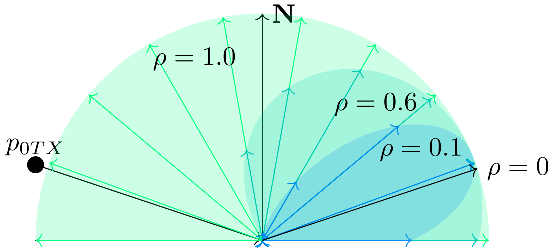

FIGURE 5. Surface roughness model.

The reflectivity of a surface is modeled using a modified Phong model [68]:

$$S = (\mathbf{R_n} \cdot \mathbf{V_n})^{\rho_n} \tag{5}$$

where the operator $(\cdot)$ denotes the dot product and

$$\mathbf{R_n} = \frac{\mathbf{R}}{\|\mathbf{R}\|} \tag{6}$$

$$\mathbf{R} = -\mathbf{L_n} + 2\,(\mathbf{N} \cdot \mathbf{L_n})\,\mathbf{N} \tag{7}$$

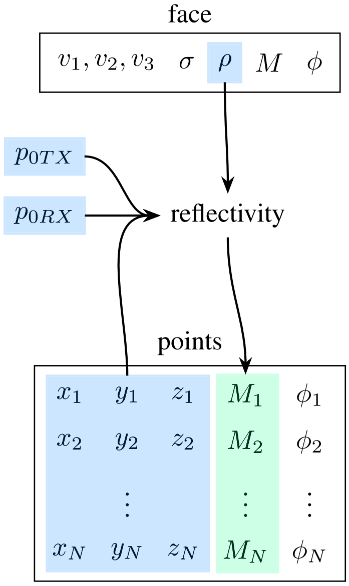
face
v_1, v_2, v_3 σ ρ M ϕ
p_{0TX}
p_{0RX}
reflectivity
points
x_1 y_1 z_1 M_1 ϕ_1
x_2 y_2 z_2 M_2 ϕ_2
x_N y_N z_N M_N ϕ_N

FIGURE 6. The dataflow of calculating the reflectivity.

$$\mathbf{L_n} = \frac{\mathbf{L}}{\|\mathbf{L}\|} \tag{8}$$

$$\mathbf{V_n} = \frac{\mathbf{V}}{\|\mathbf{V}\|} \tag{9}$$

$$\rho_n \triangleq \tan\left(\frac{\pi}{2} - \rho\frac{\pi}{2}\right) \tag{10}$$

where **N** is the normal vector of the face, **L** is the position of the face center with respect to the transmitting antenna, and **V** is the position of the receiving antenna with respect to the face center. It should be noted that although for real surfaces the reflectivity is usually a combination of specular and diffuse, the use of only one parameter does not limit the generality of this approach since the sum of specular and diffuse reflectivity can be obtained by overlaying two planes with different roughness values.

The dataflow of the reflectivity calculation is presented in Figure 6.

F. SHADOWING

If a point scatterer is obstructed by an object, its echo can be attenuated. The degree of the attenuation is specified by the face's transparency parameter σ in range $0-1$, corresponding to a portion of energy not blocked (neither reflected nor dissipated) by the surface. As the simulator is not energy preserving, this parameter is not necessarily related to the surface magnitude parameter M. The dataflow of the process is presented in Figure 7.

To determine if a point is shadowed by a face, its 3D barycentric coordinates need to be calculated as follows:

$$\begin{bmatrix}\lambda_1\\ \lambda_2\\ \lambda_3\end{bmatrix} = \boldsymbol{T}^{-1}\left(\begin{bmatrix}x_p\\ y_p\\ z_p\end{bmatrix} - \begin{bmatrix}x_0\\ y_0\\ z_0\end{bmatrix}\right) \tag{11}$$

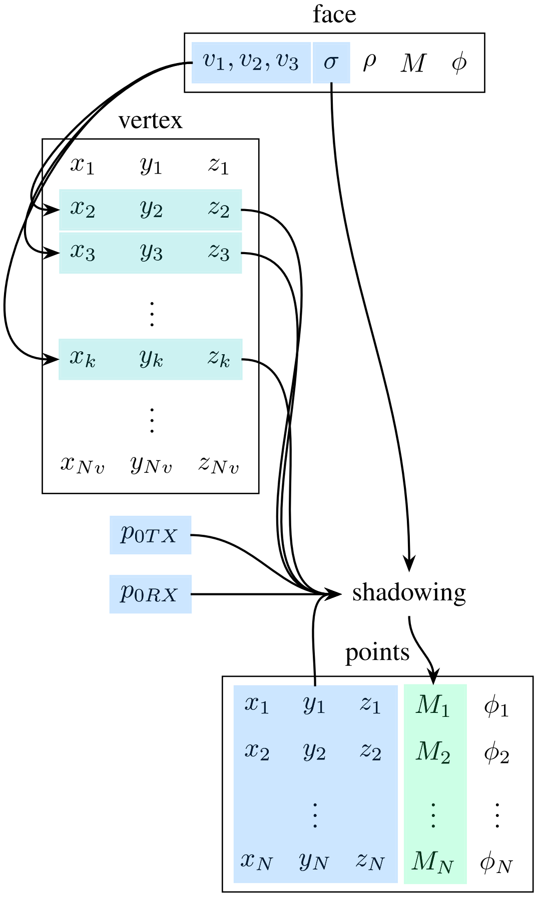
face
v_1, v_2, v_3 σ ρ M ϕ
vertex
x_1 y_1 z_1
x_2 y_2 z_2
x_3 y_3 z_3
x_k y_k z_k
x_{Nv} y_{Nv} z_{Nv}
p_{0TX}
p_{0RX}
shadowing
points
x_1 y_1 z_1 M_1 ϕ_1
x_2 y_2 z_2 M_2 ϕ_2
x_N y_N z_N M_N ϕ_N

FIGURE 7. The dataflow of shadowing.

where

$$\boldsymbol{T} = \begin{bmatrix}x_1 & x_2 & x_3\\ y_1 & y_2 & y_3\\ z_1 & z_2 & z_3\end{bmatrix} - \begin{bmatrix}x_0\\ y_0\\ z_0\end{bmatrix} \tag{12}$$

In (11) and (12) x_p, y_p, z_p is the position of the point under consideration, x_0, y_0, z_0 is the antenna position (the shadowing must be calculated for both the transmitting and receiving antennas) and $x_{1...3}, y_{1...3}, z_{1...3}$ are the coordinates of the face corners. Shadowing occurs when

$$(\forall n = 1, 2, 3 : \lambda_n \geq 0) \wedge \sum_{n=1,2,3} \lambda_n \geq 1 \tag{13}$$

It must be noted that in the case where two faces share the same edge and a point under consideration belonging to one face is exactly on the edge, it will be falsely shadowed by the other face. This could be avoided by replacing the comparison $\geq$ with $>$ in (13); however, this would result in the points lying directly behind the face edge from the antenna's perspective being falsely not shadowed. Both of these situations are very unlikely, but this must be taken into account for very specific simulation scenarios.

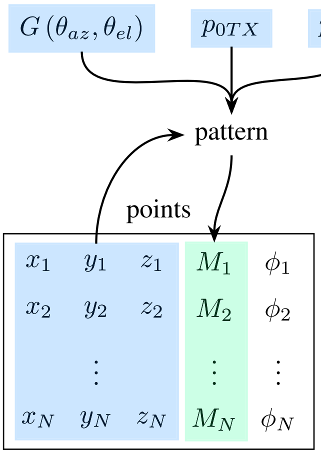
$G(\theta_{az}, \theta_{el})$
p_{0TX}
p_{0RX}
pattern
points
x_1 y_1 z_1 M_1 ϕ_1
x_2 y_2 z_2 M_2 ϕ_2
x_N y_N z_N M_N ϕ_N

FIGURE 8. The dataflow of the antenna pattern.

G. ANTENNA POSITION, ORIENTATION, AND BEAM PATTERN

If the antenna is not omnidirectional, its beam pattern must be included in the simulation. The gain pattern is specified as a function

$$G(\theta_{az}, \theta_{el}) \tag{14}$$

where θ_{az} and θ_{el} are the azimuth and elevation angles of a scatterer under consideration with respect to the antenna position. The procedure is presented in Figure 8. The function itself is defined by the simulator user, and the angles are calculated as follows:

$$\theta_{az} = \arcsin\left(\frac{p''_x}{\|\mathbf{p}''\|}\|\right) \tag{15}$$

$$\theta_{el} = \arcsin\left(\frac{p''_y}{\|\mathbf{p}''\|}\|\right) \tag{16}$$

where $\mathbf{p}'' = p''_x, p''_y, p''_z$ represent the scatterer coordinates in the antenna reference system:

$$p''_x = (\mathbf{p} - \mathbf{p_0}) \cdot \mathbf{i''_x} \tag{17}$$

$$p''_y = (\mathbf{p} - \mathbf{p_0}) \cdot \mathbf{i''_y} \tag{18}$$

$$p''_z = (\mathbf{p} - \mathbf{p_0}) \cdot \mathbf{i''_z} \tag{19}$$

and the vectors $\mathbf{i_x}''$, $\mathbf{i_y}''$, $\mathbf{i_z}''$ represent the antenna reference system:

$$\mathbf{i''_x} = \mathbf{i'_x} \cos\theta_r + \mathbf{i'_y} \sin\theta_r \tag{20}$$

$$\mathbf{i''_y} = -\mathbf{i'_x} \sin\theta_r + \mathbf{i'_y} \cos\theta_r \tag{21}$$

$$\mathbf{i''_z} = \mathbf{i'_z} \tag{22}$$

where the vectors $\mathbf{i'_x}, \mathbf{i'_y}, \mathbf{i'_z}$ are obtained by rotating the unit vectors $\mathbf{i_x}, \mathbf{i_y}, \mathbf{i_z}$ so that the $\mathbf{i_z}$ vector is rotated onto the $\mathbf{d_r}$ vector, where the unambiguous situation $\mathbf{d_r} = -\mathbf{i_z}$ is interpreted as the rotation of the antenna in the elevation direction:

$$\mathbf{i'_x} = \begin{cases} \mathbf{i_x}, & \text{if } \mathbf{d_r} = -\mathbf{i_z} \\ \boldsymbol{R}\mathbf{i_x}, & \text{otherwise} \end{cases} \tag{23}$$

$$\mathbf{i'_y} = \begin{cases} \mathbf{i_y}, & \text{if } \mathbf{d_r} = -\mathbf{i_z} \\ \boldsymbol{R}\mathbf{i_y}, & \text{otherwise} \end{cases} \tag{24}$$

$$\mathbf{i'_z} = \mathbf{d_r} \tag{25}$$

where $\boldsymbol{R}$ is the rotation matrix that corresponds to the rotation of the vector $\mathbf{i_z}$ onto the vector $\mathbf{d_r}$ [69]:

$$\boldsymbol{R} = \boldsymbol{I} + [\mathbf{v}]_\times + [\mathbf{v}]^2_\times \frac{1}{1+c} \tag{26}$$

where:

$$\mathbf{v} = \mathbf{i_z} \times \mathbf{d_r} \tag{27}$$

$$c = \mathbf{i_z} \cdot \mathbf{d_r} \tag{28}$$

$$[\mathbf{v}]_\times \triangleq \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{29}$$

and $\boldsymbol{I}$ is the diagonal matrix. For reference, see Figure 2.

V. RESULTS

The result of this work is a fully functional, open-source SAR simulator. This chapter presents a showcase of some results the simulator can produce. Figure 9 presents a scene with three scatterers that was used for a single pulse simulation. The resulting response is presented in Figure 10. The simulated radar parameters are f_0 $=1$ *GHz* and B $=$ 10 *MHz*,

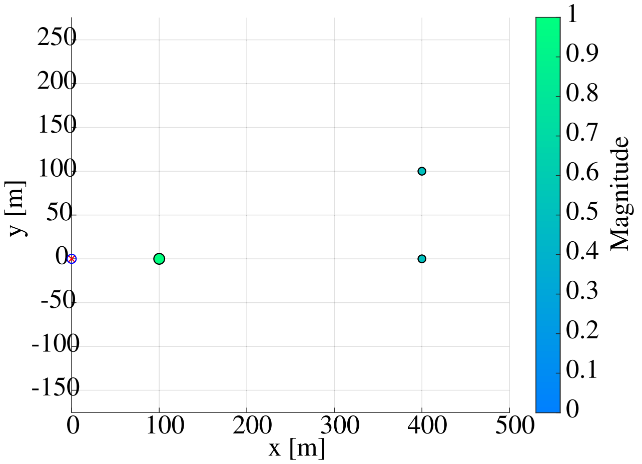

FIGURE 9. Scene definition for a single pulse simulation.

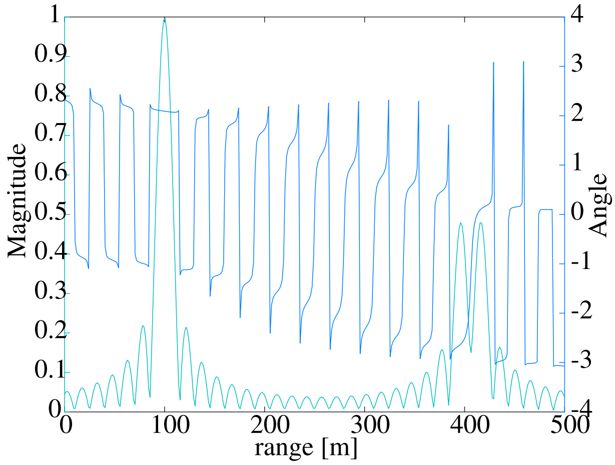

FIGURE 10. Result of a single pulse simulation.

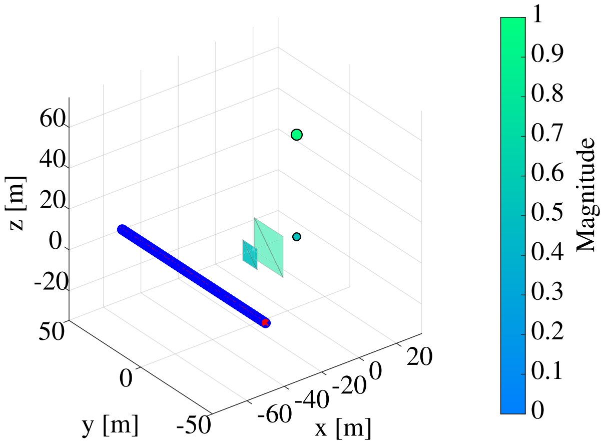

FIGURE 11. Scene definition for a simple 3D simulation.

and the targets at $x = 400\ m$ have opposite phases. This experiment confirms the proper operation of the simulator.

The next experiment was fully three-dimensional, the scene of which is depicted in Figure 11. The scene consists of two flat reflecting surfaces and two isotropic scatterers. The first $10\,m$ by $10\,m$ plate is located at $x = -20\,m$, is almost mirror-like ($\rho = 0.1$) and almost opaque ($\sigma = 0.1$). The other plate is bigger ($20\ m$ by $20\ m$), located at $x = -10\ m$, has a rough surface ($\rho = 0.8$), and is moderately transparent ($\sigma = 0.5$). The first scatterer is located at $x = 5\ m, y = 0\ m, z = 0\ m$ and the other directly above it at $z = 50\ m$. The radar carrier was moved at $x = -50\ m, z = 0\ m$ from $y = -50\ m$ to $y = 50\ m$. Two antenna beam patterns were used with this scene: one with a wide azimuth angle and narrow elevation angle, and the other with a narrow azimuth angle and wide elevation angle. The resulting raw data are presented in Figures 12 and 13, respectively. The antenna beam pattern influence is visible: in Figure 12, both surfaces are visible at smaller aspect angles compared to Figure 13, and the echo from the second scatterer is much weaker in the former Figure, as it is outside the main lobe of the antenna.

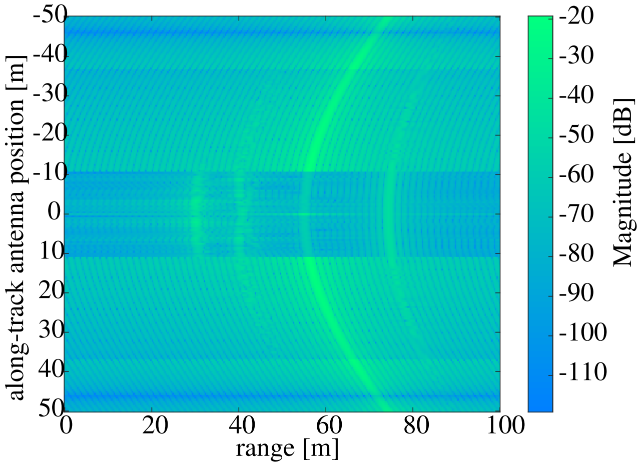

FIGURE 12. Raw radar data from a simple 3D simulation with wide azimuth angle and narrow elevation angle.

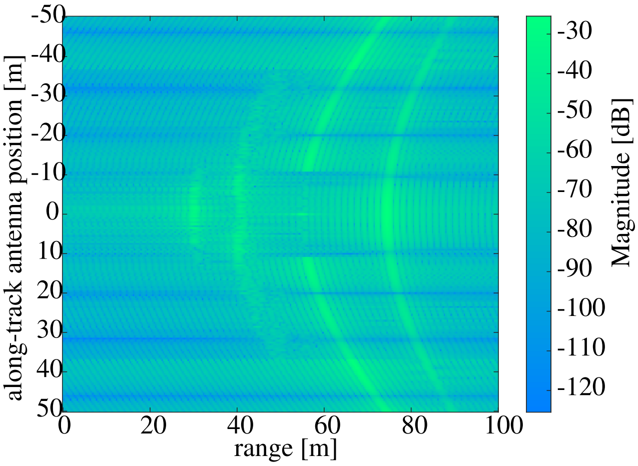

FIGURE 13. Raw radar data from a simple 3D simulation with narrow azimuth angle and wide elevation angle.

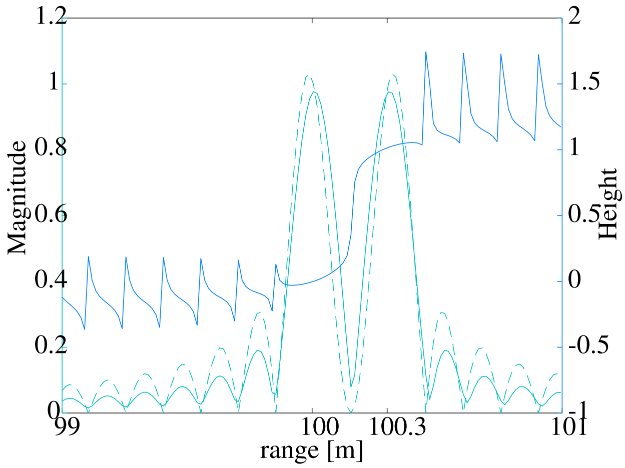

FIGURE 14. Result of an interferometric simulation.

Another experiment demonstrates the correctness of phase generation. The simulated radar parameters are $f_0 = 10\ GHz$ and $B = 1\ GHz$, and there are two targets at $x = 100\ m,\ z = 0\ m$ and $x = 100.3\ m,\ z = 1\ m$. Two antennas are placed at $x = 0\ m,\ z = 0\ m$ and $x = 0\ m,\ z = 1\ m$. Figure 14 shows the amplitude of the received signal of the first (solid line) and the second (dashed line) antenna on the left axis and the interferometric height on the right axis. The interferometric height is calculated using the formula:

$$h = \frac{\lambda}{4\pi} r \frac{\phi}{d}, \tag{30}$$

where

$$\phi = \angle\left(E_1 E_2^*\right) \tag{31}$$

is the interferometric phase and d is the distance between the antennas. The height difference between the two scatterers was correctly estimated.

To demonstrate the Doppler capability and illustrate the range-Doppler ambiguity, a simple experiment was performed. The simulated radar parameters are the same as in the previous experiment, and three scatterers are present at $x = 100\,m$ and velocities $-200,\ 0,\ 100\,m/s$. The antenna is at $x = 0\,m$. The result is presented in Figure 15, and the shift in the range is visible.

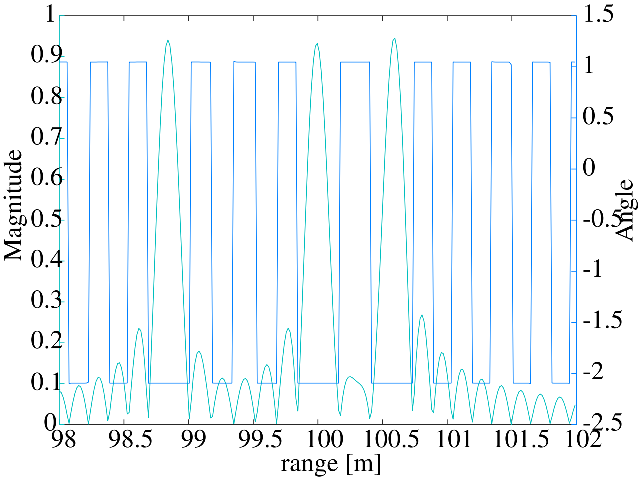

FIGURE 15. Result of a Doppler simulation.

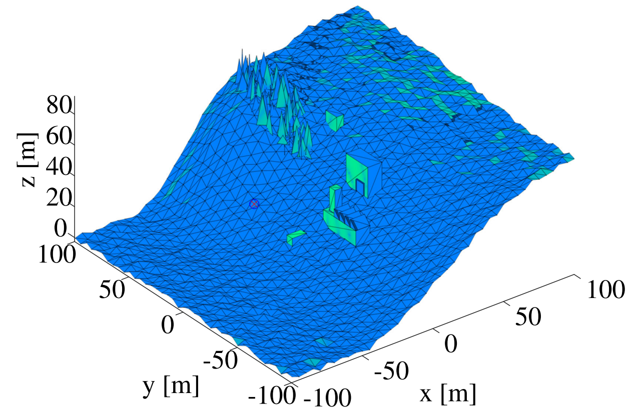

FIGURE 16. A 3D model of a complex scene with superimposed reflected energy values for a single antenna position marked with a circle.

The last experiment shows the simulator's ability to handle complex scenes. For this purpose, a scene of an uneven mountainside with some buildings and trees was defined. Figure 16 presents the scene with superimposed reflected energy. The shadow of the chimney of the factory-shaped building is visible on the building behind it, as well as the shadows of the trees. It can be seen that the reflections from the buildings are stronger than the reflections from the ground or trees. The antenna pattern is visible on the building's facade.

VI. LICENSE CONSIDERATIONS

An important decision to make when releasing open-source software is the license. There are three main types of open-source licenses [70]:

- *Free-for-all* or *academic* licenses that do not limit the usage of the software and its derivatives, other than requiring giving credit to the original author.
- *Keep-open* licenses that require derivatives to be released as open-source but permit the software to be a part of a bigger proprietary project.
- *Share-alike* or *copyleft* licenses that require any derivative or extension of the software to be released under the same or compatible license and prohibit commercial use.

Based on the above considerations, the author has decided to use the *Apache License* [71]. This allows anyone to use the software freely, and even the industry can use the software as a basis for a commercial product. The author hopes that in such a case, at least part of the improvements and modifications can be released as open-source. The *Apache License* is *copyleft compatible*, so contributors supporting the *copyleft* philosophy may release their improvements under, for example, GPL License [72].

VII. CONCLUSION

A. SUMMARY

The simulator, although not fully-featured, is a basic tool for SAR researchers. The functionality demonstrated by the results includes:

- 3D scene definition with faces and points
- surface roughness and transparency
- antenna beam pattern
- Doppler effect for GMTI

which is enough for many applications. The source code was made available to the scientific community [73].

B. LIMITATIONS

As simplicity was one of the key principles, the simulator lacks complex features, such as error and noise simulation. The noise and errors, however, can be added at any step of the response calculation if desired – see Figure 1. Another potentially interesting feature that is missing from the simulator is the clutter simulation. Ground clutter can be implemented by appropriately defining the ground surface or by adding small objects simulating vegetation. Sea clutter, however, is a more complex phenomenon [74], and, in the author's opinion, based on the results of the assumed models, it cannot be reliably and accurately modeled using this tool. Polarimetry is also not supported by the simulator, as polarimetric features of the scene depend on various factors, such as shape, orientation, and roughness, but also volumetric scattering mechanisms, such as humidity and internal structure [58]. Implementing those would make the simulator overly complicated – see [3]. There is, however, a workaround – if HH, VV and HV parameters of the scene are known, the resulting magnitude M can be calculated for each case, and the simulation can be executed three times to obtain polarimetric data.

C. FUTURE WORK

This framework should be developed further to include more features, including:

- volumetric scattering
- polarity support
- multiple reflections

Additionally, the simulator should be optimized to reduce the required memory and computational power, especially since widening the features list will inevitably increase computational complexity.

ACKNOWLEDGMENT

The author would like to thank the anonymous reviewers, whose insightful comments not only shaped the current form of this article and largely contributed to the development of the simulator but also significantly influenced the author himself.

REFERENCES

[1] S. Auer, S. Gernhardt, S. Hinz, N. Adam, N. Adam, R. Bamler, and R. Bamler, ''Simulation of radar reflection at man-made objects and its benefits for persistent scatterer interferometry,'' in *Proc. 7th Eur. Conf. Synth. Aperture Radar*, Jun. 2008, pp. 1–4.

[2] S. Auer, ''3d synthetic aperture radar simulation for interpreting complex urban reflection scenarios,'' Ph.D. dissertation, Dept. Civil Eng. Surveying, Techn. Univ. München, München, Germany, Mar. 2011.

[3] M. L. Williams, ''Polsarprosim—A coherent, polarimetric SAR simulation of forests for polsarpro. Design document and algorithm specification,'' Eur. Space Agency, Paris, France, Tech. Rep., 2006.

[4] J. W. Zhao, J. C. Wu, X. L. Ding, L. Zhang, and F. M. Hu, ''Analysis of multipath pixels in sar images,'' *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vols. XLI–B7, pp. 873–877, Dec. 2016. [Online]. Available: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLI-B7/873/2016/

[5] J. Guo, B. Lei, C. Ding, and Y. Zhang, ''Synthetic aperture radar image synthesis by using generative adversarial nets,'' *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 1111–1115, Jul. 2017.

[6] S. Niu, X. Qiu, L. Peng, and B. Lei, ''Parameter prediction method of sar target simulation based on convolutional neural networks,'' in *Proc. EUSAR 12th Eur. Conf. Synth. Aperture Radar SAR*, Jun. 2018, pp. 1–5.

[7] D. C. Mason, S. L. Dance, S. Vetra-Carvalho, and H. L. Cloke, ''Robust algorithm for detecting floodwater in urban areas using synthetic aperture radar images,'' *J. Appl. Remote Sens.*, vol. 12, Nov. 2018, Art. no. 045011, doi: 10.1117/1.JRS.12.045011.

[8] J. Tao, G. Palubinskas, and P. Reinartz, ''Automatic interpretation of high resolution SAR images: First results of SAR image simulation for single buildings,'' in *Proc. ISPRS Hannover Workshop*, 2011, pp. 14–17.

[9] S. Tajbakhsh, M.-J. Kim, H. M. Berenyi, and R. E. Burge, ''Images of urban areas by a synthetic aperture radar simulator,'' *Proc. SPIE SAR Data Process. Remote Sens.*, vol. 2316, pp. 290–300, Dec. 1994, doi: 10.1117/12.197549.

[10] H.-J. Mametsa, F. Rouas, A. Berges, and J. Latger, ''Imaging radar simulation in realistic environment using shooting and bouncing rays technique,'' *Proc. SPIE SAR Image Anal., Model., Techn.*, vol. 4543, pp. 34–40, Jan. 2002, doi: 10.1117/12.453975.

[11] T. Balz, ''Real-time SAR simulation of complex scenes using programmable graphics processing units,'' in *Proc. ISPRS TCVII Mid-term Symp.*, Enschede, The Netherlands, 2006.

[12] C. Cochin, P. Pouliguen, B. Delahaye, D. L. Hellard, P. Gosselin, and F. Aubineau, ''MOCEM-An'all in one'tool to simulate SAR image,'' in *Proc. 7th Eur. Conf. Synth. Aperture Radar*, Jun. 2008, pp. 1–4.

[13] J. Meyer-Hilberg, C. Neumann, and H. Senkowski, ''Gmti systems simulation using the sar simulation tool pirdis,'' in *Proc. 7th Eur. Conf. Synth. Aperture Radar*, Jun. 2008, pp. 1–4.

[14] T. Balz and U. Stilla, ''Hybrid GPU-based single-and double-bounce SAR simulation,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 10, pp. 3519–3529, Oct. 2009.

[15] H. Hammer and K. Schulz, ''Coherent simulation of SAR images,'' *Proc. SPIE Image and Signal Process. Remote Sens.*, vol. 7477, Sep. 2009, Art. no. 74771G, doi: 10.1117/12.830380.

[16] H. Hammer and K. Schulz, ''SAR-simulation of large urban scenes using an extended ray tracing approach,'' in *Proc. Joint Urban Remote Sens. Event*, Apr. 2011, pp. 289–292.

[17] G. Wang, F. Zhang, W. Zi, and Y. Shao, ''High resolution sar image simulation for buildings based on ray tracing algorithm,'' *Proc. SPIE Remote Sens. Image Process., Geographic Inf. Syst. Appl.*, vol. 8006, Nov. 2011, Art. no. 80060G, doi: 10.1117/12.898789.

[18] H. Hammer and K. Schulz, ''Sar simulation for urban scene analysis,'' in *Proc. 9th Eur. Conf. Synth. Aperture Radar*, Apr. 2012, pp. 91–94.

[19] H. Chen, Y. Zhang, H. Wang, and C. Ding, ''SAR imaging simulation for urban structures based on analytical models,'' *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 6, pp. 1127–1131, Nov. 2012.

[20] G. D. Martino, A. Iodice, D. Riccio, and G. Ruello, ''A physical approach for sar speckle simulation: First results,'' *Eur. J. Remote Sens.*, vol. 46, no. 1, pp. 823–836, 2013, doi: 10.5721/EuJRS20134649.

[21] T. Zeng, C. Hu, H. Sun, and E. Chen, ''A novel rapid SAR simulator based on equivalent scatterers for three-dimensional forest canopies,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5243–5255, Sep. 2014.

[22] X. Weijie and W. Xiaojie, ''Sar image simulation and verification for urban structures,'' *Int. J. Electron.*, vol. 103, no. 2, pp. 247–260, 2016, doi: 10.1080/00207217.2015.1036319.

[23] G. Franceschetti, M. Migliaccio, D. Riccio, and G. Schirinzi, ''SARAS: A synthetic aperture radar (SAR) raw signal simulator,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 30, no. 1, pp. 110–123, Jan. 1992.

[24] G. Franceschetti, R. Marino, M. Migliaccio, and D. Riccio, ''SAR simulation of three-dimensional scenes,'' *Proc. SPIE SAR Data Process. Remote Sens.*, vol. 2316, pp. 192–201, Dec. 1994, doi: 10.1117/12.197539.

[25] F. Klaus, ''SiSAR: Advanced SAR simulation,'' *Proc. SPIE Synth. Aperture Radar Passive Microw. Sens.*, vol. 2584, pp. 391–399, Nov. 1995, doi: 10.1117/12.227150.

[26] E. Boerner, F. H. Uhlmann, B. Grafmueller, R. Zahn, and H. Braumann, ''SARIS: Synthetic aperture radar instrument simulator,'' in *Proc. Scanning Present Resolving Future. Proc.*, Jul. 2001, pp. 1598–1600.

[27] S. R. J. Axelsson, ''Fast simulation of SAR raw data from complex scenes,'' *Proc. SPIE SAR Image Anal., Model. Techn.*, vol. 4543, pp. 188–197, Jan. 2002, doi: 10.1117/12.453968.

[28] G. Margarit, J. J. Mallorqui, J. M. Rius, and J. Sanz-Marcos, ''On the usage of GRECOSAR, an orbital polarimetric SAR simulator of complex targets, to vessel classification studies,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 12, pp. 3517–3526, Dec. 2006.

[29] J. Allan and M. Collins, ''Design and testing of a java-based digital SAR signal simulation system,'' in *Proc. IEEE Int. Symp. Geosci. Remote Sens.*, Jul. 2006, pp. 3204–3207.

[30] M. Wang, D. Liang, H. Huang, and Z. Dong, ''SBRAS—An advanced simulator of spaceborne radar,'' in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2007, pp. 4942–4944.

[31] J. M. Allan and M. J. Collins, ''SARSIM: A a digital SAR simulation systems,'' in *Proc. Challenges Earth Observ., Sci., Tech. Commercial*, Sep. 2007, pp. 393–398.

[32] C. H. Jung, M. S. Choi, and Y. K. Kwag, ''Parameter based SAR simulator for image quality evaluation,'' in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2007, pp. 1599–1602.

[33] T. Kim, J. Yoon, H. Lee, and S. Min, ''Raw data generation of sar simulation using scanline method,'' *Proc. SPIE Image Signal Process. Remote Sens.*, vol. 7109, Oct. 2008, Art. no. 71091J, doi: 10.1117/12.800417.

[34] S. Khwaja, ''Fast raw data generation of realistic environments for a SAR system simulator,'' Ph.D. dissertation, Signal Image Process., Univ. Rennes 1, Rennes, France, 2008.

[35] Z. Xiang, K. Wang, X. Liu, and W. Yu, ''A multilevel interferometric SAR simulator,'' in *Proc. 2nd Asian-Pacific Conf. Synth. Aperture Radar*, Oct. 2009, pp. 395–399.

[36] R. Dumont, C. Guedas, E. Thomas, F. Cellier, and G. Donias, ''Dionisos. an end-to-end sar simulator,'' in *Proc. 8th Eur. Conf. Synth. Aperture Radar*, Jun. 2010, pp. 1–4.

[37] D. Reynald, G. Christophe, and D. Jérêmy, ''Automatic generation of networks for sar simulation of moving targets,'' in *Proc. 9th Eur. Conf. Synth. Aperture Radar*, Apr. 2012, pp. 316–319.

[38] G. Franceschetti, A. Iodice, A. Natale, and D. Riccio, ''Bistatic SAR simulation: Time and frequency domain approaches,'' in *Proc. 17th Int. Conf. Digit. Signal Process. (DSP)*, Jul. 2011, pp. 1–7.

[39] Q. Chen, A. Yu, Z. Sun, and H. Huang, ''A multi-mode space-borne SAR simulator based on SBRAS,'' in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2012, pp. 4567–4570.

[40] B. J. Doring, P. R. Looser, M. Jirousek, and M. Schwerdt, ''Reference target correction based on point-target SAR simulation,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 951–959, Mar. 2012.

[41] K. S. Kulpa, P. Samczyński, M. Malanowski, A. Gromek, D. Gromek, W. Gwarek, B. Salski, and G. Taski, ''An advanced SAR simulator of three-dimensional structures combining geometrical optics and full-wave electromagnetic methods,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 776–784, Jan. 2014.

[42] W. Li, H. Zhang, H. P. Hildre, and J. Wang, ''An FPGA-based real-time UAV SAR raw signal simulator,'' *IEICE Electron. Exp.*, vol. 11, no. 11, 2014, Art. no. 20140168.

[43] Y. M. R. Sangheun Shim and S. Kim, ''High resolution image formation method based on the realistic spaceborne SAR modeling and simulation,'' *Proc. SPIE Image Anal., Model., Techn.*, vol. 9243, Oct. 2014, Art. no. 924316, doi: 10.1117/12.2067176.

[44] A. Reppucci, J. Márquez, V. Cazcarra, and G. Ruffini, ''Advanced sar simulator with multi-beam interferometric capabilities,'' *Proc. SPIE Image Anal., Model., Techn.*, vol. 9243, Oct. 2014, Art. no. 924303, doi: 10.1117/12.2067201.

[45] J. M. Allan, M. J. Collins, and C. Gierull, ''Computational synthetic aperture radar (CSAR): A flexible signal simulator for multichannel SAR systems,'' *Can. J. Remote Sens.*, vol. 36, no. 4, pp. 345–360, 2010, doi: 10.5589/m10-048.

[46] L. Yang, W. Yu, S. Zheng, and L. Zhang, ''Efficient bistatic SAR raw signal simulator of extended scenes,'' *Int. J. Antennas Propag.*, vol. 2014, Jan. 2014, Art. no. 130784.

[47] X. Yinhui, Z. Dazhi, Y. Tao, and X. Xiaoheng, ''A real-time SAR echo simulator based on FPGA and parallel computing,'' *Telkomnika*, vol. 13, no. 3, pp. 806–812, 09 2015.

[48] E. J. Balster, F. A. Scarpino, A. M. Kordik, and K. L. Hill, ''Synthetic aperture radar imaging simulator for pulse envelope evaluation,'' *J. Appl. Remote Sens.*, vol. 11, Dec. 2017, Art. no. 046022, doi: 10.1117/1.JRS.11.046022.

[49] A. Rahmanizadeh and J. Amini, ''An integrated method for simulation of synthetic aperture radar (SAR) raw data in moving target detection,'' *Remote Sens.*, vol. 9, no. 10, p. 1009, Sep. 2017, doi: 10.3390/rs9101009.

[50] Y. Liu, W. Wang, X. Pan, Z. Gu, and G. Wang, ''Raw signal simulator for SAR with trajectory deviation based on spatial spectrum analysis,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6651–6665, Nov. 2017.

[51] G. Di Martino, A. Iodice, D. Poreh, and D. Riccio, ''Pol-SARAS: A fully polarimetric SAR raw signal simulator for extended soil surfaces,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2233–2247, Apr. 2018.

[52] Y. A. Ilyushin, R. Orosei, O. Witasse, and B. Sánchez-Cano, ''CLUSIM: A synthetic aperture radar clutter simulator for planetary exploration,'' *Radio Sci.*, vol. 52, no. 9, pp. 1200–1213, 2017. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2017RS006265

[53] G. Franceschetti, M. Migliaccio, and D. Riccio, ''The SAR simulation: An overview,'' in *Proc. Int. Geosci. Remote Sens. Symp.*, Dec. 1995, pp. 2283–2285.

[54] T. Balz, H. Hammer, and S. Auer, ''Potentials and limitations of sar image simulators—A comparative study of three simulation approaches,'' *ISPRS J. Photogramm. Remote Sens.*, vol. 101, pp. 102–109, Dec. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924271614002834

[55] D. C. Munson and R. L. Visentin, ''A signal processing view of strip-mapping synthetic aperture radar,'' *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 12, pp. 2131–2147, Dec. 1989.

[56] C. Lopez-Martinez, A. Alonso, X. Fabregas, and K. P. Papathannassiou, ''Analysis of volumetric scatters based on TanDEM-X polarimetric interferometric SAR data,'' in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2011, pp. 2574–2577.

[57] P. Imperatore, A. Iodice, and D. Riccio, ''Volumetric and interfacial inhomogeneities of random semi-infinite media: A unified perturbative scattering model,'' in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2012, pp. 5794–5797.

[58] M. Skolnik, *Radar Handbook* (Electronics Electrical Engineering). New York, NY, USA: McGraw-Hill, 2008.

[59] D. C. Munson, J. D. O'Brien, and W. K. Jenkins, ''A tomographic formulation of spotlight-mode synthetic aperture radar,'' *Proc. IEEE*, vol. 71, no. 8, pp. 917–925, Aug. 1983.

[60] C. Ozdemir, *Inverse Synthetic Aperture Radar Imaging With MATLAB Algorithms* (Microwave and Optical Engineering). Hoboken, NJ, USA: Wiley, 2012.

[61] S. Auer, T. Balz, S. Becker, and R. Bamler, ''3D SAR simulation of urban areas based on detailed building models,'' *Photogramm. Eng. Remote Sens.*, vol. 76, no. 12, pp. 1373–1384, Dec. 2010. [Online]. Available: https://elib.dlr.de/66847/

[62] G. P. Zouros and G. C. Kokkorakis, ''Electromagnetic scattering by a general rotationally symmetric inhomogeneous anisotropic sphere,'' *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 10, pp. 3054–3065, Oct. 2015.

[63] A. A. Sakr, E. A. Soliman, and A. K. Abdelmageed, ''A surface integral equation formulation for electromagnetic scattering from a conducting cylinder coated with multilayers of homogeneous materials,'' *J. Appl. Phys.*, vol. 116, no. 5, Aug. 2014, Art. no. 054902.

[64] T.-K. Chan, Y. Kuga, A. Ishimaru, and C. T. C. Le, ''Experimental studies of bistatic scattering from two-dimensional conducting random rough surfaces,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 34, no. 3, pp. 674–680, May 1996.

[65] R. D. Scattering, *Scattering, Natural Surfaces, and Fractals*. New York, NY, USA: Academic, 2006.

[66] P. Imperatore, A. Iodice, and D. Riccio, ''On the regime of validity of volumetric and boundary perturbation -based scattering models for rough multilayer,'' in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2012, pp. 3198–3201.

[67] J. Embrechts, ''Light scattering by rough surfaces: Electromagnetic model for lighting simulations,'' *Lighting Res. Technol.*, vol. 24, no. 4, pp. 243–254, 1992, doi: 10.1177/096032719202400408.

[68] *The Phong Model, Introduction to the Concepts of Shader, Reflection Models and BRDF*. Accessed: Nov. 13, 2019. [Online]. Available: https://www.scratchapixel.com/lessons/3d-basic-rendering/phong-shader-BRDF

[69] J. van den Berg. *Calculate Rotation Matrix to Align Vector A to Vector B in 3D*. Accessed: Sep. 1, 2016. [Online]. Available: https://math.stackexchange.com/q/476311

[70] A. Engelfriet, ''Choosing an open source license,'' *IEEE Softw.*, vol. 27, no. 1, pp. 48–49, Jan. 2010.

[71] Apache Software Foundation. (2004). *Apache License, Version 2.0*. Accessed: Feb. 13, 2019. [Online]. Available: http://www.apache.org/licenses/LICENSE-2.0

[72] Free Software Foundation. (2002). *Gnu General Public License, Version 3*. Accessed: Feb. 13, 2019. [Online]. Available: https://www.gnu.org/licenses/gpl.html

[73] J. Drozdowicz. (2019). *3D SAR-SIM GitHub Repository*. [Online]. Available: https://github.com/JDrozdowicz/3dsar-sim

[74] S. Kemkemian, J.-F. Degurse, V. Corretja, and R. Cottron, ''Sea clutter modelling for space-time processing,'' in *Proc. 17th Int. Radar Symp. (IRS)*, May 2016, pp. 1–6.

JEDRZEJ DROZDOWICZ (Graduate Student Member, IEEE) received the B.S. and M.S. degrees (Hons.) in electronics and computer engineering from the Warsaw University of Technology (WUT), Warsaw, Poland, in 2011 and 2015, respectively, where he is currently pursuing the Ph.D. degree in telecommunications.

From 2011 to 2014, he was an Electronics Engineer with Robotics Inventions. Since 2014, he has been with the Radar Techniques Research Group, Institute of Electronic Systems, WUT, where he has been a Research Assistant with the Faculty of Electronics and Information Technology, since 2018. He has authored more than 30 articles. His research interests include interferometric SAR signal processing, three-dimensional radar imaging, and FPGA-based systems design for radar applications.