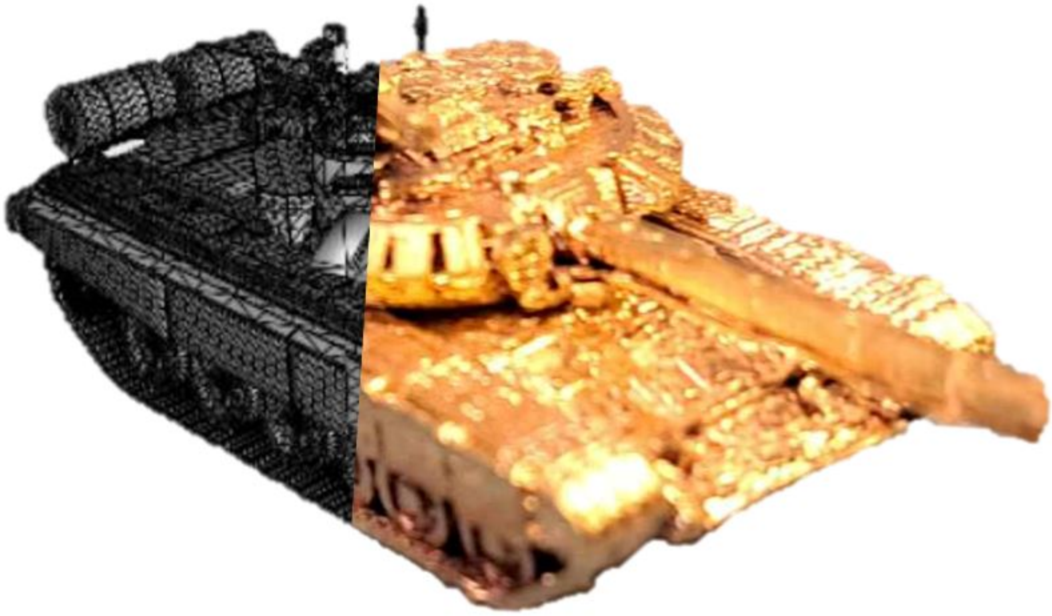
3rd and 4th SEMESTER

SYNTHETIC APERTURE RADAR
Simulations and Terahertz Applications

30th May 2025



Master's Programme in Physics and Technology, 5.319b
Department of Materials and Production, Aalborg University

AALBORG UNIVERSITY

w/
Department of Materials and Production
Master's Programme in Physics and Technology
Fibigerstræde 16
9220 Aalborg Øst

Title:
Synthetic Aperture Radar Simulations and Terahertz Applications

Project:
3rd and 4th semester

Project period:
5th September 2024 to 30th May 2025

Project group:
5.319b

Supervisor:
Esben Skovsen (AAU)
Therkel Zøllner Olesen (Terma)
Peter Lysdahl Bæk (Terma)

Number of pages:
117

Number of appendices:
2

Finish date:
30th May 2025

Abstract:

Synthetic aperture radar (SAR) is an established method at radio frequencies for long range and high spatial resolution imaging. Its use in reconnaissance include terrain mapping and object identification. In this report we showcase a ray-tracing engine based on a bounding volume hierarchy (BVH), for generating SAR images. The computation time of the engine is reduced by utilizing NVIDIA Optix, allowing for fast SAR image simulations (~30s pr. image). The resulting SAR images of armored vehicles closely match those of the MSTAR dataset. A setup for inverse SAR in the terahertz (THz) domain is used to measure models at reduced scales, for easy comparison with radar SAR. Models imaged include a T72 tank, a pickup truck and a boat. The resulting THz SAR images are shown to have a spatial resolution comparable to the diffraction limit of the THz radiation, at a working distance of 80 cm. SAR polarimetry and interferometry was tested in the THz domain. The phase drift of the system was deemed too high for good interferometry SAR (inSAR). Promising THz polarimetry SAR was showcased. Simulations of ray-tracing based polarimetry and interferometry was implemented and shown. All SAR image formatting is done with the back projection algorithm (BPA), for which an entropy based auto focus method is implemented.

Preface

This project is written in collaboration with Terma, under the supervision of Therkel Zøllner Olesen (Terma), Peter Lysdahl Bæk (Terma) and Esben Skovsen (AAU). Together we have investigated methods of simulating and formatting synthetic aperture radar (SAR) along with experimental measurements of 1:100 scale scenes with terahertz SAR.

The project starts with an introduction to the basic principles of radar and the concept of SAR. Then theory for the backprojection algorithm, ray-tracing simulation, interferometric SAR, and polarimetric SAR is presented. Then follows the methodology behind the implementation of GPU in ray-tracing and the THz setup, combined with an analysis of the setup parameters. The results are split into four different topics: SAR microscopy, computer simulations, reduced scale models, and SAR techniques. After that the bibliography and the appendix can be found.

Python[1] was used as the main tool for plotting and calculating computational methods. Furthermore, NVIDIA OptiX ray-tracing engine was used for simulations[2]. Generative AI have been used for assistance in writing code for computer vision software for interferometry, and for assistance on a few tikz pictures.

The IEEE citation style has been used for references and all references are summarized in the bibliography. Figures are named "X.X" [example figure 2.1], equations are named "X.X" [example equation 4.11], vectors are shown with **bold** font and matrices are shown with $\bar{\bar{\mathbf{A}}}$. Figures with no citation are made by the project group. All the sources are in the bibliography. In the case, where the reference are before the periods, the reference only refer to the associated sentence. References after the periods are associated with the whole paragraph.

Gustav Møller Thomsen
gthoms20@student.aau.dk

Laurits Dyhre
ldyhre20@student.aau.dk

Malthe Carl Hougaard Vang
mvang20@student.aau.dk

Aalborg University, 30th May 2025

Resumé

Synthetic Aperture Radar (SAR) er en afbildningsmetode, der anvender billede-formaterings algoritmer til at danne 2D eller 3D billeder af områder eller objekter ud fra reflektionsdata fra et spænd af vinkler. SAR er et af hovedanvendesesområderne i moderne radar teknologi. Dets vigtighed inden for fagområderne geodesy og radar rekognoscering er beskrevet i meget litteratur. SAR finder sin plads som afbildningsmetode grundet de unikke fordele den bærer ift. andre metoder. Pixelopløsningen af optiske sensorer afhænger ofte kraftigt af distancen til det object der bliver afbilledet. Dette er ikke det samme problem for SAR, da pixel opløsningen i stedet afhænger af naturen af det udsendte signal.

Ligesom mange andre felter finder SAR anvendelser det utrolig gunstigt at indkorporere machine learning ind i dets rammer. Dette vil være for automatisk genkendelse af objekter i SAR billeder såsom biler, både og andre køretøjer. For at kunne træne en machine learnig algoritme, skal man bruge meget data. Dette er ikke nødvendigvis noget der er let tilgængeligt. Det at anskaffe SAR billeder af virkelige omgivelser med reelle objekter kan være omkostningsfuldt, eller i nogle tilfælde umuligt. Derfor skal der være andre måder at kunne anskaffe dataet på. Der kommer simulatorer ind i billedet. En simulator skal producere et realistisk SAR billede af den 3D model som der er valgt at afbillede. Dog kan det være svært simulere data der er virkelighedsnært, hvorfor det er vigtigt at have data til validering. Her er offentlige SAR datasæt som MSTAR gode, men det er ikke nødvendigvis godt nok.

Terahertz stråling bærer mange af de samme egenskaber som radiobølger har. De vigtigste værende muligheden for at kunne måle både fase og amplituden af det indkomne signal, hvilket tillader brugen af THz stråling i SAR anvendelser. Hvor det afviger, er at THz har højere dæmpning grundet absorption af vand, og at det har muligheden for at opnå bedre pixel opløsning i SAR, grundet dets højere frekvenser. Faktisk er bølgelængderne for THz stråling omtrent en faktor 100 kortere end bølgelængderne brugt i almindelig SAR. Dette gør at vi kan måle scener i størrelsesorden 1:100 ift. virkelighed i laboratoriet. Dette fremstår altså som en god valideringsmetode for SAR simulationer.

SAR står ikke alene som afbildningsmetode. Der er flere underkategorier som hver kan noget forskelligt, men stadig baserer sig i SAR metoden. Specifikt polarimetrisk SAR, interferometrisk SAR og tomografisk SAR ser stor anvendelse inden for forskellige områder. Kompleksiteten i indførelsen af metoderne varierer, hvor nogle kræver få ændringer i SAR metoden, og andre er mere omstændige i udførelsen.

Contents

Intro 1

Radio detection and ranging, or radar, is a well developed and widely used technology. Radar development started only a few decades after Heinrich Hertz verified James Maxwell's equations experimentally in the late 19th century. It was the german Christian Hülsmeyer that invented the first devices able to detect distant metallic objects for the purpose of collision avoidance of ships in bad weather conditions. [3, 4] In the 1930s this technology had spread to multiple nations, separately developing their own applications within the aerospace, maritime, and military industry. The potential for military applications and the tension from the prior world war and the lurking presence of the second world war, caused a widespread classification of radar research.[3] The war effort boosted the development of radar technology, which led to the development of the resonant-cavity magnetron that revolutionized the radar array [5]. After the second world war the applications of radar technologies diversified into various fields. During the war the weather sensitivity of radar signals was noticed, which led to the modern weather radars[6]. Radar technology was also immediately adopted into air-traffic control, due to its well known detection capabilities[7]. Most importantly, the ability to range objects allows for mapping of surfaces spanning vast areas such as celestial bodies[8].

The modern radar is a sophisticated transducer/computer system. It emits and receives radar signals, which is a radiofrequency electromagnetic wave. The main components of a radar are the transmitter, antenna, receiver, and various components for postprocessing such as amplification and mixing. In basic operation, a signal is send out, which potentially will hit an object. This object will reflect the signal, thus returning the signal to the radar, which in turn detects the incoming radiation. The postprocessing steps convert this signal into readable data for the signal processor. [9]

The antenna is most important when considering the shape of the output radiation. Depending on the geometry of the antenna, different radiation patterns can be generated, with main and side lobes. This can be related to the spot size in optics, where the main lobe, depending on the relative intensity to the side lobes, would be the spot size of the incoming light. [9]

There are generally two types of radar waveforms, pulsed and continuous wave (CW). A pulsed radar setup will send out a sequence of radar pulses. This form is able to achieve high power signals and does not interfere with the receiving components. The CW radar sends out a continuous signal. Both forms can be frequency modulated. This modulation is usually linear with a center frequency f_0, we call this modulation chirp. The electric field of a single chirped signal can be formulated as [10]

$$E(t) = \mathrm{rect}\left(\frac{t}{T}\right) \exp\left(i\,2\pi\left(f_0 t + \frac{K}{2}t^2\right)\right), \tag{1.1}$$

where T is the time duration of the chirp, K is constant, and t is the time. The drawback of CW radar is the constant operation of the antenna for transmitting. This makes receiving difficult and limits this method mainly to bistatic configurations, where transmitter and receiver are spatially separated. [9]

Since the development of the radar started, prior to the second world war, numerous radar techniques within every field of application have been developed and investigated. Within radar reconnaissance the technique of synthetic aperture radar (SAR) is most prominent. The method of synthetic aperture can be compared to that of the numerical aperture, known from optics. Here, the numerical aperture, and wavelength, is related to the magnification. The numerical aperture of a system can be increased by in turn increasing the half-angle of the maximum cone of light that enters or exits the lens. Simply put, synthetic aperture imitates this by instead measuring the reflectivity of the scene from a span on angles, along a *flight path*, using image formatting algorithms to achieve 2D images of scene. This is widely used in radar reconnaissance and geodesy for vast areas spanning many kilometers. There are multiple approaches and techniques in SAR. In general, there are two approaches to SAR, stripmap and spotlight SAR. Stripmap SAR fixes the radar antenna to a specific position and covers the area of interest in a raster like pattern. Spotlight SAR points the antenna towards a specific scene and measures this scene along the *flight path*, this will be the approach chosen for this project, due to the simple implementation of inverse SAR (ISAR). In ISAR the scene itself is rotated where the radar platform is fixed in its position. Other types of SAR include interferometric SAR (InSAR), polarimetric SAR (polSAR) and tomographic SAR (tomoSAR). InSAR uses multiple SAR images or arrays of emitters and detectors to acquire height information and thus produce a 3D representation of the scene. TomoSAR is a further enhancement of the 3D imagery based on a multiple baseline dataset. PolSAR uses the polarization of the emitted light to gather additional information about the surface structure, geometry and material type. PolSAR enhances the ability to classify specific specific features through the analysis of the incoming polarization. [11–13]

SAR finds its place in imaging due to its unique advantages compared to optical based imaging. The pixel resolution of optical sensors are majorly affected by the distance to the object being imaged. Thus, increasing the distance will decrease the resolution. This is not the same problem for SAR, as SAR resolution is based on the nature of the signal. This allows SAR to be used far away from the target, without suffering major loss in resolution. Additionally, while being affected by weather, smoke and other obstructions in form of attenuation, the frequencies of SAR perform way better than those of optical based imaging. The radiation used in SAR is also invisible to the naked eye, and the technique provides its own source of radiation, which allows for the operation in darkness. [14]

SAR relies on a few important assumptions, which will make the image formatting possible and easier. First, the world seen through SAR is modeled as a series of point scatterers, whose reflections does not interfere with each other. These are in an environment which is assumed to be boundless, thus infinite in extend. The scatterers are also assumed to be stationary across the duration of the measurement. Lastly, the radar signal is assumed to not penetrate any reflecting surface. All these assumptions work in many scenarios, however, when broken, they will cause artifacts in the final SAR image. There are a handful of key terms used in SAR, which are essential for comprehension of the project. These, along with brief explanations, are presented in table 1.1. [11]

SAR Terminology	
Range	The distance between radar platform and center of scene
Slant Angle	The angle between the ground plane and the radar beam direction
Bandwidth	The spectral width of the radar signal
Center Frequency	The frequency of which the chirp is centered
Flight Path	The relative path of motion between the radar platform and scene

Table 1.1

In SAR applications, it is important to consider the sampling requirements and resolution of the setup. These are derived by Richards et al. [11] to be

$$\delta r = \frac{c}{2B}, \tag{1.2}$$

$$\delta x = \frac{\lambda}{2\theta_{int}}, \tag{1.3}$$

$$\delta\phi \leq \frac{\lambda}{2D}. \tag{1.4}$$

Here, c is the speed of light, δr and δx are the range and cross-range resolution, respectively. B is the bandwidth. θ_{int} is the integration angle, which is the angle made between the start and end position of the radar platform and the scene center. $\delta\phi$ is the angular separation between measurements. λ is the wavelength of the center frequency. D is the diameter of the scene. The absence of standoff range in the resolution expressions also showcase the unique advantages of the SAR technique. Furthermore, the sampling constraint for a frequency step scan follows Nyquist sampling theorem, which in our case can be shown in the form of

$$R \leq \frac{c}{2df}, \tag{1.5}$$

where, R is the unambiguous range, and df is the resolution of the frequency scan. Thus, to measure at longer distances, a higher resolution of the frequencies is necessary. [11]

With these sampling resolutions and constraints, it is possible to customize the radar parameters to fit the imaging requirements for a given scenario. Thus, SAR is a versatile method of imaging, which is used when large areas needs to be surveyed in a short timespan. As previously mentioned, to format these images certain algorithms are necessary. There are multiple algorithms used in the field, all with pros and cons. These include the range-Doppler algorithm[15], polar formatting algorithm[11] and the backprojection algorithm (BPA) [14], which can be in the frequency domain[16] or time domain form[14]. Frequency domain methods have the advantage of being computationally effective. However, these are based on assumptions that does not hold in all scenarios, like for spotlight mode SAR. This can be compensated for by modifying the initial method, but that comes with inherent artifacts, such as stretching and shifts in the position of objects. Time domain methods, specifically time domain BPA, does not suffer the same limitations, and is generally a more reliable imaging method. However, time domain BPA suffers from being more computationally expensive. This drawback has become evermore insignificant with the development in increasingly more powerful computers, and combined with its rigorous nature, makes it a powerful tool for SAR image formatting. [14]

As well established fields of research in the arts of imaging continue to develop, such as SAR, it is natural to move into new and uncharted territories within the field. For applications of electromagnetic radiation, this uncharted territory includes the terahertz (THz) band, where $1\,\text{THz} = 10^{12}\text{Hz}$. The THz band includes frequencies in the interval $[0.3; 10]$ THz.[17, 18]

Especially in radar applications[19–23], telecommunications[24, 25] and spectroscopy[26–29] the move into the THz region seems imminent. This in turn requires new technologies and techniques within the different fields as THz applications lacks the general development due to the youth of the field, and difficulty in generating THz radiation. In fact, it was with the invention of the photoconductive antenna (PCA) in the 1980s that the first means of reliant terahertz emission and detection was seen [30]. This resulted in the first commercial THz spectrometers going to market in the early 2000s [31, 32]. In general, THz emission and detection can be acquired with both time and frequency domain methods[33]. Time domain methods utilizes an ultrashort laser pulse in the order of femtoseconds (fs), which through devices such as the PCA is used to emit a longer THz pulse. These THz pulses are considered to have a wide bandwidth of up to $\approx 4\,\text{THz}$. The detection scheme utilizes a beam splitter for the fs pulse, which is sent through an optical delay line to give the ability to probe the incoming THz radiation. This scheme directly outputs the electric field of the THz radiation, which proves to be useful for SAR applications. [34]

Likewise, the frequency domain method also directly outputs the electric field. The generation and detection scheme of this method uses two CW distributed feedback lasers operating in slightly different frequencies. This when coupled together induces interference with a certain beatnote frequency. The semiconductor device in the emitter and detector is designed to only respond to the beatnote frequency, which is in the THz regime. This is elaborated on in appendix A. [34–36]

The THz band exhibits some important and exploitable properties. Which is why THz time domain spectroscopy (TDS) has seen wide applications in the fields of chemistry, material sciences, and medicine. It is important to restate its ability to directly output the electric field of the radiation. Furthermore, its interaction with materials is what makes the THz band interesting to explore. Many materials display absorption lines in the THz regime. THz can provide information on the lattice vibrations in solid materials. Non-metallic and non-polar materials, such as paper, polymers, dry food, cloth and leather, are penetrated by THz radiation[26, 27, 37–39]. This, combined with the fact that THz radiation is non-ionizing, makes THz radiation an interesting alternative to x-ray radiation. With the development of THz technology both cost, robustness and measuring speed is on par with current methods[40, 41]. THz can thus be a contender for non-destructive testing of plastics, and food and agricultural products. [40, 42–44]

Like radiowaves, THz radiation reflects off metal surfaces. The ability to directly measure the electric field makes the acquisition of in-phase quadrature data possible. These two facts allows for SAR imaging methods to be applied to the THz band. There are generally two schools of thought when generating THz radiation for the use in radar applications. One is the photonic approach, generating the THz, as previously mentioned, by lasers and semiconductor devices[19, 20]. The second is THz generation by frequency extenders from the GHz bands, which is an electronic method, due to the use of traditional components used in normal radar setups[21–23]. Both methods are valid for generating THz SAR images.

There are different types of imaging geometry. Depending on the application and available material, one may be preferred over another. In regular spotlight SAR, the geometry is determined by the flight path. This is often curved in a way, such as to point the radar platform towards the desired scene. In THz SAR, the setup is often seen in a lab, which allows for more freedom and precision in determining imaging geometry. Current literature shows linear [19], planar [20–23] and cylindrical [22] imaging geometries. The choice of geometry depends on the nature of the setup. Cylindrical geometry is much easier to implement for ISAR setups, where the radar platform usually is stationary and the scene is rotated. In lab setups, the resolution of the THz SAR system is of particular interest. As the center frequency and bandwidth increase, so does the resolution. Thus, at THz frequencies, we would expect sub-mm resolution, theoretically down to the diffraction limit of THz radiation. This can allow for THz microscopy. [21, 23]

Regular airborne SAR uses frequencies spanning a wide range including the L-band $[1; 2]$ GHz, the Ka-band $[27; 40]$ GHz, and most in between [11]. THz wavelengths are thus a factor $\approx 1 : 100$ smaller. This allows for the imaging of objects scaled down by the same factor, retaining the same resolution to size ratio. Why is that important? With the advent of machine learning (ML), the automatization of many SAR processes is possible, specifically object recognition and tracking. This is seen in geodesy applications, due to the vast amounts of data available for training.[45] However, certain applications requires SAR imagery of objects for which limited data is available, additionally generating that data would be difficult and in certain scenarios impossible. Instead, effort is being put towards creating SAR simulators, which can provide realistic SAR images of digitally made scenes and objects.[46] This can in turn provide the training data for ML algorithms. However, a validation method will still be needed for ML algorithms trained on simulated data. A reduced scale measuring setup therefore becomes useful, as the same digital models can be 3D printed and measured in a short timespan and at cheap costs.

The Aim of the Project

The aim of this project is to create a SAR simulator which can provide realistic SAR images. Additionally, the development of a synthetic aperture THz imaging system is intended, to produce reduced scale SAR images of objects of interest, and to explore the possibilities and limitations of THz microscopy. Furthermore, interferometric, polarimetric and hyperspectral SAR will be investigated using both SAR image acquisition methods.

Theory 2

2.1 Auto Focus in the Back Projection Algorithm

Multiple methods of processing data from synthetic aperture radar exist, however, in this project the time domain backprojection algorithm (BPA) has been used to format images. BPA has the advantage of being able to format images for spotlight and stripmap mode SAR and without making assumptions on the flightpath. The following section builds on Backprojection for Synthetic Aperture Radar by Duersch [14].

2.1.1 Back Projection Algorithm

In the Back Projection Algorithm we assume that a transmitted radar pulse $s(t)$ is linearly frequency modulated (LFM) on the form

$$s(t) = \sum_n A_T^n \text{rect}\left(\frac{t - t_n}{T}\right) e^{i2\pi(f_0(t-t_n)+\frac{K}{2}(t-t_n)^2+\phi_0)}, \tag{2.1}$$

where t denotes the time, t_n the time center of the pulse, A_T is the amplitude, T is the pulse duration, f_0 is the center frequency, K is the chirp rate and ϕ_0 is the initial phase. The rect function is equal to 1 if $-0.5 \leq \frac{t-t_n}{T} \leq 0.5$ and otherwise zero. To make a SAR image multiple pulses have to be emitted and measured along a flight path. The pulses emitted from different positions will be indexed by n and we will assume the motion of the moving platform is slow compared to the speed of light. This leads us to a *stop and hop* approximation where the platform position has not been changed between emitting and receiving an n'th signal.

Upon detection, the signal is typically demodulated by mixing the raw received signal with a wave oscillating with the center frequency on the form $e^{-i2\pi f_0 t}$. Hence, for an n'th platform position the measured signal for a single point target with time delay t_n will be

$$s_n(t) = A_n(t - t_n)\text{rect}\left(\frac{t - t_n}{T}\right) e^{i2\pi(-f_0 t_n+\frac{K}{2}(t-t_n)^2+\phi_0))} + \eta, \tag{2.2}$$

where η is noise. The LFM signal is spread out in the time duration T, and to obtain a optimal range resolution and better signal to noise ratio, a matched filter can be applied to obtain the range compressed signal. This is performed with a reference pulse with the LFM phase on the form

$$h_r(t) = \text{rect}\left(\frac{t}{T}\right) e^{i2\pi\frac{K}{2}t^2}. \tag{2.3}$$

The matched filter is then applied by taking a convolution between the measured signal and the complex conjugate of the reference signal

$$\begin{aligned} g_n(t) &= \int s_n(\tau) h^*(\tau - t)\, d\tau && (2.4)\\ &\approx A_n(t - t_n) H(K, T, t, t_n) e^{i2\pi(-f_0 t_n+\phi_0)}. && (2.5) \end{aligned}$$

The integral can be approximated[14], where H is proportional to a sinc function. The now range compressed signal has a phase related to the propagation time t_n and the initial phase. To perform an azimuth compression and generate a two dimensional SAR image the image plane is first divided into cells. For each of the cells a reference function is generated with the expected propagation phase from a scattering point located in the center of the cell with a expected time delay $\tilde{t_n}$ and neglecting the initial phase

$$h_a(\tilde{t_n}) = e^{-i2\pi f_0 \tilde{t_n}}. \tag{2.6}$$

The complex pixel value I for a single pixel is then found from the contribution of N pulses measured from different positions along the flightpath and the complex conjugate of their respective azimuth reference function

$$I = \sum_{n=1}^{N} g_n(\tilde{t_n} - \Delta t_n) h_a^*(\tilde{t_n}). \tag{2.7}$$

Where $\Delta t_n = \tilde{t_n} - t_n$ is the time difference between the expected range to a cell center and the actual range to a scatterer. Assuming the exact position of the emitter and receiver antennas is known relative to the imaged plane and a single perfect point scatterer is located exactly at a cell center, then Δt_n would equal zero and there will be no residual phase for each of the N contributions for a pixel. However, this scenario is unlikely and there will be a residual phase with $e^{i2\pi f_0 \Delta t_n}$ for each pulse. If the scatteres are lying in a ideal flat plane or with exactly known location in a digital elevation map (DEM), this residual phase error can be minimized by making the cells in the image plane smaller. However

2.1.2 Parallelization of BPA

The BPA described in the previous section has a computational complexity of $\mathcal{O}(m^3)$ for an image with $m \times m$ pixels. Hence, the time to process an image can quickly escalate when going to a finer resolution. Luckily, the BPA can be parallelized on a processor or GPU, since the pixel values in an image does not depend on each other and requires similar operations for calculation. The first step in formatting a BPA image is to discretize an area in the image plane into equally sized scattering cells. The center coordinates of all the cells can be arranged in three matrices $\bar{\bar{\mathbf{P}}}^x$, $\bar{\bar{\mathbf{P}}}^y$, $\bar{\bar{\mathbf{P}}}^z$, containing the x, y and z values respectively. Assuming $m \times m$ scattering cells, $\bar{\bar{\mathbf{P}}}^x$ is given by

$$\bar{\bar{\mathbf{P}}}^x = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mm} \end{bmatrix}. \tag{2.8}$$

The $\bar{\bar{\mathbf{P}}}^y$ and $\bar{\bar{\mathbf{P}}}^z$ matrices are arranged similarly with the y and z values. Furthermore, the N different positions for the emitter and detector can also arranged in matrices $\bar{\bar{\mathbf{E}}}$ and $\bar{\bar{\mathbf{D}}}$ by

$$\bar{\bar{\mathbf{E}}} = \begin{bmatrix} x_1^e & y_1^e & z_1^e \\ x_2^e & y_2^e & z_2^e \\ \vdots & \vdots & \vdots \\ x_N^e & y_N^e & z_N^e \end{bmatrix}, \bar{\bar{\mathbf{D}}} = \begin{bmatrix} x_1^d & y_1^d & z_1^d \\ x_2^d & y_2^d & z_2^d \\ \vdots & \vdots & \vdots \\ x_N^d & y_N^d & z_N^d \end{bmatrix}. \tag{2.9}$$

Now a matrix $\bar{\bar{\mathbf{T}}}^n$ containing the distances from the emitter to the scattering cells and back to the detector can be constructed, where each input is found by

$$T_{ij}^n = \sqrt{(T_{ij}^{x,n})^2 + (T_{ij}^{y,n})^2 + (T_{ij}^{z,n})^2}, \tag{2.10}$$

where

$$T_{ij}^{x,n} = |P_{ij}^x - x_e^n| + |x_d^n - P_{ij}^x|, \tag{2.11}$$
$$T_{ij}^{y,n} = |P_{ij}^y - y_e^n| + |y_d^n - P_{ij}^y|, \tag{2.12}$$
$$T_{ij}^{z,n} = |P_{ij}^z - z_e^n| + |z_d^n - P_{ij}^z|. \tag{2.13}$$

The expected traveling distances for all the pulses to the different scattering cells are found, the values from the range compressed signals can be extracted. The range compressed signals are arranged in a matrix $\bar{\bar{\mathbf{g}}} = [\bar{g}_1 \; \bar{g}_2 \; \dots \; \bar{g}_N]$, where $\bar{g}_n$ is a vector of the an n'th range compressed pulse. The indexes to be extracted from an n'th pulse is then found by

$$\bar{\bar{\mathbf{J}}}^n = \frac{\bar{\bar{\mathbf{T}}}^{\mathbf{n}}}{dt \cdot c} \tag{2.14}$$

where dt is the discrete time steps in the signals. In our case the values in $\bar{\bar{\mathbf{J}}}^n$ are then rounded to the nearest integer and extracted values from an n'th pulse are stored in a matrix $\bar{\bar{\mathbf{G}}}^n$ with indices

$$G_{ij}^n = \bar{g}_n[J_{ij}^n] \tag{2.15}$$

where $\bar{g}_n[J_{ij}^n]$ is the complex value of $\bar{g}_n$ corresponding to an expected distance to scattering cell center at (i, j). The wave number is defined as $k = \frac{2\pi f_0}{c}$, then a matrix containing values for the azimuth reference function for a single pulse to all the scattering cells, $\bar{\bar{\mathbf{H}}}^n$, can be formed with inputs of

$$H_{ij}^n = e^{ikT_{ij}^n}. \tag{2.16}$$

The contribution from an n'th single pulse to $m \times m$ pixels is then given by

$$\bar{\bar{\mathbf{I}}}^n = \bar{\bar{\mathbf{G}}}^n \odot \bar{\bar{\mathbf{H}}}^n, \tag{2.17}$$

where $\odot$ is element wise multiplication. In Python[1] using the CuPy package[47] all the matrix operations can be performed in parallel on the GPU to speed up the image generation. Furthermore, the matrices $\bar{\bar{\mathbf{T}}}$, $\bar{\bar{\mathbf{J}}}$, $\bar{\bar{\mathbf{G}}}$ and $\bar{\bar{\mathbf{H}}}$ can be arranged in three dimensional matrices along the n, i and j dimension. Then the final matrix $\bar{\bar{\mathbf{I}}}$ can be summed up along all the pulses in the n dimension to obtain the final matrix with complex values for all the $m \times m$ pixels.

2.1.3 Autofocus

Ideally, for a perfectly matched azimuth filter, all the N contributions from a scatterer in a cell should have a similar phase such that the amplitude of the pixel value will be high. However, phase errors can occur due to various factors which will degrade the pixel value and drastically degrade the quality of the resulting image. Sources of these errors can be radar system phase errors, errors in measurement of position, or errors in the digital elevation map of the imaged surface. Errors in the radar system could be drift of a local oscillator or errors in the time of flight measurements of the pulses.

Multiple autofocus methods exist to limit these errors and obtain a better SAR image quality, however, in this project a method for autofocus in BPA proposed by M. Duersch [14] was applied.

The first step in this method is to choose a set of M pixels which have been calculated from N different pulses. The main error in the experimentally measured data in this project is the position of antennas in a bistatic setup with respect to a rotation center of a scene. Hence, we want to optimize these positions to obtain better quality images. To perform an iterative process to optimize the positions, an image cost function is applied as a measure to how well focused an image is. The cost function used here is an entropy function, F, given by Xi, Guosui and Ni [48] as

$$F = -\sum_{m=1}^{M} \bar{I}_m \ln\left(\bar{I}_m\right), \tag{2.18}$$

where $\bar{I}_m$ is the normalized pixel values defined as

$$\bar{I}_m = \frac{|I_m|^2}{\sum_{m=1}^{M} |I_m|^2}. \tag{2.19}$$

Here, $|I_m|$ denotes the absolute value of the complex number I_m. The cost function F can then be minimized changing the positions in the BPA. The parameter A is constrained by $A_{min} \leq A < A_{max}$ and it can now be varied to minimize F iteratively. Multiple parameters such as bistatic angle, slant angle, distance to emitter and detector can be minimized together. The proposed method of this minimization by Duersch [14] is a coordinate descent method, where a k'th iteration for a parameter A_n, noted as y, is minimized while keeping the others constant.

$$A_n^{k+1} = \text{arg min } F(A_1^{k+1}, ..., A_{n-1}^{k+1}, y, A_{n+1}^{k}, ..., A_N^{k}). \tag{2.20}$$

Here A_n^{k+1} becomes equal to the value of y which gives the minimal value of F. This will fullfil the condition

$$F(\mathbf{A^0}) \geq F(\mathbf{A^1}) \geq F(\mathbf{A^2}) \geq ... , \tag{2.21}$$

where $\mathbf{A}$ is the set of parameters to be optimized. In this manner the entropy function can be iteratively minimized.

2.2 SAR Interferometry

As of now only amplitude have been considered when formatting SAR images, from this we have been able to calculate reflectivity of a scene. However, this leaves out a significant amount of information as the phase is ignored. The phase of each pixel is in itself not widely usable, but when compared with one or more high resolution SAR images of a scene, it is possible to gain precise measurements of topology and deformation. This is known as interferometric synthetic aperture radar (InSAR) and is widely used in the fields of geodesy and remote sensing. These fields use spaceborne and airborne InSAR systems which can, depending on the system, produce images of large scenes spanning multiple kilometers with down to centimeter resolution. These images are then used to generate digital surface models (DSMs). [11]

DSMs are a subcategory of the digital terrain model (DTM). A digital terrain model describes the elevation of some surface. It includes the elevation information often with interpolation in between points and relevant metadata such as estimated error, real-world position of the cell etc. The DSM is a simple array of surface elevation values projected onto a coordinate system of the measured scene, functioning similarly to the digital elevation model (DEM), where the difference in the models is the considered height. The DEM provides the elevation, bare earth height, ignoring surface structure such as trees etc. while the DSM considers the elevation including surface elements. [11]

This section is mainly based on the works of Duersch [14] *Backprojection for Synthetic Aperture Radar*. In this section common concepts regarding backprojection InSAR is introduced along with fundamental equations derived from geometric considerations with emphasis on DSMs.

2.2.1 Estimation of Elevation from Phase

The received coherent radar signal contains a phase element. This element indicates the distance traveled. The phase is wrapped to 2π giving an expression of

$$\phi = 2kr \mod 2\pi, \tag{2.22}$$

where k is the wavenumber of the radar signal and r is the range to a given point. Applying this to a simple two receiver system with antennas a and b, it is possible to determine the difference in path length between the two radars with high precision. The residual phase of a backprojected pixel is directly tied to the residual range of the pixel Δr. Thus, the backprojected interferometric phase difference Ψ between two backprojected pixels (registered to the same location) can be expressed as

$$\Psi \equiv \Delta_{a,b}(\Delta\phi_i) = k(\Delta r_a - \Delta r_b). \tag{2.23}$$

A simple interferometric setup is stereo imaging, see figure 2.1. The scene is imaged from two different sensor platform positions, denoted by a and b, displaced by a baseline B, which is orthogonal to the azimuthal direction x, and is at an angle β to the ground plane. The average height of the sensor platform is z=H at position $y = 0$.

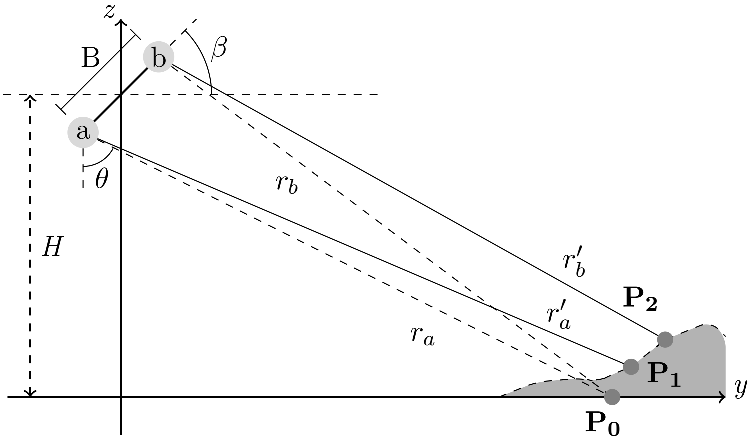


Figure 2.1. *Geometry of stereo imaging applied for interferometry.*

SAR images are 2D images of a chosen plane, usually the ground plane, formatted by certain algorithms such as the backprojection algorithm (BPA). The algorithm produces the image in an equally spaced grid of points (x, y) with amplitudes representing the complex reflectivity of the scene. From the data provided by the SAR images, the position of a given pixel (x_0, y_0) and its complex reflectivity $R = A \exp i\phi$ should be known. Additionally, the position of the radar platform and the baseline is known, giving the height H, the baseline length B and the baseline tilt β.

Counterintuitively, backprojection removes the phase component of the radar signal and replaces it with a residual phase directly dependent on the range error. Many focusing methods try to minimize or remove this residual. However, for interferometric backprojection SAR, this phase can and will be utilized to gain the height information, as the different platform positions will observe different scattering cells and have different residual phases. Figure 2.1 shows these different points in an exaggerated manner. $\mathbf{P}_0$ is the estimated center of the scattering cell, while $\mathbf{P}_1$ and $\mathbf{P}_2$ are the actual centers of the scattering cell measured by platform a and b, respectively. The different positions of the observed centers of the radar platforms can be expressed as displacements from the estimated center, i.e.

$$\mathbf{P}_1 = \mathbf{P}_0 - \boldsymbol{\delta}, \tag{2.24}$$

$$\mathbf{P}_2 = \mathbf{P}_0 - (\boldsymbol{\delta} + \boldsymbol{\epsilon}). \tag{2.25}$$

Here, $\boldsymbol{\epsilon}$ is the displacement between $\mathbf{P}_1$ and $\mathbf{P}_2$, and $\boldsymbol{\delta}$ the displacement of $\mathbf{P}_1$ to $\mathbf{P}_0$, which is defined as the interferometric displacement, and will be the variable that determines the height in the scene. The range residuals are defined as

$$\Delta r_a = r_a - r'_a, \tag{2.26}$$

$$\Delta r_b = r_b - r'_b. \tag{2.27}$$

These distances can be expressed as the distance between various points in the scene.

$$r_a = |\mathbf{P}_a - \mathbf{P}_0|, \tag{2.28}$$

$$r'_a = |\mathbf{P}_a - \mathbf{P}_1|, \tag{2.29}$$

$$r_b = |\mathbf{P}_b - \mathbf{P}_0|, \tag{2.30}$$

$$r'_b = |\mathbf{P}_b - \mathbf{P}_2|. \tag{2.31}$$

Here, $\mathbf{P}_a$ and $\mathbf{P}_b$ denotes the position of emitter a and b, respectively. Applying equation 2.24 and 2.25 to the above equations and splitting points and vectors into their respective x, y and z components, the residuals can be expressed as [14]

$$\Delta r_a = \frac{\delta_x(x_a - x_0) + \delta_y(y_a - y_0) + \delta_z(z_a - z_0)}{r_a}, \tag{2.32}$$

$$\Delta r_b = \frac{(\delta_x + \epsilon_x)(x_b - x_0) + (\delta_y + \epsilon_y)(y_b - y_0) + (\delta_z + \epsilon_z)(z_b - z_0)}{r_b}. \tag{2.33}$$

The above expressions, when inserted into equation 2.23, gives an equation with six unknowns. Therefore, it is necessary to apply some simplifying assumptions to reduce the amount of terms. First, we assume that $B << r$. This indirectly assumes a high correlation of pixels, which essentially sets $\epsilon \approx 0$, eliminating 3 terms. Next, we assume that the lateral displacement is 0, thus we only see a displacement in the height. Duersch [14] discusses the consequences thereof, however, for the scope of this report it is left out. This eliminates two terms more, leaving only the displacement in height. Inserting the reduced residuals into equation 2.23 we get

$$\Psi = k\left(\frac{\delta_z(z_a - z_0)}{r_a} - \frac{\delta_z(z_b - z_0)}{r_b}\right). \tag{2.34}$$

Solving for δ_z we get

$$\delta_z = \frac{\Psi r_a r_b}{k(r_b(z_a - z_0) - r_a(z_b - z_0))}, \tag{2.35}$$

This can be rewritten by introducing the angle of incidence θ, introduced in figure 2.1, where $r_i \cos\theta_i = z_i - z_0$,

$$\delta_z = \frac{\Psi}{k(\cos\theta_a - \cos\theta_b)}. \tag{2.36}$$

This is a simple expression relating the phase to the height of a pixel.

2.2.2 Phase Unwrapping

As mentioned previously, the radar signal is wrapped to 2π, which means, to get a continuous height function, the phase needs to be unwrapped. This posts a problem following the equation

$$\phi_{i,j} = \phi'_{i,j} + k_{i,j} \cdot 2\pi, \tag{2.37}$$

where ϕ' is the wrapped phase, ϕ is the unwrapped phase, and k is an integer. As k is unknown, the true height is difficult to obtain. However, this can be done by analyzing the change of phase between two points. If the phase moves across a fringe and changes from $-\pi$ to π or vice-versa, it can be recorded and the k-value changed accordingly. However, this is not a simple task. While other 2D signal processing methods, such as the fast Fourier transform, can be decomposed into in 1D operations on rows or columns, phase unwrapping cannot. Thus, phase unwrapping remains an active field of research, as to get reliable results in a timely manner. In general, most of the phase unwrapping methods can be classified as either path-following methods, minimum norm methods [49] or network flow methods [50]. The method used in this project is the path-following method presented by Herráez et al. [51].

Path-Following Method

The path-following method is simple in nature. Most cases follows a *flood-fill* algorithm, where a starting pixel is chosen and the phase of that pixel is stored as unwrapped in a *solution array.* The four nearest neighbors of the pixel are then unwrapped and added to a list which records the indexes of adjoining pixels to the unwrapped pixels, the *adjoin list.* This process is illustrated in figure 2.2. The unwrapping follows the procedure of

$$\phi_n^u = \begin{cases} \phi_n^w - 2\pi, & \text{for } \phi_p - \phi_n^w > \pi, \\ \phi_n^w + 2\pi, & \text{for } \phi_p - \phi_n^w \leq -\pi, \\ \phi_n^w, & \text{otherwise.} \end{cases} \tag{2.38}$$

Here, the superscript u and w denotes the unwrapped and wrapped phases, respectively. Subscripts, p and n, denotes the pixel and neighbor, respectively. These unwrapped neighbors are then stored in the *solution array.* The *adjoin list* is then updated with the new adjoining pixels and the process of unwrapping is repeated.

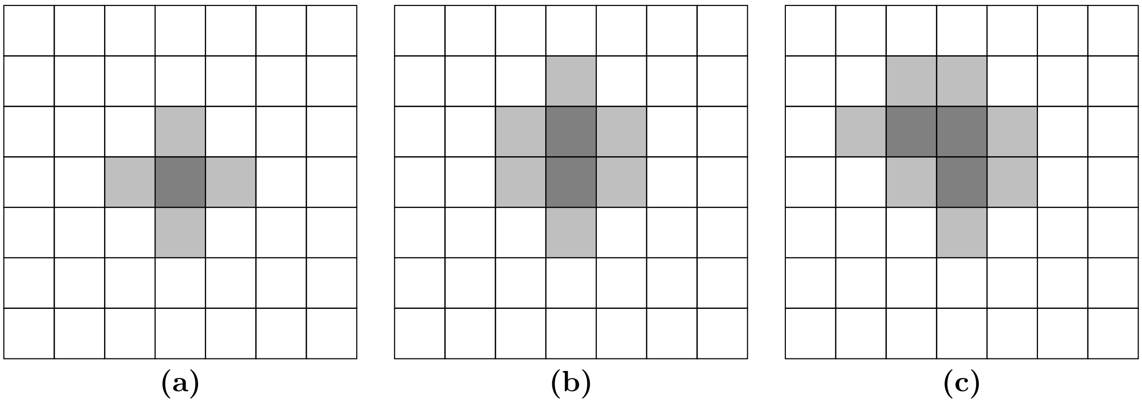


Figure 2.2. *Illustration of the flood-fill algorithm process.* ***(a)*** *The starting pixel is dark gray while adjoining pixels are light gray.* ***(b)*** *A new pixel added to the solution array, adding new pixels to adjoin list.* ***(c)*** *The process is repeated for a new pixel in the adjoin list.*

The main difference between path-following methods is the initial conditions dictating the flood-fill algorithm. Two methods will breifly be introduced. [49]

Branch-cut algorithms preprocess the interferogram, identifying residues (i.e. residual errors) in the array, and connect them through branch cuts. This restricts the flood-fill algorithm from crossing these cuts, thus reducing errors in unwrapping. [49]

Quality guided unwrapping method uses a quality map to guide the unwrapping process. This means that when the next pixel is to be unwrapped, the *adjoin list* sorted for the best quality pixel, which then will be the path of unwrapping. Thus, the flood-fill algorithm is governed by the quality of the individual pixels. One might think this method would be unreliable when trying to avoid residues and phase errors, but is proven to be efficient and reliable, and is the method of choice due to its efficiency and simplicity. The quality map guiding this flood-fill algorithm can vary, many use the phase derivative or second derivative in some combination of the coherence of the interferogram. The coherence is an important quantity in InSAR, it describes the cross-correlation between the used SAR images for interferogram generation. The coherence map generates a normalized description of the correlation between points. The coherence can be calculated by [52]

$$C = \frac{\sum_{i=1}^{L} S_a(i) S_b^*(i)}{\sqrt{\sum_{i=1}^{L} |I_a(i)|^2 \sum_{i=1}^{L} |I_b(i)|^2}}. \tag{2.39}$$

Here, L is the number of looks, S_a and S_b is image a and b, respectively, and i is the index of a given pixel. [49]

2.3 SAR Polarimetry

This section on SAR polarimetry is primarily based on the works of Hajnsek and Desnos [53] in *Polarimetric Synthetic Aperture Radar.*

SAR polarimetry is a multidimensional data technique, whereby the polarization parameters of the radar radiation is used to gain additional target information. Based on target material and geometry, the target can behave very differently under different polarizations. These differences are used in characterization and classification. All polarimetry stems from polarization of electromagnetic radiation, for which the E-field may be expressed by two orthogonal directions $\hat{\mathbf{h}}$ and $\hat{\mathbf{v}}$

$$\mathbf{E} = E_h \hat{\mathbf{h}} + E_v \hat{\mathbf{v}}, \tag{2.40}$$

where E_h and E_v are complex amplitudes. The two polarization vectors $\hat{\mathbf{h}}$ and $\hat{\mathbf{v}}$ are referred to as horizontal (H) and vertical (V). When a transmitted electromagnetic field $\mathbf{E}^t$ hits a target and scatters, the scattered electric field $\mathbf{E}^s$ may be written as

$$\mathbf{E}^s = \begin{bmatrix} S_{hh} & S_{hv} \\ S_{vh} & S_{vv} \end{bmatrix} \begin{bmatrix} E_h^t \\ E_v^t \end{bmatrix} = \bar{\bar{\mathbf{S}}} \mathbf{E}^t, \tag{2.41}$$

where $\bar{\bar{\mathbf{S}}}$ is the so called scattering matrix. The goal of polarimetry can then be expressed as mapping every pixel of a SAR image with a 2×2 scattering matrix. In ordinary SAR each pixel is mapped with a single complex number, but with polarimetry each pixel has four complex numbers, one for each matrix element. To probe each of these matrix elements a quad-pol radar setup must be able to coherently transmit and receive both H and V polarization. By setting one component of the transmitted electric field equal to zero, it becomes possible to measure two components of the scattering matrix simultaneously. Mathematically this operation looks like

$$\begin{bmatrix} S_{hh} \\ S_{hv} \end{bmatrix} = \begin{bmatrix} S_{hh} & S_{hv} \\ S_{vh} & S_{vv} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{2.42}$$

for emitting H polarized radiation. S_{hh} and S_{hv} can then be detected by two orthogonally polarized detectors.[12]

The scattering matrices for simple geometries are known, however for more complex scenes, such as vegetation, snow and ice, knowing the exact scattered complex E-field is practically impossible. The resulting scattering matrix for each pixel is instead thought of in a statistical form. Each pixel or scatterer, typically referred to as a partial scatterer, is then though of as containing several uncorrelated descriptive scatterers. To accurately describe partial scatterer it is necessary to use the so called coherence matrix $\bar{\bar{\mathbf{C}}}$. The coherence matrix is fully defined by the scattering matrix in the following way. First construct the so called target vector by vectorizing the 2×2 scattering matrix. In this project we use the lexicographic basis, where the target vector is given by

$$\mathbf{k}_l = \begin{bmatrix} S_{hh} \\ S_{hv} \\ S_{vh} \\ S_{vv} \end{bmatrix}. \tag{2.43}$$

We can then create a preliminary matrix $\bar{\bar{\mathbf{K}}}$ by taking the outer product of the target vector with its transpose conjugate

$$\bar{\bar{\mathbf{K}}} = \mathbf{k}_l \mathbf{k}_l^{T*} = \begin{bmatrix} |S_{hh}|^2 & S_{hh}S_{hv}^* & S_{hh}S_{vh}^* & S_{hh}S_{vv}^* \\ S_{hv}S_{hh}^* & |S_{hv}|^2 & S_{hv}S_{vh}^* & S_{hv}S_{vv}^* \\ S_{vh}S_{hh}^* & S_{vh}S_{hv}^* & |S_{vh}|^2 & S_{vh}S_{vv}^* \\ S_{vv}S_{hh}^* & S_{vv}S_{hv}^* & S_{vv}S_{vh}^* & |S_{vv}|^2 \end{bmatrix}, \tag{2.44}$$

where ^{T} is the matrix transpose and * is the complex conjugate. To get the coherence matrix the we must take the expectation operator $\langle \cdot \rangle$ of the matrix $\bar{\bar{\mathbf{K}}}$, giving the finial definition of the coherence matrix as,

$$\bar{\bar{\mathbf{C}}} = \langle \bar{\bar{\mathbf{K}}} \rangle = \begin{bmatrix} \langle |S_{hh}|^2 \rangle & \langle S_{hh}S_{hv}^* \rangle & \langle S_{hh}S_{vh}^* \rangle & \langle S_{hh}S_{vv}^* \rangle \\ \langle S_{hv}S_{hh}^* \rangle & \langle |S_{hv}|^2 \rangle & \langle S_{hv}S_{vh}^* & \langle S_{hv}S_{vv}^* \rangle \\ \langle S_{vh}S_{hh}^* \rangle & \langle S_{vh}S_{hv}^* \rangle & \langle |S_{vh}|^2 \rangle & \langle S_{vh}S_{vv}^* \rangle \\ \langle S_{vv}S_{hh}^* \rangle & \langle S_{vv}S_{hv}^* \rangle & \langle S_{vv}S_{vh}^* \rangle & \langle |S_{vv}|^2 \rangle \end{bmatrix}. \tag{2.45}$$

The maximum likelihood estimator of the expectation value for the coherence matrix can be shown to correspond to the spacial average, referred to as multilook. With multilook, neighboring pixels in range and azimuth are averaged, at the expense of spatial resolution. The number of independent pixels averaged, is referred to as the number of looks N. Doing so also serves to reduce speckle in the SAR image. It can be shown that the standard deviation of pixel signals S_N is inverse proportional to the square root of the number of looks N [12]

$$S_N \propto \frac{1}{\sqrt{N}}. \tag{2.46}$$

The coherence matrix given by equation 2.45 is the most general expression. However in polarimetry it is common to express the target vector in the backscatter alignment (BSA) coordinate system, see figure 2.3.

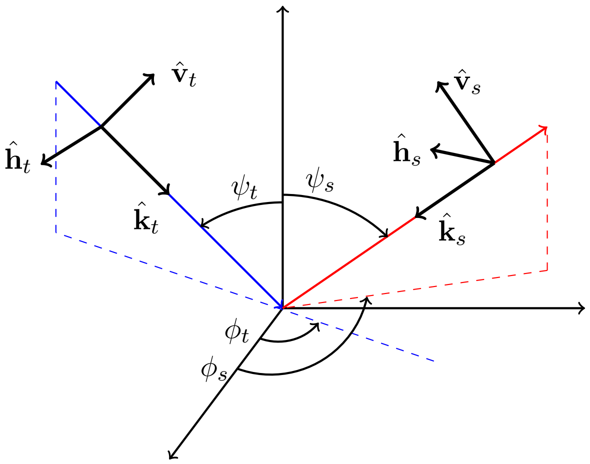


Figure 2.3. *Schematic of the BSA coordinate system.*

The direction of the transmitted and scattered field is given by the angle pair (ϕ_n, ψ_n) respectively. Note how the scattered field is given in a basis with $\hat{\mathbf{k}}_s$ pointing backwards, which means that in BSA the transmitted and receiving polarization coincide for monostatic setups. As most polarimetry systems transmit and receive from the same radar, the framework of monostatic BSA is often assumed. An important property of this framework is reciprocity, which states

$$S_{hv} = S_{vh}. \tag{2.47}$$

With reciprocity the scattering matrix can be written as

$$\bar{\bar{\mathbf{S}}} = \begin{bmatrix} S_{hh} & S_{hv} \\ S_{hv} & S_{vv} \end{bmatrix}. \tag{2.48}$$

The total power of the polarimetry radar setup is known as the span, and in the general case it is calculated by

$$\text{Span}(\bar{\bar{\mathbf{S}}}) = \text{Trace}(\mathbf{k}_l \mathbf{k}_l^{T*}) = |S_{hh}|^2 + |S_{hv}|^2 + |S_{vh}|^2 + |S_{vv}|^2. \tag{2.49}$$

And for the BSA case it can be calculated by

$$\text{Span}(\bar{\bar{\mathbf{S}}}) = |S_{hh}|^2 + 2|S_{hv}|^2 + |S_{vv}|^2. \tag{2.50}$$

The span is what is referred to as polarimetrically invariant, which means it dose not depend on the choice of polarization basis or coordinate system. Considering the definition of span, it is possible to express the target vectors in the BSA framework as

$$\mathbf{k}_l = \begin{bmatrix} S_{hh} \\ \sqrt{2} S_{hv} \\ S_{vv} \end{bmatrix}. \tag{2.51}$$

It is then also possible to construct a 3×3 coherence matrix in the BSA framework, based on the new target vector, which is given by

$$\bar{\bar{\mathbf{C}}} = \begin{bmatrix} \langle |S_{hh}|^2 \rangle & \langle \sqrt{2} S_{hh} S_{hv}^* \rangle & \langle S_{hh} S_{vv}^* \rangle \\ \langle \sqrt{2} S_{hv} S_{hh}^* \rangle & \langle 2 |S_{hv}|^2 \rangle & \langle \sqrt{2} S_{hv} S_{vv}^* \rangle \\ \langle S_{vv} S_{hh}^* \rangle & \langle \sqrt{2} S_{vv} S_{hv}^* \rangle & \langle |S_{vv}|^2 \rangle \end{bmatrix}. \tag{2.52}$$

This definition of the coherence matrix is the most general case for monostatic setups in the BSA framework.

2.4 Ray Tracing

A ray tracer is a program that approximates beams of light with rays, and creates pictures based on the rays that successfully navigate the scene from the source to the detector. A simple introduction can be found in *Fundamentals of computer science* by Shirley, Ashikhmin and Marschner [54], but at its core, a ray tracer tries to find the closest ray-surface intersection in the scene and calculate scattering on the surface. The most popular use case is that of video graphics for computing realistic lighting. However, the same technique also applies to different parts of the electromagnetic spectrum, like for radar, as shown by Schüßler et al. [55]. The biggest difference occurs in the so called shading step, where the output image, or in the case of radar, output signals are computed based on rays traversing the scene.

2.4.1 Bounding Volume Hierarchy

To model large complex scenes and objects requires many thousands of surfaces, like triangles and polygons, called primitives. However doing so without further optimization is very expensive as the computational time for checking ray-primitive intersections increases linearly with the number of primitives. To overcome this limitation the scene geometry can be sorted into a binary tree, called a bounding volume hierarchy (BVH). A tree is a data structure consisting of connected nodes in a hierarchical order. Each node is connected to some amount of child nodes and exactly one parent node, except for the root node which has no parents. A binary tree refers to a tree structure where the amount of child nodes is limited to 2. A bounding volume is a virtual shape, i.e. added by the ray tracer but not part of the actual scene, to check for any possible ray-primitive intersections of the primitives located inside. The BVH then recursively splits up the primary bounding volume (the root node) into smaller and smaller bounding volumes, each with one parent volume and two child volumes, with the leafs pointing to the enclosed primitives. A simple 2D sketch of a BVH can be seen in figure 2.4. Note how a ray not intersecting C, makes it possible to exclude both the red and blue triangle for possible ray-primitive intersections. A BVH can effectively eliminate the need to calculate ray-primitive intersections far from the path of the ray, and thus on average reduce the time complexity from linear to logarithmic.

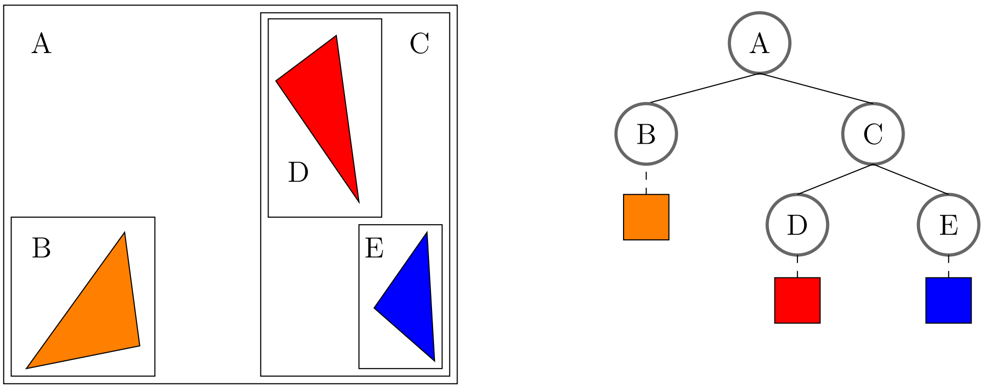


Figure 2.4. *Schematic of a BVH in 2D and a possible binary tree construction.*

In practice there are many different algorithms to construct BVH's. When choosing an algorithm it is often a choice between construction speed and BVH quality. Here quality refers to the ordering of primitives and bonding volumes for the most effective tree traversal. The computational cost associated with a subtree of the BVH with root node N, can be formulated with the recurrence equation:

$$c(N) = \begin{cases} c_T + \sum_{N_c} P(N_c|N)c(N_c) & \text{if N is interior node,} \\ c_I|N| & \text{otherwise,} \end{cases} \tag{2.53}$$

where $c(N)$ is the cost of a subtree with root N, N_c is a child of the root node N, and $P(N_c|N)$ is the probability of traversing a child node N_c when the root N is hit. c_T and c_I are the average cost of a traversing step and a ray-primitive intersection, and $|N|$ is the number of primitives with root N. In practices c_T and c_I are software and hardware specific. $c(N)$ can be used to determine if a bounding volume should be split or left whole. In practice $P(N_c|N)$ is hard to know, but under the assumption of uniformly distributed ray origins and directions, surface area heuristic (SAH) can be used to express the probability by

$$P(N_c|N)^{SAH} = \frac{S(N_c)}{S(N)}, \tag{2.54}$$

where $S(N)$ is the surface area of the bounding box of node N. Other heuristics based on direction distributions and occlusion of scene objects also exist. The main advantage of SAH is the ease of calculating the probability. Many BVHs use so called axis aligned bounding boxes (aabb), with easy to calculate surface area.[56]

Building and traversing the BVH

This section outlines the algorithm used to build and traverse BVHs in this report, and is based on the work by Shirley [57] in *Ray Tracing in a Weekend*. The algorithm is completely general, but here it is applied to triangular meshes.

The type of bounding volume used will be aabbs. As the name implies, they are boxes where all edges are parallel with the x-, y- and z-axis of the coordinate system. The definition of aabbs allows us to uniquely define each aabb by an x-, y- and z-interval. When a ray intersects an aabb, the results will be two points of intersection defined by the intersection parameters t_0 and t_1, that define each intersection point in 3D by

$$\mathbf{p}(t) = \mathbf{O} + t\mathbf{v}, \tag{2.55}$$

where $\mathbf{O}$ is the ray origin, $\mathbf{v}$ is the ray direction and $\mathbf{p}(t)$ the ray end points. In 3D the x-, y- and z-intervals define the location of boundary planes, which for the yz-planes are defined by $x = x_0$ and $x = x_1$ and so on. However, to calculate the intersection parameters for a ray with the x-interval, that is only a 1D problem. It amounts to solving the x-components of equation 2.55 for when $p_x(t) = x_0$ and $p_x(t) = x_1$. The final equations for the x-interval intersection parameters t_{0x} and t_{1x} are thus given by

$$t_{0x} = (x_0 - O_x)/\, v_x, \tag{2.56}$$

$$t_{1x} = (x_1 - O_x)/\, v_x. \tag{2.57}$$

The same is then done for the y- and z-intervals as well, resulting in y- and z-interval intersection parameters t_{0y}, t_{1y}, t_{0z} and t_{1z}. For a ray to hit the box, the t-intervals along x, y and z must overlap. The ray-aabb hit test then simply amounts to solving 6 one dimensional equation for the intersection parameters and computing the overlap of the three resulting intervals, which is very efficient! This is very important as the main bottleneck of traversing a BVH is ray-aabb hit calculations. The full implementation of the ray-aabb hit test is based on the work of Williams et al. [58], in which edge cases such as $v_x = 0$ are dealt with. Furthermore simple techniques for performance enhancement are also presented.

A top-down approach was used to construct the BVH, by recursively splitting the set of triangles into two subsets (the children). For each new subsets, an aabb is constructed to contain the set completely. This step is then repeated until the desired distribution of triangles and bounding volumes is obtained. There are many different algorithms for choosing how and where to splitting bounding volumes. The BVH will be most effective if the two child boxes have smaller volumes than their parent box, and minimum overlap.

The splitting algorithm used in this project can be summarized as:

1. Identify the longest axis of the aabb
2. Sort all triangles by their centroid along said axis
3. Put half in each child box

This algorithm secures even splitting in the BHV tree, and an even distribution of primitives in the leaf nodes. This algorithm works best on meshes with triangles of roughly equal size. The implemented data structure for the BVH is outlined in A.

To find the nearest ray-primitive intersection we traverse the BVH in a top-down manner. Firstly the ray-aabb intersection of the root node of the BVH is computed. If there is no intersection we know the ray to have missed, however if there is an intersection we check for ray-aabb intersections with it's child nodes. Any child node that is hit it will be pushed onto the stack. The stack is a list of possible bounding volumes containing the nearest ray-primitive intersection.

We then repeat the process of calculating ray-aabb intersections with the nodes on the stack, while pushing child nodes we hit with the ray (t_c) onto the stack as we go. When picking nodes of the stack we prioritize nodes deeper in the tree, until we hit a leaf node containing actual primitives. When a leaf node is hit we calculate ray-primitive intersections with all primitives in said leaf, giving a ray parameter t_p, the closest tested ray-primitive intersection. Then compare all child nodes on the stack with t_p and remove all child nodes with $t > t_p$. The pseudocode of the process tree traversal can be seen in algorithm 1

Algorithm 1 BVH traversal

```
Let $t_p$ be a large number
Push *root* onto the stack
**while** stack is not empty **do**
    pop deepest *node* from stack
    **if** ray intersects *node* **then**
        **if** *node* is not leaf **then**
            **for** each child in node **do**
                get $t_c$
                **if** $t_c < t_p$ **then**
                    push child *node* onto stack
                **end if**
            **end for**
        **else**
            **for** each primitive in leaf **do**
                get $\mathbf{t_l}$
            **end for**
            get $t_l^{min} = min(\mathbf{t_l})$
            **if** $t_l^{min} < t_p$ **then**
                $t_p = t_l^{min}$
                **for** each *node* on stack **do**
                    **if** $t_c > t_p$ **then**
                        remove *node* from stack
                    **end if**
                **end for**
            **end if**
        **end if**
    **end if**
**end while**
```

Here $\mathbf{t_l}$ is the list of all ray-primitive intersections in said leaf, and t_l^{min} is the minimum value of $\mathbf{t_l}$.

Scattering Profile

The main scattering model used in the ray-tracer is a linear weighting of specular and diffuse scattering. The parameter α controls the degree of diffuse and specular scattering of the scattered ray. Let $\hat{\mathbf{v}}$ be the ray directions and $\hat{\mathbf{n}}$ be the surface normal of the scattering surface, then the specular direction is given by

$$\hat{\mathbf{s}} = \hat{\mathbf{v}} - 2(\hat{\mathbf{v}} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}. \tag{2.58}$$

A random Lambertian diffuse direction can be calculated by

$$\mathbf{d} = \hat{\mathbf{n}} + \hat{\mathbf{o}}, \tag{2.59}$$

where $\hat{\mathbf{o}}$ is a random point on the unit sphere. The final scattering direction is then given by [55] as

$$\mathbf{v}_\alpha = \mathbf{v}_\alpha(\alpha, \hat{\mathbf{d}}, \hat{\mathbf{s}}) = \alpha\hat{\mathbf{d}} + (1-\alpha)\hat{\mathbf{s}}. \tag{2.60}$$

The $\hat{\mathbf{v}}_\alpha$ is characterized by limited backscattering from most α values. To get even a small amount of back scattering the specular scattering direction must be blurred heavily by weighting $\hat{\mathbf{d}}$ highly. In addition to the α-model two new scattering profiles were also tested. The first model, the γ-model, exhibits backscattering by sending rays back along the $-\hat{\mathbf{v}}$ direction. Let γ determine the probability function $p_\gamma(x)$. For a random x-value in the interval $[0,1]$, $p_\gamma(x)$ is given by,

$$p_\gamma(x) = \begin{cases} 1, & \text{if } 0 \leq x < \gamma \\ 0, & \text{if } \gamma \leq x \leq 1 \end{cases}. \tag{2.61}$$

The final scattering direction of the γ-model can then be expressed as,

$$\mathbf{v}_\gamma = \hat{\mathbf{v}}_\alpha(\alpha_1, \hat{\mathbf{d}}, \hat{\mathbf{s}})p_\gamma(x) + \hat{\mathbf{v}}_\alpha(\alpha_2, \hat{\mathbf{d}}, -\hat{\mathbf{v}})(1 - p_\gamma(x)). \tag{2.62}$$

Here α_1 and α_2 can be different for the scattered and backscattered cases, and γ controls the probability of a ray being backscattered. The second model, the Λ-model, is very similar to the γ-model, except the directional backscattering direction is replaced by lambertian scattering. The final scattering direction of the Λ-model is given by,

$$\mathbf{v}_\Lambda = \hat{\mathbf{v}}_\alpha(\alpha, \hat{\mathbf{d}}, \hat{\mathbf{s}})p_\Lambda(x) + \hat{\mathbf{d}}(1 - p_\Lambda(x)). \tag{2.63}$$

Examples of the different scattering profiles are shown in figure 2.5.

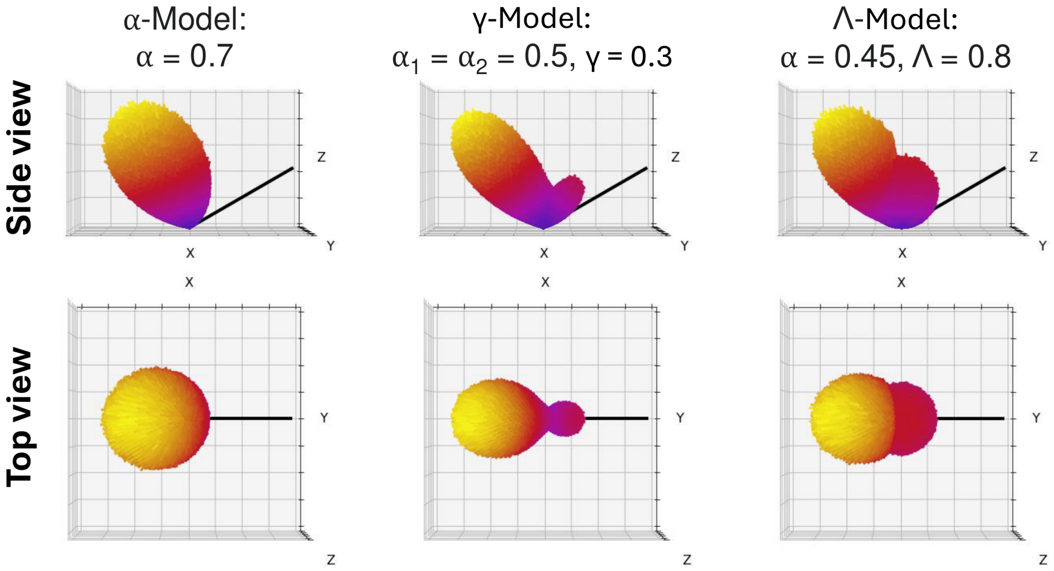


Figure 2.5. *Showcase of the distribution of different scattering models generated with 10^6 runs. The black line is the incoming ray $\hat{\mathbf{v}}$. Colors show amplitude, with yellow being the largest and blue the lowest.*

Triangle Subdivision Algorithm

A good BVH has minimal overlap and total area of all bounding boxes. But sometimes to construct a good BVH it can be necessary to split up larger scene primitives. In this report all scenes are constructed with triangular meshes. To identify and split up large triangles the following algorithm was used. First, the centroid coordinate for all triangles in the mesh is calculated by,

$$\mathbf{C} = (\mathbf{V}_1 + \mathbf{V}_2 + \mathbf{V}_3)\,/\,3, \tag{2.64}$$

where $\mathbf{C}$ is the centroid and $\mathbf{V}_n$ for $n = 1, 2, 3$ are the three vertices of the triangle. The distance between centroid and vertices is then computed by,

$$D_n = |\mathbf{C} - \mathbf{V}_n|, \tag{2.65}$$

where $|\mathbf{x}|$ denotes the length of the vector $\mathbf{x}$. For any triangle where either of D_1, D_2, D_3 is larger than the chosen max centroid-vertex distance D_{max}, the triangle is split up into three smaller triangles in the following way. First identify the vertex associated with the largest centroid-vertex distance, here defined as $\mathbf{V}_l$. The two remaining vertices are then defined as $\mathbf{V}_k$ and $\mathbf{V}_m$. Next calculate the location of two new vertices $\mathbf{V}_n$ and $\mathbf{V}_o$ by

$$\mathbf{V}_n = (\mathbf{V}_l + \mathbf{V}_k)/\,2 \tag{2.66}$$

$$\mathbf{V}_o = (\mathbf{V}_l + \mathbf{V}_m)/\,2 \tag{2.67}$$

$\mathbf{V}_n$ and $\mathbf{V}_o$ are located at the midpoint between $\mathbf{V}_l$ and respectively $\mathbf{V_k}$ and $\mathbf{V}_m$. There are now two scenarios, depending on which of the two vertices $\mathbf{V}_k$ and $\mathbf{V}_m$ is closest to $\mathbf{V}_l$. The distance from $\mathbf{V}_k$ and $\mathbf{V}_m$ to $\mathbf{V}_l$ are given by

$$L_{lk} = |\mathbf{V}_l - \mathbf{V}_k|, \qquad L_{lm} = |\mathbf{V}_l - \mathbf{V}_m|. \tag{2.68}$$

If $L_{lk} > L_{mk}$ we proceed without changes, however if $L_{lk} < L_{mk}$, switch the vertices of $\mathbf{V}_k$ and $\mathbf{V}_m$, such that $\mathbf{V}_k \rightarrow \mathbf{V_m}$ and $\mathbf{V}_m \rightarrow \mathbf{V_k}$. The three new triangles are then defined by the the vertices $T_1 = [\mathbf{V}_l, \mathbf{V}_n, \mathbf{V}_o]$, $T_2 = [\mathbf{V}_k, \mathbf{V}_m, \mathbf{V}_n]$ and $T_3 = [\mathbf{V}_m, \mathbf{V}_n \mathbf{V}_o]$. The resulting subdivision of an arbitrary triangle can be seen in figure 2.6.

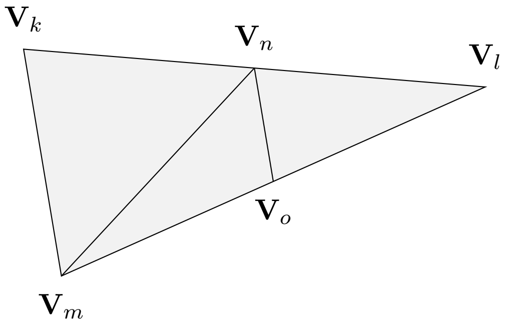


Figure 2.6. *Resulting subdivision of a single large triangle into three smaller triangles.*

For meshes with very large variation in the size of the triangles it can be necessary to carry out the subdivision algorithm on some of the newly created triangles.

2.4.2 Ray Tracing for Lens Design

This section will offer a brief overview of how a ray tracer might be modified to aid in the design process of optical lenses. If we assume the lenses exhibit circular symmetry, the ray tracer can be brought down into 2D, speeding up the calculation time. The main algorithm then involves sending out rays on the 2D lens, and calculating refraction of the individual rays. To model refraction we will implement, Snell's law which is given by

$$n_0 \sin(\theta_0) = n_1 \sin(\theta_1), \tag{2.69}$$

where n_0 and n_1 are the refractive indices of the incident and refractive mediums and θ_0 and θ_1 are the angles of incidence and refraction, respectively. For use in the ray tracer we would like to reformulate Snell's law with 2D-vector notation. See figure 2.7 for a full picture of needed vectors and angles for the vector implementation. Consider a ray with a direction vector $\hat{\mathbf{v}}_0$, with the subscript denoting the media of propagation. Let the ray be incident on some interface (line in 2D) with the normal vector $\hat{\mathbf{n}}$, that always fulfills $\hat{\mathbf{n}} \cdot \hat{\mathbf{v}}_0 < 0$. The incident angle can then be calculated by

$$\theta_0 = \cos^{-1}((-\hat{\mathbf{n}}) \cdot \hat{\mathbf{v}}_0), \tag{2.70}$$

assuming both $\hat{\mathbf{v}}_0$ and $\hat{\mathbf{n}}$ are normalized. θ_1 can then be calculated by using Snell's law. To get the direction vector of the diffracted ray, we will introduce the vector $\hat{\mathbf{a}}$ that lies in the interface, which is given by

$$\hat{\mathbf{a}} = \begin{bmatrix} \hat{n}_y & -\hat{n}_x \end{bmatrix}^T. \tag{2.71}$$

The vector $\hat{\mathbf{a}}$ is used to ensure the correct rotation direction of the normal $-\hat{\mathbf{n}}$ to create the new refracted ray $\hat{\mathbf{v}}_1$. Depending on the sign of the dot product $\hat{\mathbf{a}} \cdot \hat{\mathbf{v}}_0 = a_0$, the normal $-\hat{\mathbf{n}}$, will be rotated either clockwise or anticlockwise according to

$$\theta_1 = \begin{cases} \sin^{-1}\left(\frac{n_0}{n_1}\sin(\theta_0)\right) & \text{for } a_0 > 0 \\ -\sin^{-1}\left(\frac{n_0}{n_1}\sin(\theta_0)\right) & \text{for } a_0 < 0 \end{cases} \tag{2.72}$$

The refracted ray $\hat{\mathbf{v}}_1$ is then given by the matrix equation

$$\hat{\mathbf{v}}_1 = -\bar{\bar{\mathbf{R}}}\hat{\mathbf{n}}, \tag{2.73}$$

where $\bar{\bar{\mathbf{R}}}$ is the 2D rotation matrix given by

$$\bar{\bar{\mathbf{R}}} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) \end{bmatrix}. \tag{2.74}$$

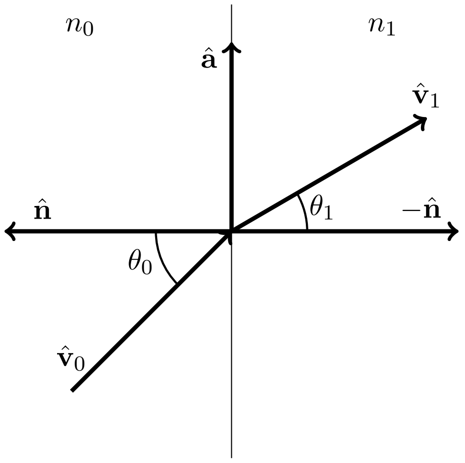


Figure 2.7. *Refraction of incoming ray $\hat{\mathbf{v}}_0$ on an interface, by Snell's law.*

Next comes the challenge of calculating ray-surface intersection of complex shapes in 2D, like the surfaces of the lens. To do so, let the complex surface be approximated by straight line segments. For the sake of simplicity, uniform discretization along the axis perpendicular to the optical path is chosen, see figure 2.8.

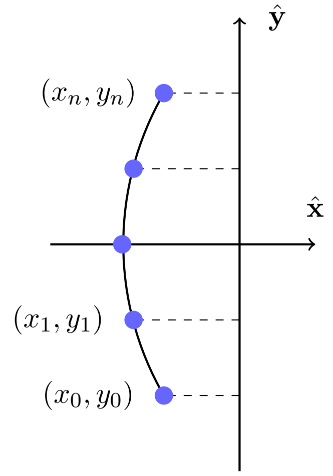


Figure 2.8. *Discretization of an arbitrary line segment of a lens.*

For Lenses aligned in the same plane, this assumption work well with enough small line segments. Then for each surface we have a vector of y coordinates, $\mathbf{y}$, and a surface-function $f(y) = x$ that calculates $\mathbf{x}$. For N points (x, y), the resulting number of line segments is $N - 1$, defined by $x_n, x_{n+1}, y_n, y_{n+1}$ for $n \in N - 1$. To implement an algorithm for finding ray-surface intersection, calculate the line parametrization of the form

$$y = a_n x + b_n, \tag{2.75}$$

for each segment, with

$$a_n = \frac{y_{n+1} - y_n}{x_{n+1} - x_n}, \tag{2.76}$$

$$b_n = y_n - a_n x_n. \tag{2.77}$$

In addition the surface normal of the segments is also needed for calculating refraction. They can be found by a 90 degree rotation of the normalized vector $\hat{\mathbf{d}}$ from (x_n, y_n) to (x_{n+1}, y_{n+1})

$$\mathbf{d} = \begin{bmatrix} x_{n+1} - x_n & y_{n+1} - y_n \end{bmatrix}^T. \tag{2.78}$$

The normal vector is then simply given by

$$\hat{\mathbf{n}} = \bar{\bar{\mathbf{R}}}_{90} \hat{\mathbf{d}} = \begin{bmatrix} -\hat{d}_y & \hat{d}_x \end{bmatrix}^T, \tag{2.79}$$

where $\hat{d}_x$ and $\hat{d}_y$ are the x and y components of $\hat{\mathbf{d}}$. Now for calculating intersection, a ray in 2D can be written as

$$\mathbf{P}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} O_x + tv_x \\ O_y + tv_y \end{bmatrix}. \tag{2.80}$$

We then calculate the intersection between the ray and the infinite line defined by the line segment through a_n and b_n by

$$y(t) = a_n x(t) + b_n. \tag{2.81}$$

By inserting the definitions of $x(t)$ and $y(t)$ from equation 2.80 and solving for the ray parameter t, we get the following expression

$$t = \frac{a_n O_x + b_n - O_y}{v_y - a_n v_x}. \tag{2.82}$$

Lastly we check if the point $\mathbf{P}(t)$ lies on the line segment. This can easily be done by checking if $x_n < x(t) < x_{n+1}$ and $y_n < y(t) < y_{n+1}$, assuming $\alpha_{n+1} \geq \alpha_n$ for $\alpha = x, y$. If not, there is no ray-intersection with the tested segment. We do this calculation for every segment and identify the closest segment.

2.4.3 Simulating Realistic Clutter

A realistic method for modeling clutter is presented by Malmgren-Hansen et al. [59]. To simulate the received radar signal from realistic clutter, start by defining an array of point scatterers in the background. They will be defined by their location (x, y, z) and complex reflectivity R^c. For each point scatter, the reflectivity is defined by picking a real and an imaginary part from the same normal distribution $\mathcal{N}(0, \sigma^2)$ with σ^2 given by

$$\sigma^2 = \frac{\sigma_0 \delta_r \delta_a}{2\cos(\theta)}, \tag{2.83}$$

where δ_r and δ_a are the range and azimuth pixel spacings in the final SAR image, θ the slant angle and σ_0 the terrain reflectivity. For different terrain and clutter types σ_0 is given by the following equation

$$\sigma_0 = P_1 + P_2 \exp(-P_3\alpha) + P_4 \cos(P_5\alpha + P_6), \tag{2.84}$$

where α is the angle of incidence and P_n for $n = 1, 2...6$ are parameters worked out in *Handbook of Radar Scattering Statistics for Terrain*[60], by fitting experimental data to the empirical equation 2.84. The P_n values are functions of terrain type, frequency, and polarization.

To obtain the radar signal from the background, calculate the two way distance d to each visible point scatter from the point of view of the source and emitter. The final signal can be calculated by a superposition of pulses by

$$s(t) = \sum_i^N p(t_i) R_i, \tag{2.85}$$

where $p(t)$ is the radar pulse, $t_i = d_i/c$ and R_i are the displacement time and reflectivity for each point scatter, with c being the speed of light.

2.4.4 Polarization Ray Tracing

Without the implementation of polarization, a ray tracer can effectively be thought of as following unpolarized light rays around the scene. Up until this point that is exactly what has been done. However, for polarization based effects, a new way of handling the rays is needed. The polarization ray tracer is based on Jones calculus, for describing polarization states. With Jones calculus each state of polarization can be described by the Jones vector

$$\mathbf{J} = \begin{bmatrix} E_x \\ E_y \end{bmatrix}, \tag{2.86}$$

where E_x and E_y are complex amplitudes of the electric field in the transverse plane, i.e. perpendicular to $\mathbf{k}$. If the wave is not propagating in the z-direction, x, y refers to a local coordinate system in that particular transverse plane. Typically linearly polarized light in the x- and y-direction are referred to as horizontal $|H\rangle$ and vertical $|V\rangle$ polarizations respectively, given by

$$|H\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad |V\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.87}$$

In Jones calculus the change in polarization states by optical components are modeled with 2×2 Jones matrices $\bar{\bar{\mathbf{M}}}$. Examples of a linear horizontal polarizer $\bar{\bar{\mathbf{M}}}_H$ and a quarter-wave retardation plate $\bar{\bar{\mathbf{M}}}_q$ are given below.

$$\bar{\bar{\mathbf{M}}}_H = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad \bar{\bar{\mathbf{M}}}_q = e^{i\pi/4} \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}. \tag{2.88}$$

The horizontal linear polarizer can be seen to eliminate the vertical components of $\mathbf{J}$, by

$$\mathbf{J}_{out} = \bar{\bar{\mathbf{M}}}_H \mathbf{J}_{in} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} E_x \\ E_y \end{bmatrix} = \begin{bmatrix} E_x \\ 0 \end{bmatrix}. \tag{2.89}$$

To model the propagation through several optical components, the Jones matrix of the total system of N components can be calculated by

$$\bar{\bar{\mathbf{M}}}_{tot} = \bar{\bar{\mathbf{M}}}_N \bar{\bar{\mathbf{M}}}_{N-1} \cdots \bar{\bar{\mathbf{M}}}_i \cdots \bar{\bar{\mathbf{M}}}_2 \bar{\bar{\mathbf{M}}}_1 = \prod_{i=N}^{1} \bar{\bar{\mathbf{M}}}_i, \tag{2.90}$$

where component 1 is the first encountered optical component in the system. With a brief introduction to Jones calculus, we are well equipped to dive in to polarization ray tracing.

The goal of polarization ray-tracing is the introduction of the E-field vector as a property of a ray. The following treatment of the polarization ray-tracing is based on *Three-dimensional polarization ray-tracing calculus I* by Yun, Crabtree and Chipman [61]. The E-field vector is given in global coordinates of (x, y, z), as

$$\mathbf{E} = \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}, \tag{2.91}$$

where E_x, E_y and E_z are complex amplitudes. Note x and y always refer to the global coordinate system, as opposed to the field components of $\mathbf{J}$. Similarly as with Jones calculus, the evolution of the state of polarization of a ray by optical components is modeled with matrices, called Polarization ray-tracing matrices $\bar{\bar{\mathbf{P}}}$. The two E-fields before and after interactions with an optical component, $\mathbf{E}_{i-1}$ and $\mathbf{E}_i$, are related through $\bar{\bar{\mathbf{P}}}_i$ by

$$\mathbf{E}_i = \bar{\bar{\mathbf{P}}}_i \mathbf{E}_{i-1}. \tag{2.92}$$

The matrix $\bar{\bar{\mathbf{P}}}$ is a 3×3, and the net ray-tracing matrix $\bar{\bar{\mathbf{P}}}_{tot}$ for several components can be calculate by

$$\bar{\bar{\mathbf{P}}}_{tot} = \prod_{i=N}^{1} \bar{\bar{\mathbf{P}}}_i, \tag{2.93}$$

in a similar fashion as equation 2.90. In order to simulate physical polarization elements the global E-field components must be transformed into the local coordinates of the polarization elements. For reflection and refraction of flat surfaces the s- and p-polarization states, along with the wave vector $\hat{\mathbf{k}}$ form a natural basis $\{\hat{\mathbf{s}}, \hat{\mathbf{p}}, \hat{\mathbf{k}}\}$. Let the local coordinates of the ray before and after the q'th surface be given by, $\{\hat{\mathbf{s}}_q, \hat{\mathbf{p}}_q, \hat{\mathbf{k}}_q\}$ and $\{\hat{\mathbf{s}}'_q, \hat{\mathbf{p}}'_q, \hat{\mathbf{k}}_{q-1}\}$. In the ray tracer $\hat{\mathbf{v}}_q = \hat{\mathbf{k}}_q$. The equations for $\hat{\mathbf{s}}_q$, $\hat{\mathbf{s}}'_q$, $\hat{\mathbf{p}}_q$ and $\hat{\mathbf{p}}'_q$ can then be written in terms of the wave vectors as

$$\hat{\mathbf{s}}_q = \frac{\hat{\mathbf{k}}_{q-1} \times \hat{\mathbf{k}}_q}{|\hat{\mathbf{k}}_{q-1} \times \hat{\mathbf{k}}_q|}, \qquad \hat{\mathbf{p}}_q = \hat{\mathbf{k}}_{q-1} \times \hat{\mathbf{s}}_q, \qquad \hat{\mathbf{s}}_q = \hat{\mathbf{s}}'_q, \qquad \hat{\mathbf{p}}'_q = \hat{\mathbf{k}}_q \times \hat{\mathbf{s}}_q. \tag{2.94}$$

Note the direction $\hat{\mathbf{s}}_q$ is conserved across the interface. With the definitions of the local coordinates we may formulate transformation matrices to transform the E-field $\mathbf{E}_{q-1}$ into the $\left\{\hat{\mathbf{s}}_q, \hat{\mathbf{p}}_q, \hat{\mathbf{k}}_{q-1}\right\}$ basis and for transforming the E-field $\mathbf{E}_q$ of the $\left\{\hat{\mathbf{s}}'_q, \hat{\mathbf{p}}'_q, \hat{\mathbf{k}}_{q-1}\right\}$ basis back out to the global coordinates. These transformation matrices are $\bar{\bar{\mathbf{Q}}}_{in,q}^{-1}$ and $\bar{\bar{\mathbf{Q}}}_{out,q}$ respectively, and are given by

$$\bar{\bar{\mathbf{Q}}}_{in,q}^{-1} = \begin{bmatrix} \hat{s}_{x,q} & \hat{s}_{y,q} & \hat{s}_{z,q} \\ \hat{p}_{x,q} & \hat{p}_{y,q} & \hat{p}_{z,q} \\ \hat{k}_{x,q-1} & \hat{k}_{y,q-1} & \hat{k}_{z,q-1} \end{bmatrix}, \qquad \bar{\bar{\mathbf{Q}}}_{out,q} = \begin{bmatrix} \hat{s}_{x,q} & \hat{p}'_{x,q} & \hat{k}_{x,q} \\ \hat{s}_{y,q} & \hat{p}'_{y,q} & \hat{k}_{y,q} \\ \hat{s}_{z,q} & \hat{p}'_{z,q} & \hat{k}_{z,q} \end{bmatrix}. \tag{2.95}$$

For refraction and reflection described in terms of $\left\{\hat{\mathbf{s}}, \hat{\mathbf{p}}\right\}$ the new E-field components can be calculated by the matrices $\bar{\bar{\mathbf{J}}}_{t,q}$ and $\bar{\bar{\mathbf{J}}}_{r,q}$, which are given by

$$\bar{\bar{\mathbf{J}}}_{t,q} = \begin{bmatrix} \alpha_{s,t,q} & 0 & 0 \\ 0 & \alpha_{p,t,q} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \bar{\bar{\mathbf{J}}}_{r,q} = \begin{bmatrix} \alpha_{s,r,q} & 0 & 0 \\ 0 & \alpha_{p,r,q} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.96}$$

Where the subscript t is for refraction, r is for reflection, and s and p indicate polarization. $\alpha_{s,t,q}$ and $\alpha_{p,t,q}$ are the s and p-amplitude transmission coefficients and $\alpha_{s,r,q}$ and $\alpha_{p,r,q}$ are the s and p-amplitude reflection coefficients. The amplitude coefficients can be calculate with Fresnel's equations for dielectric surfaces. The final polarization ray-tracing matrix for refraction and reflection is then

$$\bar{\bar{\mathbf{P}}}_q = \bar{\bar{\mathbf{Q}}}_{out,q} \bar{\bar{\mathbf{J}}}_q \bar{\bar{\mathbf{Q}}}_{in,q}^{-1}. \tag{2.97}$$

For multiple bounces we simply compute $\bar{\bar{\mathbf{P}}}_q$ for each surface q.

Constraints

As $\bar{\bar{\mathbf{P}}}_i$ is defined by equation 2.92, $\bar{\bar{\mathbf{P}}}_i$ is underconstrained. Equation 2.92 is equivalent with the Jones matrix equation

$$\mathbf{J}_i = \bar{\bar{\mathbf{M}}}_i \mathbf{J}_{i-1}, \tag{2.98}$$

except that 2.92 is formulated in global coordinates, as opposed to the local coordinates of $\mathbf{J}$. Any independent polarization state $\mathbf{J}$ can be represented in a basis of only two vectors, $\mathbf{J}_a$ and $\mathbf{J}_b$. As $\mathbf{J}$ is constrained to the transverse plane, $\bar{\bar{\mathbf{M}}}_i$ is constrained fully by the two matrix equations

$$\mathbf{J}'_a = \bar{\bar{\mathbf{M}}}_i \mathbf{J}_a, \qquad \mathbf{J}'_b = \bar{\bar{\mathbf{M}}}_i \mathbf{J}_b. \tag{2.99}$$

By fully constrained, it means that any change to two polarization states $\mathbf{J}_{i-1}$ and $\mathbf{J}_i$, before and after, are uniquely described by the matrix $\bar{\bar{\mathbf{M}}}_i$. In equation 2.92 any polarization state can similarly be represented by two basis vectors $\mathbf{E}_a$ and $\mathbf{E}_b$. If $\bar{\bar{\mathbf{P}}}_i$ operates on each basis we get

$$\mathbf{E}'_a = \bar{\bar{\mathbf{P}}}_i \mathbf{E}_a, \qquad \mathbf{E}'_b = \bar{\bar{\mathbf{P}}}_i \mathbf{E}_b. \tag{2.100}$$

which amounts to six equations, one for each rows, but $\bar{\bar{\mathbf{P}}}_i$ has 9 elements. To uniquely define the renaming 3 elements of $\mathbf{P}_i$, three new constraints are needed. The new constraints can be defined by letting $\bar{\bar{\mathbf{P}}}_i$ operate on the wave vector $\hat{\mathbf{k}}_{i-1}$, resulting in

$$\bar{\bar{\mathbf{P}}}_i \hat{\mathbf{k}}_{i-1} = \gamma \hat{\mathbf{k}}_i. \tag{2.101}$$

Only two values of γ are allowed 0 or 1. Either value will result in three new valid constraints. In accordance with Yun, Crabtree and Chipman [61], let us set $\gamma = 1$, thus uniquely defining each element of $\bar{\bar{\mathbf{P}}}_i$ by the following nine equations:

$$\bar{\bar{\mathbf{P}}}_i \hat{\mathbf{k}}_{i-1} = \hat{\mathbf{k}}_i, \qquad \mathbf{E}'_a = \bar{\bar{\mathbf{P}}}_i \mathbf{E}_a, \qquad \mathbf{E}'_b = \bar{\bar{\mathbf{P}}}_i \mathbf{E}_b. \tag{2.102}$$

Polarimetry implementation

Our implementation of polarization ray tracing for simulating polarimetry involves sending out H and V polarized rays from the emitter and detecting returning rays through a H or V-polarization filter. If we assume a monostatic detector and emitter setup, it is useful to define two different coordinate systems, a global $\left\{\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}\right\}$ with all the scene information and a local emitter and detector coordinate system $\left\{\hat{\mathbf{V}}, \hat{\mathbf{H}}, \hat{\mathbf{k}}_e\right\}$, defined by the polarization states V and H and the wave vector of the emitted signal $\hat{\mathbf{k}}_{\mathbf{e}}$, which moving forward will be denoted $\hat{\mathbf{k}}$. In the $\left\{\hat{\mathbf{V}}, \hat{\mathbf{H}}, \hat{\mathbf{k}}\right\}$ basis, V and H polarized E-fields are given by

$$\mathbf{V}_{\{VHk\}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{H}_{\{VHk\}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \tag{2.103}$$

The E-fields in the global coordinates can then be calculated by

$$\mathbf{E}_{\{xyz\}} = \bar{\bar{\mathbf{T}}}\mathbf{E}_{\{VHk\}}, \tag{2.104}$$

with the transformation matrix given by

$$\bar{\bar{\mathbf{T}}} = \begin{bmatrix} \hat{V}_x & \hat{H}_x & \hat{k}_x \\ \hat{V}_y & \hat{H}_y & \hat{k}_y \\ \hat{V}_z & \hat{H}_z & \hat{k}_z \end{bmatrix}. \tag{2.105}$$

The ray generation step now involves generating E-field vectors for each ray based on the emitter position ($\mathbf{P}_e$), scene center ($\mathbf{P}_s$) and emitter polarization. Rays are launched along the emitter vector $\hat{\mathbf{v}}_e = (\mathbf{P}_s - \mathbf{P}_e)/|\mathbf{P}_s - \mathbf{P}_e|$, with random variations in $\hat{\mathbf{v}}_e$ for a diverging beam. In principle $\hat{\mathbf{k}}_e$ for each ray can be written as

$$\mathbf{k}_e = \hat{\mathbf{v}}_e + \delta\mathbf{v}, \tag{2.106}$$

where $\delta\mathbf{v}$ are small random variations. The polarization vectors $\hat{\mathbf{H}}$ and $\hat{\mathbf{V}}$ in the global basis are calculated from the emitted wave vector for each ray and the slant angle θ, by

$$\hat{\mathbf{k}}_e^p = \frac{\hat{\mathbf{k}}_e - \hat{\mathbf{z}}k_e^z}{|\hat{\mathbf{k}}_e - \hat{\mathbf{z}}k_e^z|}, \qquad \hat{\mathbf{V}} = \hat{\mathbf{z}}\cos\theta + \hat{\mathbf{k}}_e^p \sin\theta, \qquad \hat{\mathbf{H}} = \hat{\mathbf{V}} \times \hat{\mathbf{k}}_e, \tag{2.107}$$

where $\hat{\mathbf{k}}_e^p$ is the normalized projection of $\hat{\mathbf{k}}_e$ into the xy-plane.

To summarize, each ray in the polarization ray tracer has the following four properties, an origin $\mathbf{O}$, a direction $\hat{\mathbf{v}}$, a traveled distance from the emitter d and an electric field vector $\mathbf{E}$. At ray generation and launching the ray is initialized with $\mathbf{O} = \mathbf{P}_e$, $\hat{\mathbf{v}} = \hat{\mathbf{k}}_e$ and $d = 0$, with the E-field being assigned to either H or V (e.g. $\mathbf{E} = \hat{\mathbf{H}}$) depending on the polarization of the emitter. For each intersection with the scene the new polarization state of the ray is calculated with equation 2.97 and d is updated by the value of t, the ray parameter. At the intersection the ray is given a new direction based on the scattering mechanism defined in the ray-tracer and the origin is moved to $\mathbf{P}(t)$.

For each pair of emitter and detector polarizations, (e.g. HH), we generate a SAR image of S_{hh}. By going through all polarization pairs, we map every element of the scattering matrix, one element at the time. In practice all elements may be calculated at once by giving each ray two polarization, one for H and V and treating them independently through the ray-tracer. At the detector we then let both E-fields get detected through both polarizers resulting in all four scattering matrix elements. With this method we minimize the amount of ray-tracing needed, which is the computationally heavy part of the algorithm.

Detection of Rays

The following implementation for detecting rays can be omitted if enough rays are able to be launched, i.e. by implementing polarization into the Optix ray-tracer. However for slower CPU code faster convergence can be obtained by allowing for back scattering. With every ray-scene intersection, some fraction of the new ray with the new polarization state is sent straight back to the detector. In practice this is done by calculating the distance from an intersection to the detector l, and adding l to the traveled distance d and recording the detected intensity of the E-field after going through the polarizer at the detector.

2.5 Rasterization - Shooting Rays

This is an introduction to a fast shooting rays method for generating radar signals, in which bounces are omitted, akin to rasterization. The main algorithm involves finding the closest ray-surface intersection with the scene, and calculating back scattering. There are two main ways in which this beats out a ray tracer for speed. Firstly by omitting bounces, and the ray-surface intersections calculation associated with bounces, and secondly backscattering is assumed to always hit the detector, skipping calculating ray-detector intersections. For N rays this method only needs to calculate N ray-surface intersections, compared to a ray tracer that needs minimum $2N$ ray-surface calculations, more with bounces. The main ray-surface intersection algorithm is the same as presented in section 2.4. The nuances of this method are related to calculating weightings of the back scattered rays. Here, a weight is used to control the intensity of the received final signal in the shading step. A simple weighting scheme is to weigh the backscattered ray by

$$w_n = -\hat{\mathbf{n}} \cdot \hat{\mathbf{v}}, \tag{2.108}$$

where $\hat{\mathbf{n}}$ is the surface normal at the intersection point, and $\hat{\mathbf{v}}$ is the ray direction. With this definition the intensity of the returning backscatter is weighted by how perpendicular the hit surface is to the source. Alternatively, one may also define the weights as $w_1 = 1$ for all intersections. In figure 2.9, the backscattered surfaces have been colored according to each weighting scheme, with white surfaces having higher weights.

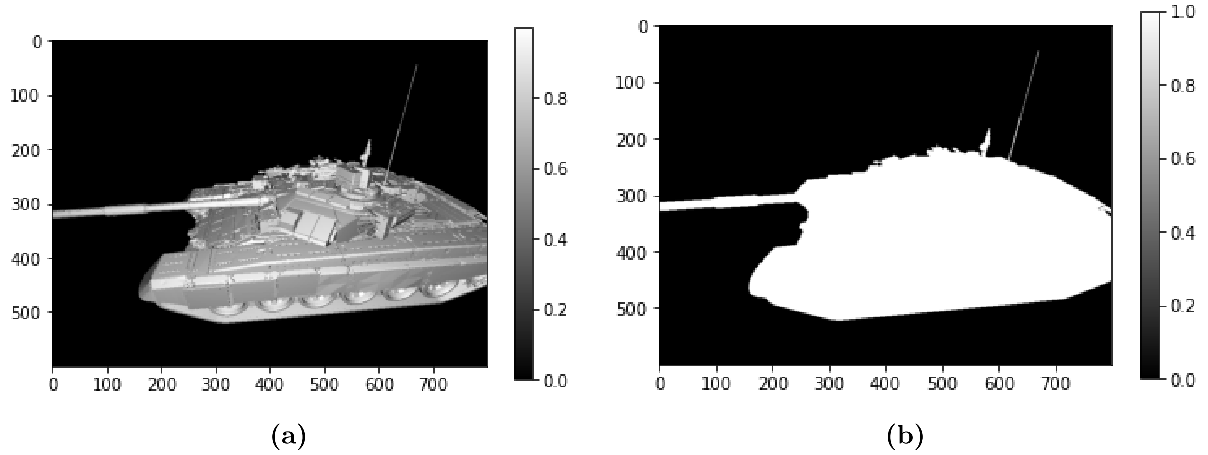


Figure 2.9. *T72 tank model with its surfaces shaded according to the weights of the backscattered rays (**a**) for the w_n and (**b**) for the w_1 schemes.*

This method is inspired by the method *shooting and bounding rays*, in which rays are split into a backscattered ray and a reflected ray on intersection. The intensity of the backscattered ray is then determined by physical optics calculations. Similarly the weights of the backscattered rays can also be based on more physical calculations.

Method 3

3.1 Ray Tracing on a GPU

Ray tracing as described in section 2.4 was implemented and written in Python[1], where the resulting program is executed on the central processing unit (CPU). However, the ray tracing algorithms can be executed in parallel since all rays are treated with the same set of instructions and are not dependent on each other. This parallel nature can be exploited by utilizing a graphics processing unit (GPU), which is optimized for parallel workloads. This is desirable since it can reduce the computational time of our simulations. Before describing how this is done using the NVIDIA Optix ray tracing engine[2], a brief overview on the GPU structure and programming will be given in the following section.

3.1.1 Parallel Computing on a GPU

The name GPU originates from the task of rendering 2D images in computer graphics, where calculations for different pixels can be carried out independently with similar instructions in parallel. The main difference between a CPU and GPU is that a CPU is optimized for executing operations in a sequence with low latency while a GPU is optimized for massive parallel operations but has a higher latency, which is illustrated in figure 3.1.

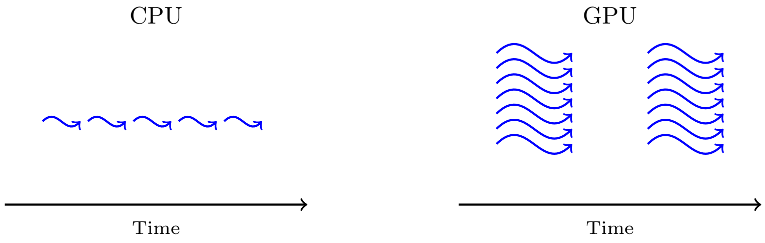


Figure 3.1. *Difference between the sequential and parallel processes in a CPU and GPU respectively. The curved arrows indicate operations.*

While CPUs can also run operations in parallel, modern CPUs only have between a few and tens of cores, while GPUs can have between a thousand and tens of thousands of cores. Comparisons in the literature have shown a reduction in computational time by a factor of 1000 to 10000 when performing ray tracing on the GPU compared to the CPU[62, 63]. Hence, to achieve a higher throughput in our radar simulations, the ray tracing software must be changed to run on a GPU.

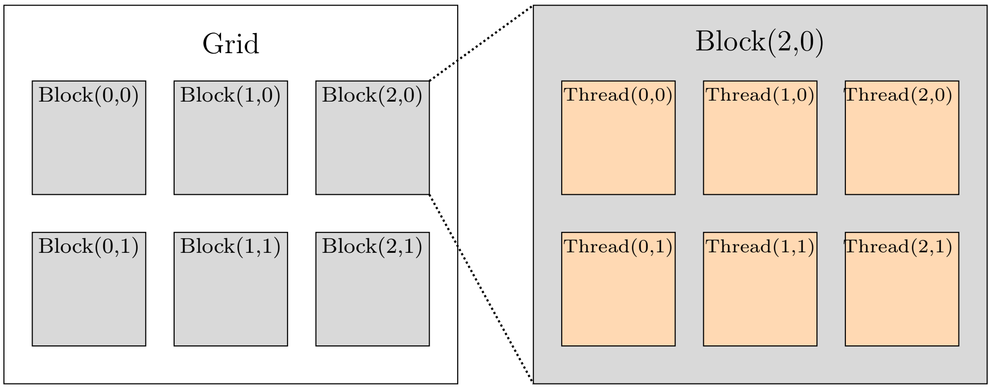


Figure 3.2. *The grid structure used to utilize the parallel threads in GPU programming.*

One of the most used methods to utilize the GPU for computing is CUDA[64], which is an extension to the C and C++ programming languages. Importantly for CUDA, it has a few abstraction levels which are different from more traditional sequential programming. Functions in CUDA are called kernels which consists of many threads we want to run in parallel. Here a thread is a single to a few instructions or operations. The threads are then organized in blocks which are indexed in a grid. This structure can be seen in figure 3.2. All blocks will have the same number of threads and the threads inside each block can work together and share memory. Individual blocks are independent and can be run in parallel[64]. The number of blocks which can be run in parallel depends on the hardware. Thus a given set of blocks is run in parallel at a time and hereafter a new set is queued and run until the grid is finished.

To change and optimize the different ray tracing functions from the CPU ray tracer to utilize a GPU is not a trivial task. Therefore NVIDIA Optix was used instead since it has many ray tracing functions which are already implemented and highly optimized to GPUs.

3.1.2 Nvidia Optix

Optix is a ray tracing application programming interface (API) developed by NVIDIA since 2009 and is written in the C++ and CUDA programming languages. The API allows for a single ray programming model and many ray tracing functions are built-in, such that much of the programming burden for optimization can be avoided[2].

Program Structure

The overall structure of Optix will be briefly described in this section and the main source behind the presented information is the Optix 8.0 programming guide[65]. Optix consist of four main blocks, which are the acceleration structure, program pipeline, shader binding table and launch, which can be seen in figure 3.3. The acceleration structure block is responsible for optimizing a given mesh to accelerate the ray tracing process. This is done by a bounding volume hierarchy (BVH) algorithm which has been previously described in section 2.4.1. The optix acceleration structure has inbuilt support for meshes which consist of primitives of triangles, curves and spheres. Here, the curves consist of control points and the surface is characterized by a spline curve. However, in this project only triangle primitives were used in the meshes. A raw mesh is loaded into the GPU memory and the acceleration structure is generated on the GPU.

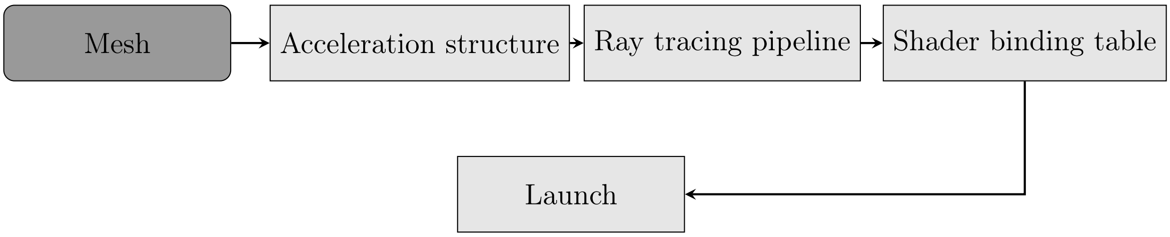


Figure 3.3. *Program structure of NVIDIA Optix.*

Ray Tracing Pipeline

The pipeline block is responsible for linking different programs which is invoked when tracing a ray. These programs include the ray generation, closest hit and miss program, which controls how rays are casted, what happens when they intersect a mesh and when they dont intersect the mesh, respectively. The programs mentioned above are user defined, written in CUDA code and are the most important part for a physical simulation that decides how the rays will behave. Hence a section with more in-depth description of these programs can be found below. Furthermore, algorithms to traverse the BVH and ray intersection with mesh primitives is also included in the pipeline, they are however already built-in functions in Optix and are hardware specific. Different GPU architectures prefer different traverse and intersection algorithms for performance and Optix chooses an algorithm depending on the available hardware.

Shader Binding Table

The role of the shader binding table is to organize data and memory locations from the acceleration structure and program pipeline to ensure optimal and fast access to this data during the launch process. The shader binding table is located as an array on the GPU and some of the most important information it can hold is material types related to the mesh. When a ray intersects a triangle, the material type of the triangle can be obtained fast from the shader binding table and a new ray can then be casted with the characteristics of the material.

Launch

The launch block initiates the ray tracing process on the GPU. Here a kernel is started where an array of threads in parallel will begin the ray tracing pipeline starting with ray generation and ending with the miss program, or if the process has otherwise been terminated. The built-in launch function in Optix manages the handling and distribution of threads on the GPU in an optimized manner.

3.1.3 Ray Tracing Pipeline for Radar Simulation

The ray tracing pipeline in this project consists of four programs which are ray generation (RG), traversal, miss (MS) and closest hit (CH). The RG, MS and CH programs are programmable and controls how a ray behaves. Furthermore, the programs can share information through a per ray data (PRD) structure in Optix, which can be ray origin, direction or other relevant values. Below the design of RG, MS and CH programs will be described and the flow can be seen in figure 3.4.

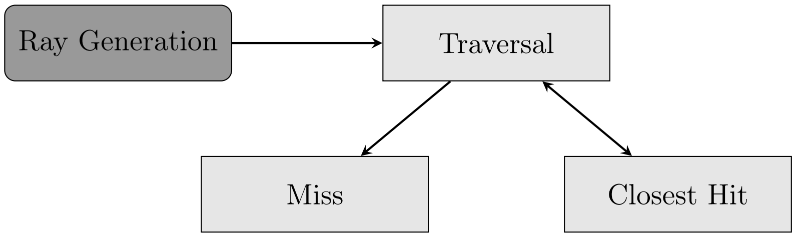


Figure 3.4. *The ray tracing pipeline used in the simulation software. The arrows indicate which programs can call each other.*

Ray generation

The ray generation step starts with choosing how the rays should be emitted. This can be done in a way depending on the desired wavefront or beam profile. If a plane wave is desired, the rays can be casted orthogonal to a plane. A spherical wavefront can be formed by casting the ray orthogonal to the surface of a sphere. After the initial parameters for a ray has been set, an iterative ray tracing process will begin, where a new origin and direction will be generated if the mesh is hit. The ray will be terminated if it misses or has achieved a maximum amount of bounces in the scene. An overview of the process can be seen in algorithm 2.

Algorithm 2 Ray generation program

```
Get new origin and direction of a ray, depending on a chosen beam profile or wavefront.
Set T=0, which will accumulate the distance traveled by a ray.
Set Depth=0, which counts ray bounces.
while Depth<(Max Depth+1) do
    Invoke Traversal algorithm with origin and direction.
    if Miss then
        Break
    else
        Get new origin and direction from CH program.
    end if
    Depth+=1
end while
```

Closest hit

When the traversal algorithm finds intersections with the mesh, the closet hit algorithm is invoked. Here, the intersection point and information about the intersected primitive is used to calculate a new ray origin and direction. The new ray direction is found under a statistical distribution from the scattering profiles mentioned in section 2.4.1. In the case of clutter described in section 2.4.3, the intersected point will be moved to the centroid of the primitive, such that a surface will appear as points. An overview of the process can be seen in algorithm 3.

Algorithm 3 Closest hit

```
Get material of the primitive.
Get ray direction, origin and intersection point.
Find new direction from known material type, previous ray direction and normal to
intersected primitive.
T+=Distance between previous and new origin
Update ray origin and direction.
```

Miss

The detector in the simulation is built into the miss program, where a ray is counted if it intersects a sphere. An array with range bins is allocated on the GPU, where the values from the detected rays will be stored. After the miss program the ray will always be terminated and the ray generation program can start a new ray. An overview of the process can be seen in algorithm 4.

Algorithm 4 Miss

```
Check if ray direction is towards detector sphere.
if Intersection with detector then
    T+=Distance between previous origin and center of detector.
    Save value in range bin corresponding to the value of T.
end if
```

3.2 Range Histogram

In the most general case, a detected ray has an intensity weight w and a total travel distance d. For N received rays, the received radar signal is often computed by a sum of the following form

$$s(t) = \sum_{n=1}^{N} p(t_n) w_n, \tag{3.1}$$

where $p(t_n)$ is the emitted radar pulse shifted in time by the displacement time $t_n = d/c$, and w_n is the ray weight of the n'th ray. For fast oscillating signals, which require fine discretization in time, the sum over many millions of rays can become very computationally expensive. Instead let us introduce the range histogram, by sorting every ray into range bins with a range resolution given by δr. The range histogram $R_H(r)$ is then defined on the interval $[r_0, r_1]$, with $r_0 = 0m$ and r_1 being the upper limit of received ranges from the given scene. A ray with d and w, is then counted in the bin for which $m\delta r < d < (m+1)\delta r$, where m is the bin index, by adding the weight w to the bin total. Each bin is associated with a small range interval, which is equivalent with a small time interval in the displacement time, $\delta t = \delta r/c$. The range histogram with time bins is notated as $R_H(t)$. The sum in equation 3.1 can then instead be computed by a convolution sum of the form

$$s(t) = (R_H * p)(t). \tag{3.2}$$

In practice many of the bins are empty, and a sparse histogram can be saved instead by only saving the bin index and bin total for non empty bins. This speeds up the act of saving the data from the ray tracer by lowering the file size.

3.3 Terahertz Imaging Setup

The basic principle of operation behind the THz-FDS spectrometer, and the constructed ISAR setup is extensively described in previous works [66]. A brief introduction to the setup can be found in appendix A.2. An image of the setup is shown in figure 2.80, where the left hand side contains the THz emitter and detector, the control unit for both lasers and stepper motors. The rotatable stage appears on the right hand side. All is mounted on rails, connected at a point of rotation to allow for the change of slant.

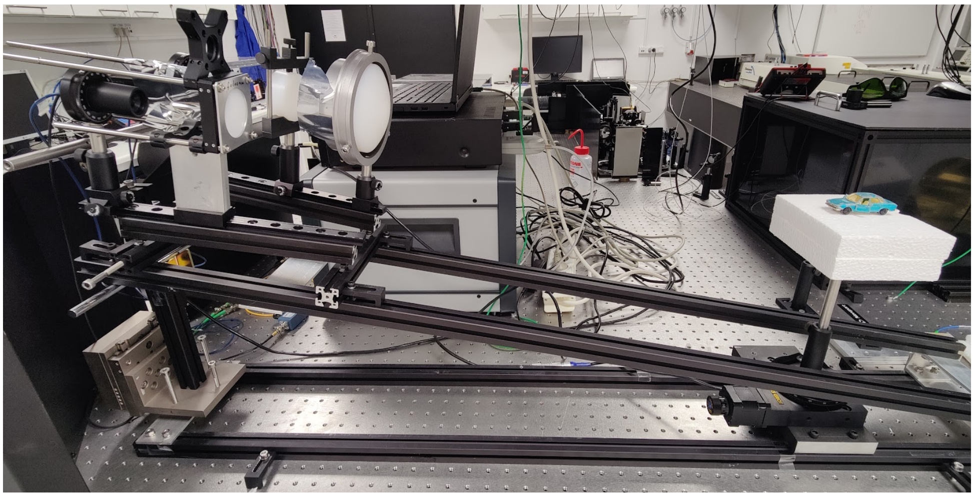


Figure 3.5. *Image of the ISAR setup used for real SAR measurements.*

Several new feature have been added to the setup. These are either small quality of life improvements or allow for completely new methods of imaging compared to the setup in [66]. First, the rotatable stage have been switched to a Melles Griot Nanostepper, which allows full and continuous 360° rotation, compared to the prior 14° limit. The integration angle can consequently be extended to accommodate finer cross-range resolution, thus matching the achievable range resolution.

An additional set of lenses have been designed and mounted on the setup. These allow for a larger spot size to image larger and more complex scenes. An extensive description is presented in section 6.2.1.

Furthermore, InSAR requires for vertical movement of the "radar" platform. Thus a linear stepper motor have been added to the setup to allow for changes in the vertical position of the platform. Due to the nature of the setup, i.e. the platform being mounted on rails with constant range regardless of slant angle. The linear motion of the stepper motor can be approximated to a change in slant angle following the geometry presented in figure 3.6

$$\theta_2 = \tan^{-1}\left(\frac{H+\Delta h}{\cos\theta_1 \cdot R}\right), \tag{3.3}$$

where H is the height prior to the change in height, R is the range, and θ_1 and θ_2 are the angles prior to and after the change of height Δh respectively.

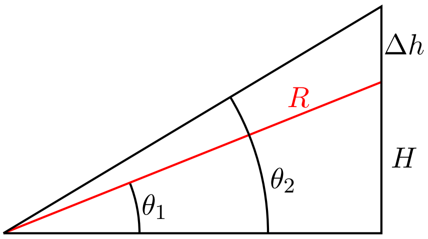
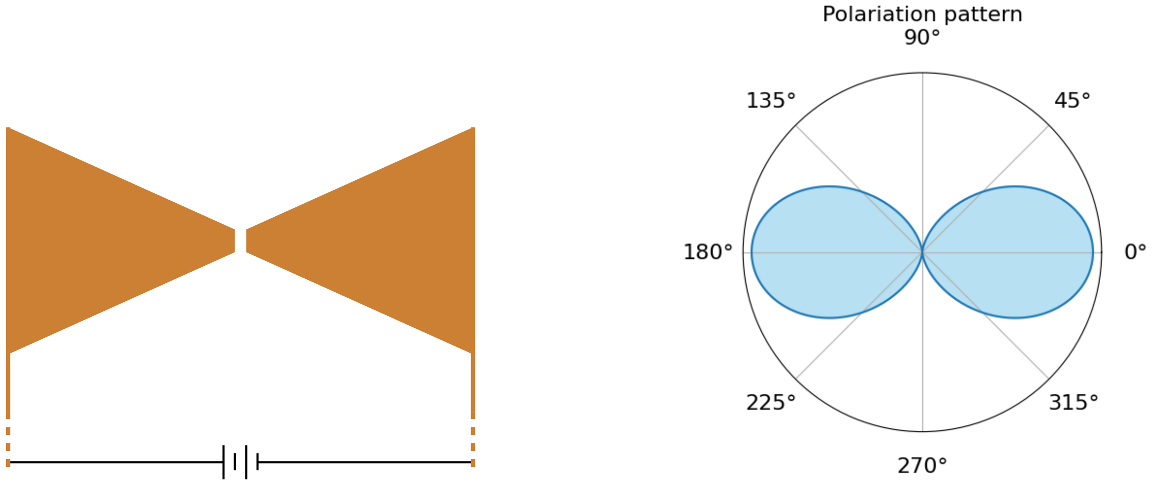


Figure 3.6. *Triangular geometry of THz SAR rig.*

Last additional feature is the change of mounts for emitter and detector, such that they can rotate in the plane perpendicular to the look direction. This feature is important due to the nature of the THz generation. The antenna of the photomixer is a bow-tie antenna, A simplified illustration of the bow-tie antenna is shown in figure 3.7a.[35]



(a) **(b)**

Figure 3.7. *(**a**) Schematic of a bow-tie antenna, and (**b**) the expected polarization pattern of the emitted THz radiation.*

The triangular shape of the antenna allows for currents to run in other directions than the major axis, which allows emission of THz radiation in a quasi-linear polarization, where a part of the light is emitted linearly, with some cross-polarization.[35] This is illustrated in the radiation pattern of the antenna, see figure 3.7b. The radiation pattern is an idealized model of what is expected, which is the inevitable cross-polarization.

3.4 Processing of THz-FDS Data

In this section the methodology of going from raw THz-FDS data to a SAR image will be described. The THz-FDS setup has coherent detection and measures in discrete steps along a frequency sweep. The measured photocurrent, i_{ph}, is proportional to the electric field and on the form

$$i_{ph}(f) = \mathrm{Re}\{A\,(f)\,e^{-i2\pi f\Delta t)}\}. \tag{3.4}$$

To obtain the phase and amplitude, which is required to format a SAR image, a Hilbert transform can be applied to the measured photocurrent. However, the Hilbert transform requires data points with a uniform frequency spacing, hence the data is first interpolated by a linear interpolator. After the interpolation and Hilbert transform, the following complex valued signal is obtained

$$S(f) = A\,(f)\,e^{-i2\pi f\Delta t}. \tag{3.5}$$

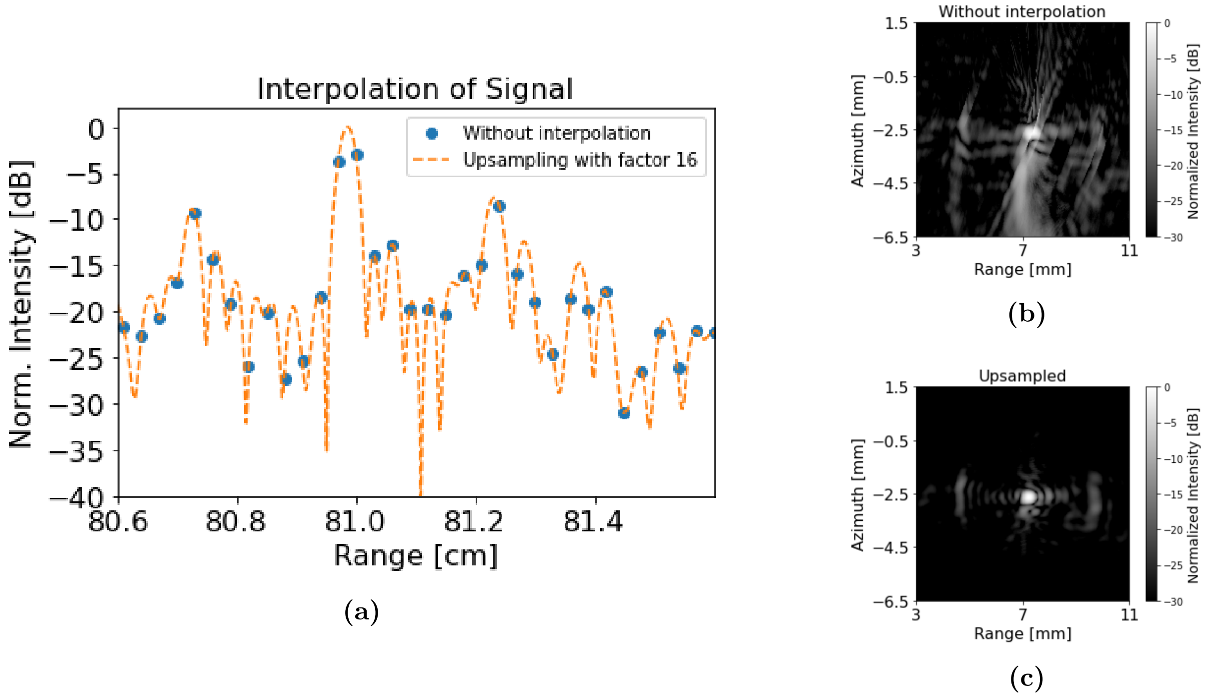
$S(f)$ can then be transformed by an inverse Fast Fourier Transform (IFFT) to obtain the response in time-domain,

$$s(t) = IFFT(S(f)). \tag{3.6}$$

In section 2.1.1 it was assumed that the time-domain signals have a chirp waveform however, that is not the case with the experimental THz data. Here reflected signal will appear as narrow sinc-like peaks at their respective time delay, hence the range compression step is not required. Otherwise the BPA is performed as described in section 2.1.1.

Upsampling

Often it is desirable to interpolate $s(t)$ to reduce residual phase errors in the BPA, that can cause artifacts in the generated image. This comes from equation 2.15 where the value taken from a signal will be closest in time delay to the center of a scattering cell, hence a higher sampling will give less error when choosing the closest value in range. $s(t)$ can be interpolated by zero padding $S(f)$ before taking the IFFT. A comparison between with and without interpolation of the time-domain signal can be seen in figure 3.8. The unit normalized intensity in decibel for a signal is found by $20\log_{10}\left(\frac{s(t)}{max(s(t))}\right)$ and similary for image pixel values. Here the target is a $750\mu m$ diameter metal sphere measured from 430 to 930Ghz over a span of 50° in azimuth. The upsampled signal has 16 times more data points in the same range span.



(a)

(b)

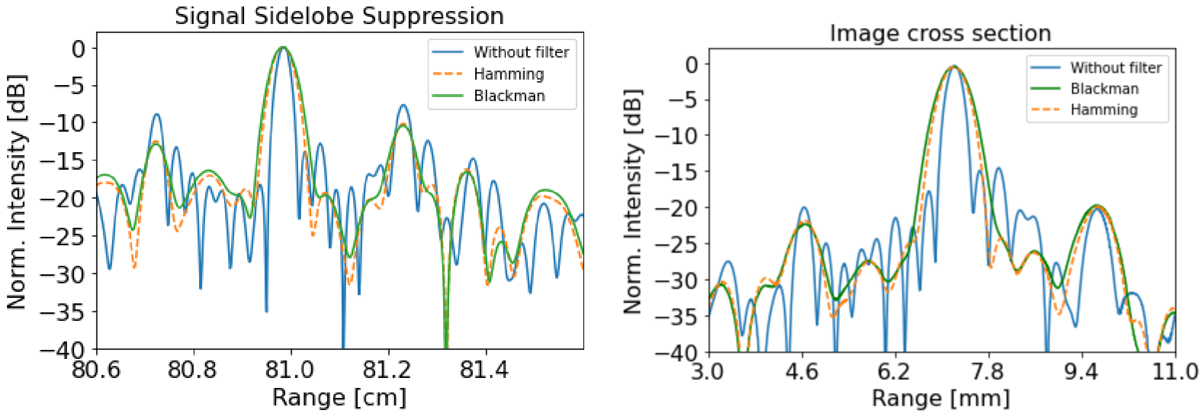
(c)

Figure 3.8. *(**a**) Comparison between Initial data points and a the same signal which has been upsampled by zero padding before IFFT. Generated SAR images (**b**) without and (**c**) with upsampling.*

Sampling higher than a factor of 16 was found not to improve image quality significantly and therefore most experimental data presented in this project will use the factor of 16. In figure 3.8c the metal sphere is located in the center of the image with sidelobes in the range direction. Furthermore, two artifacts are present proximately 2.5mm in front and behind the actual sphere as two strips parallel with the azimuth direction. These artifact will be described later.

Sidelobe Suppression

The sidelobes can be suppressed by applying filters in the frequency domain of the signals. The THz-FDS system has a higher dynamic range at the lower frequencies, which can be seen in figure 3.15. Hence, the amplitudes across the signals frequency spectra are firstly normalized before the filters are applied. A Hamming and Blackman filter have been applied to a signal and the differences can be seen in figure 3.9a. Here it can be seen that the sidelobe intensities are reduced, but it comes at a cost of a broadened main lobe.



(a)

(b)

Figure 3.9. *(**a**) The effect of filters on the main lobe and side lobes in a single signal. Range axis is here defined as distance to emitter/detector. (**b**) Cross section along range of generated images with and without filters seen in figure 3.10. Range is here defined as distance to rotation center of the scene.*

Images generated with and without a filter can be seen in figure 3.10. The images have been generated with the same parameters except for the filter. A cross section of pixel values parallel with the range direction can be seen in figure 3.9b. Both filters suppress the side lobes in the images but the main lobe becomes more elongated in the range direction. Hann and Taylor filters were also experimented with but gave similar results to the Hamming and Blackman filters. Hamming resulted in a slightly narrower main lobe while maintaining similar side lobe suppression as the other filters; hence Hamming is used for THz data unless stated otherwise. While the sidelobes in range have been suppressed, there is still azimuth sidelobes present.

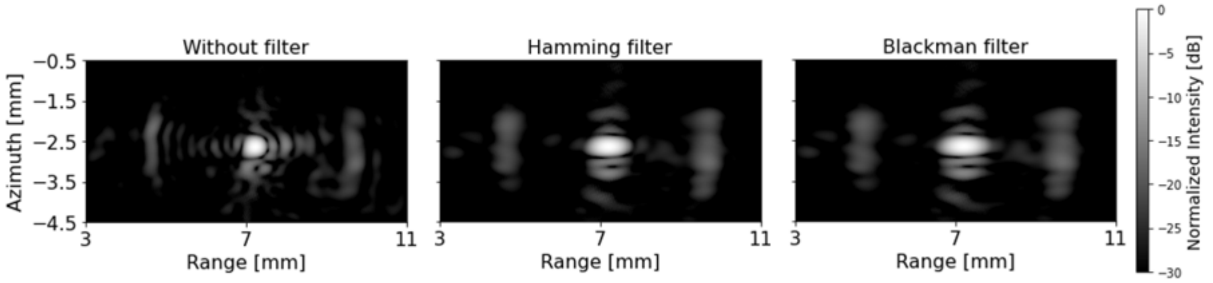


Figure 3.10. *Image without filter and images with Hamming and Blackman filter respectively. Sidelobes are suppressed by the filters while the main lobe becomes wider in range.*

Suppression of Artifacts

The artifacts in the images mentioned before are still present after the range filtering for side lobe suppression seen in figure 3.10. The artifacts were found to be present when measuring other objects than the metal sphere and do not appear to be a product of the measured scene. Instead, it is assumed to be artifacts in the measured data, which we would like to remove or at least suppress. The artifacts always appear to be at a consistent time shift in front of and behind the main signal. This can be seen in figure 3.11a, where the intensity of signals measured at different azimuth angles is plotted along the range direction. The artifacts can be seen at a constant range shift along all the different signals. Assuming our measured data in frequency domain is on the form

$$S(f) = S_{ideal}(f) + E_1(f) + E_2(f) + \phi, \tag{3.7}$$

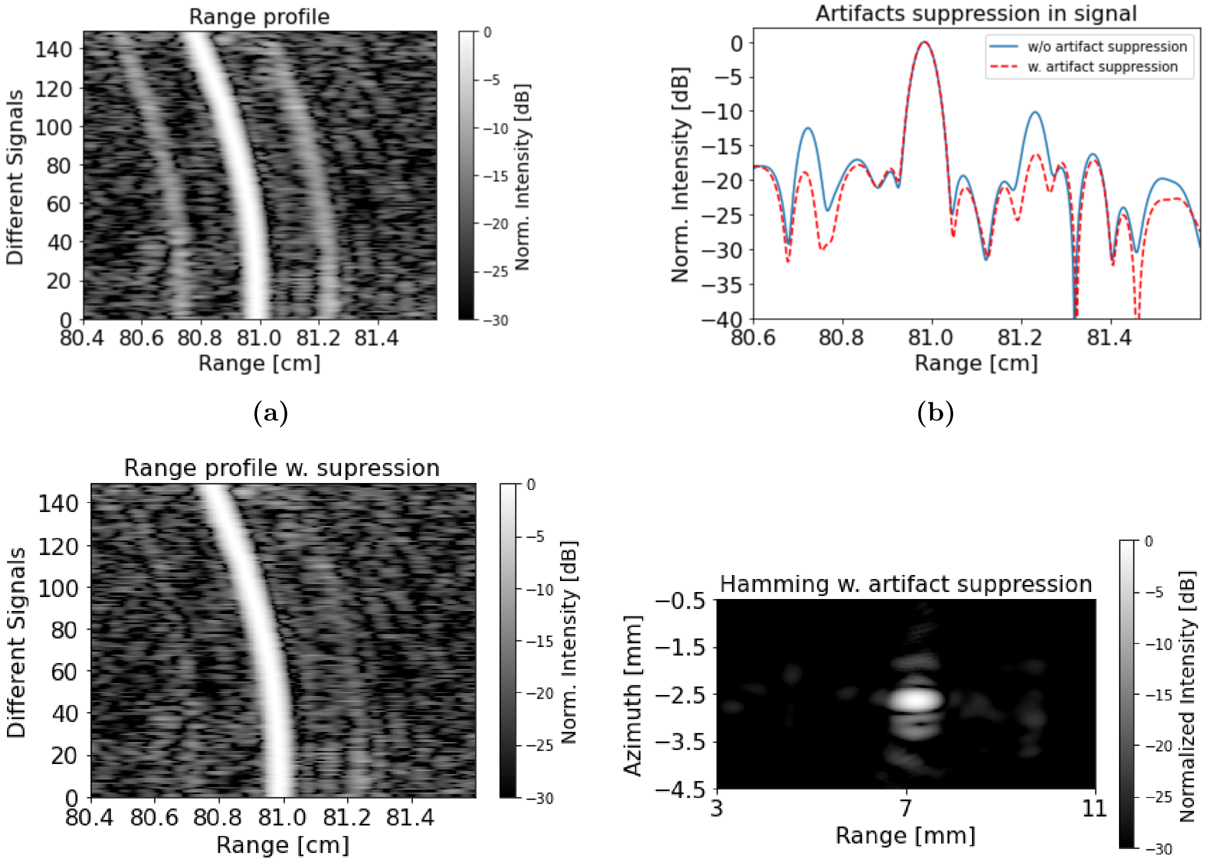
where S_{ideal} is the main lobe, E_1,E_2 are the artifacts and ϕ is noise. The artifacts can then be expressed by weighting and shifting the signal from the main lobe

$$E(f) = S_{ideal}(f)Ae^{i2\pi f\Delta t}. \tag{3.8}$$

Here, Δt is the time shift and A is a weight. Ignoring the noise term, the ideal signal from the main lobe can found by

$$S_{ideal}(f) = \frac{S(f)}{1 + A_1e^{i2\pi f\Delta t_1} + A_2e^{i2\pi f\Delta t_2}}. \tag{3.9}$$

Setting $A_1 = A_2 = 0.15$ and $\Delta t_1 = -\Delta t_2 = \frac{2\cdot 2.45\text{mm}}{c}$ will lower the intensity of the artifacts significantly across all the measured signals, which can be seen in figure 3.11c. These values were found heuristically. The resulting image in figure 3.11d will have the artifacts suppressed.



(a)

(b)

(c)

(d)

Figure 3.11. *(**a**) Range profile with artifacts and (**c**) without. (**b**) Single signal with and without suppression of artifact. (**d**) Final image with artifacts suppression, that can be compared to the image without in figure 3.10.*

Autofocus

An important requirement for the BPA is the precise knowledge of the location of the emitter and detector with respect to the imaged scene. In figure 3.12 three images of the metal sphere are plotted where the rotation center of the scene is set at three different locations.

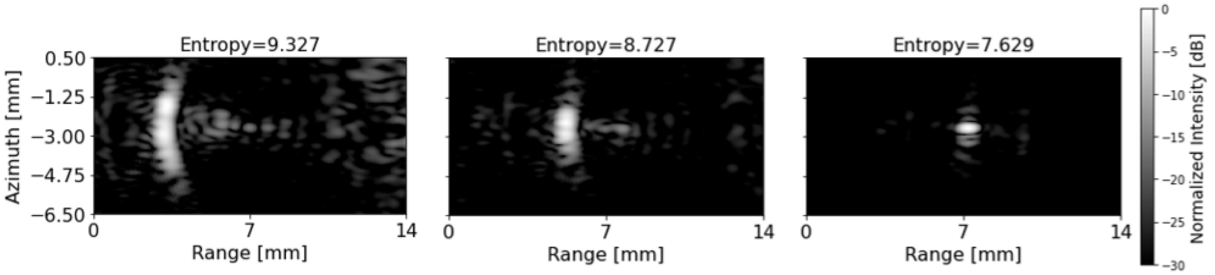


Figure 3.12. *Images of three different rotation centers used in the BPA for the metal sphere. The image to the right have the minimum entropy and the two others have rotation centers shifted 4 and 2*mm *from the minimum.*

Fine-tuning the location manually is time consuming and instead the autofocus algorithm described in section 2.1.3 is used. Entropy is used as a measurement of the image quality from which the rotation center is adjusted iteratively. The outer most right image in 3.12 is the final image with minimum entropy. This approach have been used for all generated THz SAR images in this project.

3.5 Analysis of THz-FDS Parameters

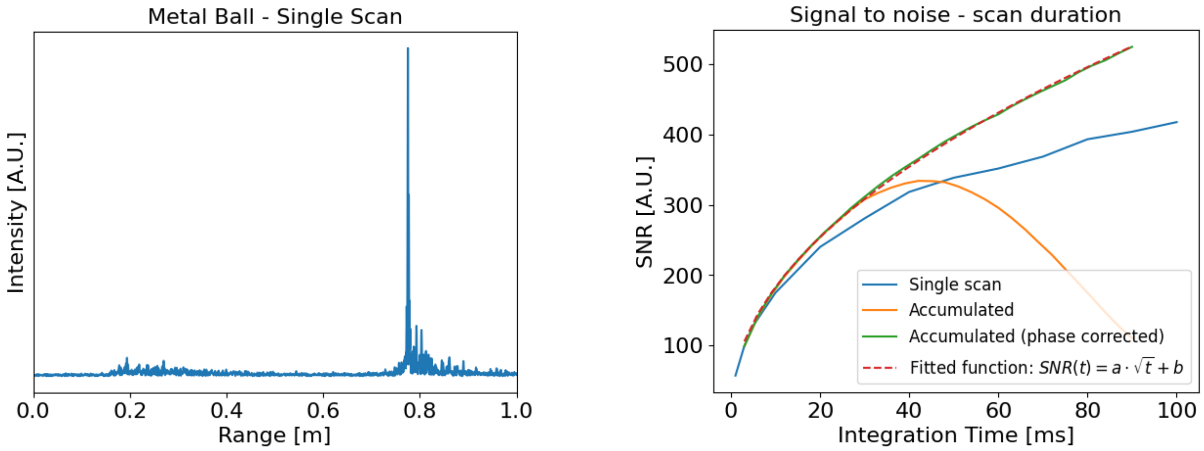
To acquire the best resolution possible while keeping imaging errors at a minimum, several of the THz-FDS parameters must be analyzed thoroughly for their effect on image quality. Additionally, the duration of the experiment must also be considered as many parameters directly influence scan time.

Simply put, a single scan of the TOPTICA TeraScan system divides a frequency interval of bandwidth B_c into an amount of steps N_f of step size $df = B_c/N_f$. The incoming radiation is then collected by the system receiver and accumulate over a specified time commonly known as the integration time. All of these parameters affect the image quality and limitations of the SAR setup, and must be considered when high quality images are to be acquired in a realistic timeframe.

3.5.1 Integration Time and Phase of Signal

The integration time is directly proportional to the total scan time. Thus, doubling the integration time will double the scan time. Say a scan has 5000 frequency steps each with an integration time of 100 ms, this will give a total scan duration of approximately 8 minutes, which by itself appears reasonable. However, if the bandwidth is increased by a factor of 10, the scan time becomes 80 minutes. This is problematic especially when the SAR technique is to be applied, requiring hundreds of scans, causing SAR measurements to have durations up to several weeks. The TeraScan system allows for integration times down to 1 ms, which allows high precision (high bandwidth and high integration angle) SAR measurements to be done within a day. However, decreasing the integration time will in turn also decrease the signal to noise ratio (SNR). A balance must therefore be found between scan time and SNR, where a sufficient SNR can be achieved within a realistic time frame. This can be analyzed by applying a parameter sweep of the integration time in the interval $[1; 100]$ ms.

For the parameter sweep, a metal ball of diameter 2 mm have been measured 25 times for each integration time to calculate an effective average SNR. Figure 3.13a shows the signal of a frequency scan of a metal ball, converted into time domain, where the distance has been calculated.



(a)

(b)

Figure 3.13. *(a) Single scan of a metal ball of diameter* 2 mm. *(b) Signal to noise ratio achieved with different methods of data acquisition.*

The SNR is calculated by first calculating the envelope, with that we then choose an interval where no signal is measured and find the average intensity of the envelope, that would be considered the noise. The maximum signal is then divided by the noise to find the SNR

$$SNR = \frac{signal_{max}}{noise_{avg}}. \tag{3.10}$$

The parameter sweep is programmed into the embedded software, which sets the integration time for each frequency. However, increasing the integration time in the system is not the only method of increasing the SNR. Averaging multiple scans of the same sample should also increase SNR, and is the method of choice for other measuring techniques such as Raman Spectroscopy and X-Ray Diffraction. Therefore, 30 scans of 3 ms integration time have been conducted for comparison with the parameter sweep. Both the parameter sweep and the 30 3 ms scans provide 25 individual scans of the sample, allowing for averaging of the SNR, providing accurate results. Figure 3.13b shows the signal to noise ratio for the two methods of data acquisition. The single scan showed a steady increase in SNR for increasing integration times. The single scan method follows a logarithmic function, thus increasing the integration time will produce decreasing improvements in the SNR. However, the phase corrected accumulated signal proves to be more effective for the SNR. At 90 ms integration time, the accumulated SNR is larger by a factor of 1.3 compared to the single scan SNR. This difference diminishes as the integration time decreases, at 30 ms to a factor of 1.1. Thus, for low integration times, this difference is insignificant, and the single scan is the method of choice due to less steps in post processing of the signal.

Additionally, the SNR usually follows a root function [67]

$$SNR(t) = a \cdot \sqrt{t} + b, \tag{3.11}$$

where t is the integration time, and a and b are fitting parameters. We see this is holds true for the accumulated SNR. However, the single scan method instead follows a logarithmic function. We attribute this discrepancy to phase problems.

Figure 3.13b shows the accumulated SNR with and without phase correction. When there is no phase correction, at an integration time of approximately 30 ms, the SNR starts to be negatively influenced by additional accumulations. This can be attributed to a phase shift in the received signal. The data processing behind the accumulation averages the raw signal. This makes it sensitive to the phase of the signal. If the phase of the peak shifts by π we would see that it reduces the SNR when adding the scan. This is not problematic if the phase shift is insignificant. However, that is not the case for this system. Figure 3.14 shows that across 25 scans with an integration time of 10 ms, the phase of the peak would shift by over π.

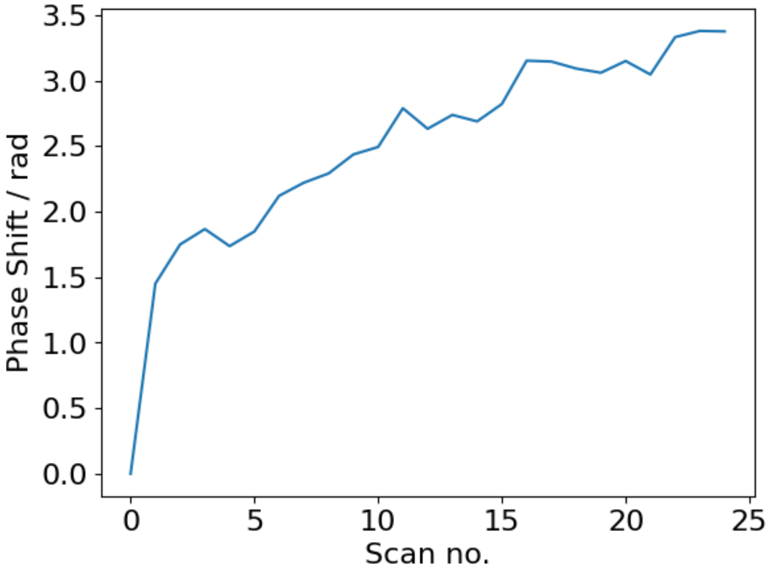


Figure 3.14. *Phase shift of the measured peak from a* 2 mm *metal ball.*

Thus, for long measurements, the phase shift would be considerable, and across multiple individual measurements, the phase will be inconsistent. This problem can be mitigated by correcting the phase, which can be a simple task. A requirement is that an object in the scene needs to be stationary across measurements. However, in ISAR the scene is moving across measurements, which makes phase correction difficult. A stationary point needs to be introduced, which can be mounted on a stand. This will interfere with the scene, and thus will be avoided when not necessary.

3.5.2 Bandwidth and Center Frequency

The inverse proportionality between the range resolution and bandwidth, and center frequency and cross-range resolution, as seen in equation 1.2 and 1.3, indicates a simple method for increasing resolution; increase the bandwidth. There are two obvious upper limits to this method. This, however, comes with implications caused by limitations of the TeraScan system. The system can operate in the frequency interval of [310; 1500] GHz, which sets an upper limit for the achievable bandwidth of approximately 1200 GHz. Furthermore, the dynamic range of the system is not constant throughout all frequencies of the spectrum.

Figure 3.15 shows the achievable dynamic range across the spectrum measured by Toptica for this system specifically.

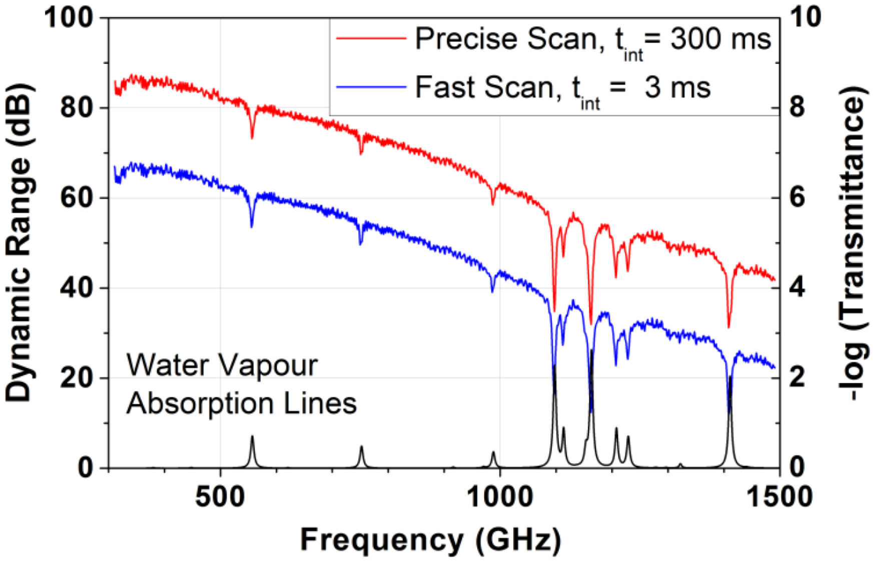


Figure 3.15. *Spectrum provided by Toptica of the dynamic range of the TeraScan1550 (2 mirrors, focal length = 2")*

The dynamic range evidently drops significantly as the frequency of the light increases. This affects the signal to noise ratio and thus the image quality. Additionally, the frequency and absorptive effect of water lines increases with frequency. This introduces a constraint when considering the frequency interval of choice, as both a lower dynamic range and ambient absorption will affect the final image. Thus the band of [310; 1050] GHz would be optimal for synthetic aperture applications.

3.5.3 Frequency Scan Constraint

While bandwidth and integration angle directly influences resolution of the formatted image, other parameters serve as constraints, which, for the most part, does not influence the image after a certain threshold. This includes the step size between two frequencies in the frequency scan, which is directly connected to the unambiguous range of the SAR system. This can be shown through the range resolution and bandwidth. The bandwidth can be expressed as N frequency steps of df length

$$B = N \cdot df. \tag{3.12}$$

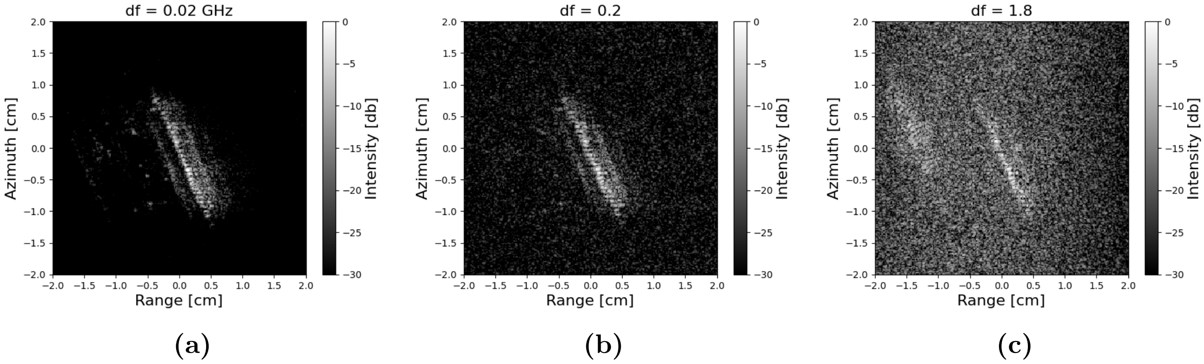
The bandwidth B is used in the expression for the range resolution, equation 1.2. The maximum unambiguous range can be expressed as

$$R_{max} = N \cdot dr. \tag{3.13}$$

Combining the above equations we get the following relation

$$R_{max} = \frac{c}{2df}. \tag{3.14}$$

The range of the system varies, but is centered around 80 cm. Inserting this into equation 3.14 and solving for df sets the constraint to be in the range of 0.1-0.2 GHz



(a) **(b)** **(c)**

Figure 3.16. *SAR image of a microscopy target with df steps of* ***(a)*** 0.02 GHz, *and* ***(b)*** 0.2 GHz, *and* ***(c)*** 1.8 GHz.

Figure 3.16 shows three different SAR images of the same target, with varying frequency step size. These images illustrate the difference for when the constraint is fulfilled and when it is not. As the step size is increased, we see a decrease in signal to noise. This is shown as an increase in noise around the target, and comes as a consequence of the decrease in measuring time due to the net loss in frequency steps. This decrease in measuring time should be inverse proportional to the step size. Thus doubling the step size would approximately half the measuring time. However, most importantly, different artifacts show up in the images when the constraint is not fulfilled. This effect is shown in figure 3.16c, where a ghost target is visible on the left hand side of the SAR image. The noise is also significantly more noticeable.

The chosen step size has from the start been 0.02 GHz. This was to account for the internal optical path length of the system, and to avoid artifacts. This can be increased to 0.1 GHz, when other factors, such as bandwidth and integration angle, are required to be large. However, this can be unreliable as the noise from the internal optical path would be superimposed on the rest of the signal, possibly causing artifacts within the SAR image.

3.5.4 Angular Separation Constraint

The theoretical angular separation for a given spot size is presented in 1.4. This gives a decent knowledge of the amount of scans necessary for a given integration angle. However, the effect of this constraint may not be entirely transparent in the final SAR image, as the general method is to calculate the angular separation constraint for the beam diameter and then round down to a slightly lower separation.

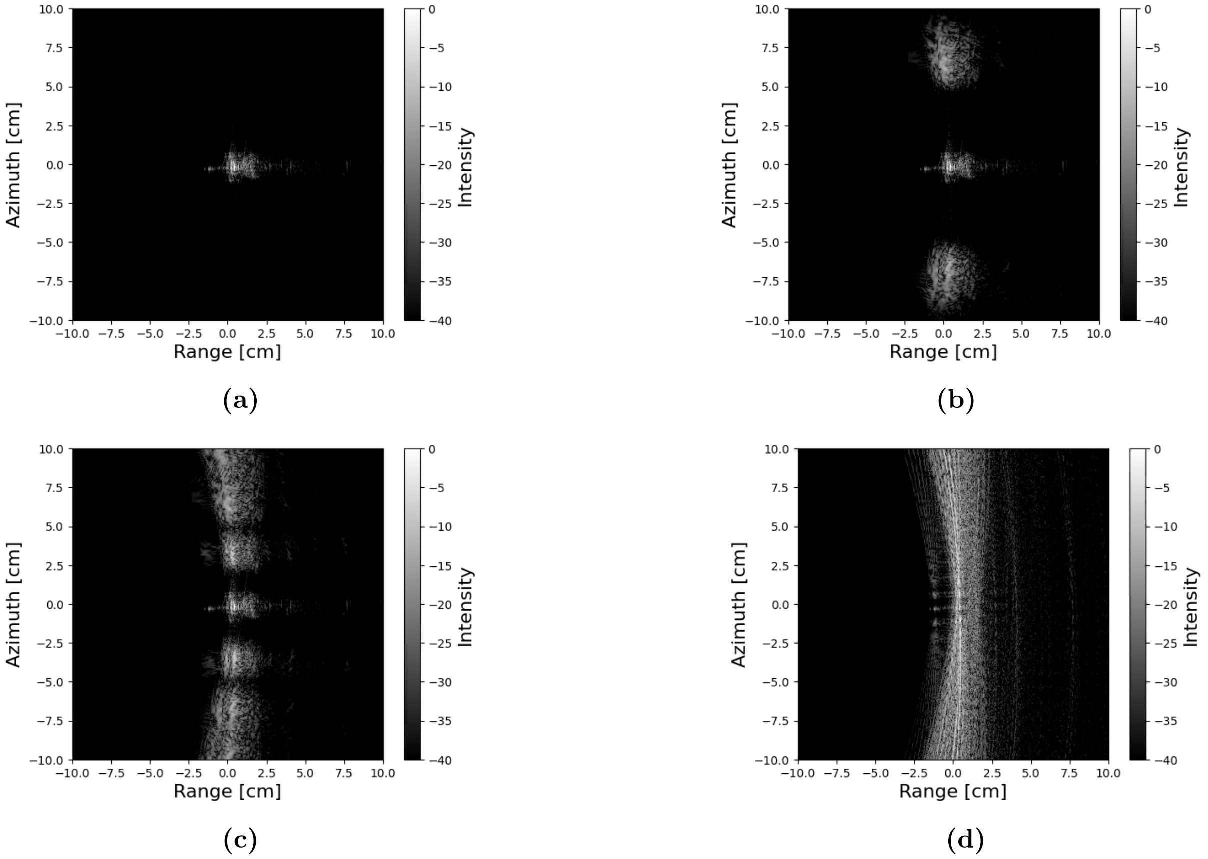


Figure 3.17. *SAR image of a T72 with angular separations of **(a)** 0.1°,**(b)** 0.2°, **(c)** 0.8°, and **(d)** 1.6°.*

The effect of too high angular separation is shown in figure 3.17. As the angular separation is increased we see noise emerging in the azimuthal direction. This, however, does not interfere with the T72 before an angular separation of ≈1°. When the angular separation becomes too high as in figure 3.17d, the image starts to appear as if no azimuth compression has been performed.

3.6 Fabrication of Models for THz SAR

This section will outline the workflow and considerations of the fabrication process of the THz scale models, test targets and backgrounds. In the THz-region it is important to remember that many optical and radar materials will behave differently. For any scale measurements in the THz domain, made to simulate radar signals with scales of either 100:1 or 200:1, it is therefore important to use materials that behave similarly at both frequencies. When this is impractical, it can be necessary to find other materials in the THz-domain that scatter THz waves like the original material would scatter radar waves.

In this report it was found that a lot of materials in the THz-domain could be made to scatter THz waves like radar waves by coating the material with metal. The most used method was gold coating of the models by physical vapor deposition. However, tests were also performed with copper spray paint. The copper particles were listed as being under 100 µm and having a mass aerodynamic diameter of 28 µm, i.e. 50% are larger and 50% are smaller[68]. It is important that the particle diameters be smaller than the wavelength of the THz radiation, to avoid dominant particle scattering.

The general workflow for fabrication THz models involve modeling the target in the 3D modeling software blender[69] or finding free available models online. Then 3D printing the models with either extrusion printing or resin printing depending on the desired quality of the print. Finally the 3D prints are coated in a thin layer of gold. Generally there might be some iterations between printing and modeling to get a good print. The workflow is shown in figure 3.18.

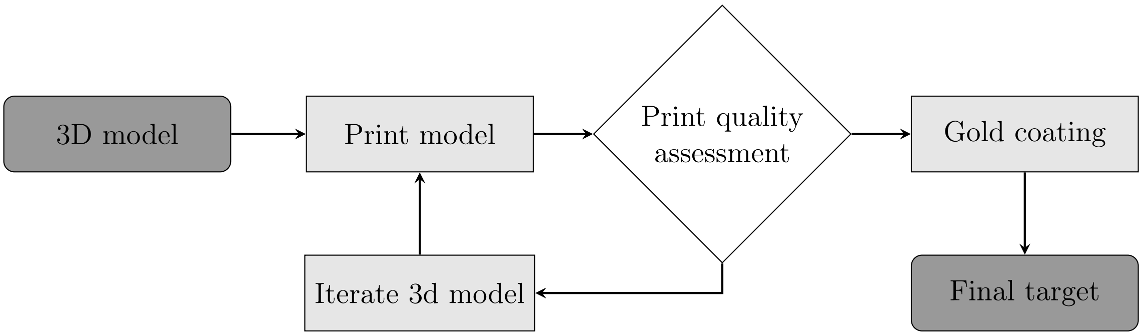


Figure 3.18. *Flow-chart of fabrication process.*

The specific 3D printers and gold coater used in this project are listed in table 3.1, detailing every machine used in the fabrication flow. Extrusion prints were done in PLA, and resin prints were done with the resin formula accompanying the specific resin printer.

machine	name
Extrusion printer	Ultimaker 3
Resin printer	ELEGOO Satrun 3 Ultra
Gold coater	Edwards Sputter Coater S150B

Table 3.1. *Name of equipment used in the fabrication process.*

3.7 Digital Surface Map Generation

This section briefly explains the steps necessary for DSM generation and the actual implementation in the SAR pipeline.

Equation 2.36 presented in section 2.2 is in itself not enough to create a height map of the SAR image. It requires many prerequisites and post processing for it to be applicable.

As previously mentioned, two different SAR images of the same scene are necessary. We also need the coherence of the images to be high for the InSAR equation to be valid. This can require image coregistration and alignment, such that features within the SAR images line up. It can be necessary to interpolate between points in the image, as InSAR can require alignment to the tenth of a pixel. With aligned images, a usable interferogram can be obtained. The interferogram is a map of the phase difference between the two SAR images. As all cells of the SAR image have a complex value, the phase difference can be calculated by the complex product of the individual pixels

$$\psi_{i,j} = \arg(a_{i,j} b^*_{i,j}), \tag{3.15}$$

where a and b are matrices representing SAR image 1 and 2 respectively, and ψ is the resulting matrix representing the interferogram. Most of the time, filtering is necessary to reduce noise, this is done by averaging the phase compared to the nearest neighbors. Additionally, masks can be applied that removes the pixels of low quality, usually measured by coherence. If the quality of the interferogram is high enough, it can be unwrapped, this follows methods presented in 2.2.2.[11]

With the unwrapped phase it is possible to determine the height using equation 2.36. However, this result is usually not the correct height nor the correct placement of the pixels due to iso-range projection. To get an accurate DEM, orthorectification is needed. The applied method of orthorectification is an iterative process, where a fraction of the calculated height will be included in the grid as new z-values for a new SAR image, which in turn will be used to calculate a new interferogram. This process will be repeated until convergence is reached. [11]

THz InSAR Pipeline

As presented in section 3.3, a stepper motor has been implemented to precisely modify the slant angle of the ISAR setup. This allows to precisely create an ISAR image with small baselines (∼5 mm) without any external influence on the setup, allowing for great precision in measurements. We use the skimage phase unwrapping function in python to acquire the unwrapped phase. The skimage unwrapper uses non-continuous path-following method producing fair quality unwrapped images[51]. The rest of the THz InSAR pipeline follows the method presented in the previous section, and is illustrated in the flowchart in figure 3.19.

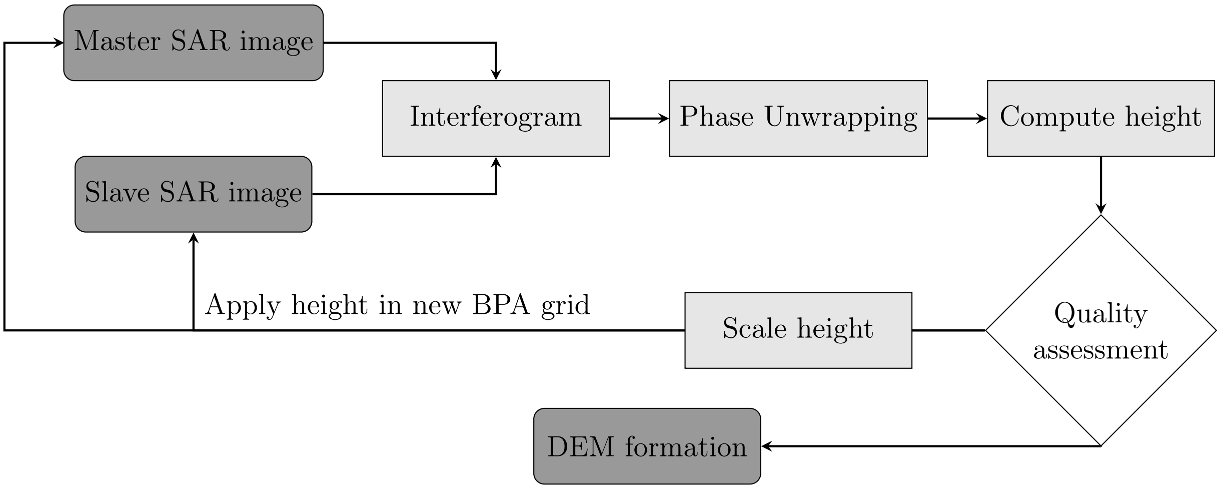


Figure 3.19. *Flow-chart of InSAR processing.*

3.8 Convolution Neural Network

To test the accuracy of the simulated SAR images against the MSTAR dataset a simple convolution neural network (CNN) is used for object classification. The CNN is trained on simulations and validated on MSTAR, and is then tested on MSTAR. It is beyond the scope of this project to dive fully into the CNN, and it is therefor treated as a black box. The CNN is from a MATLAB example on MSTAR image classification.[70]

SAR Microscopy 4

4.1 Microscopy Results

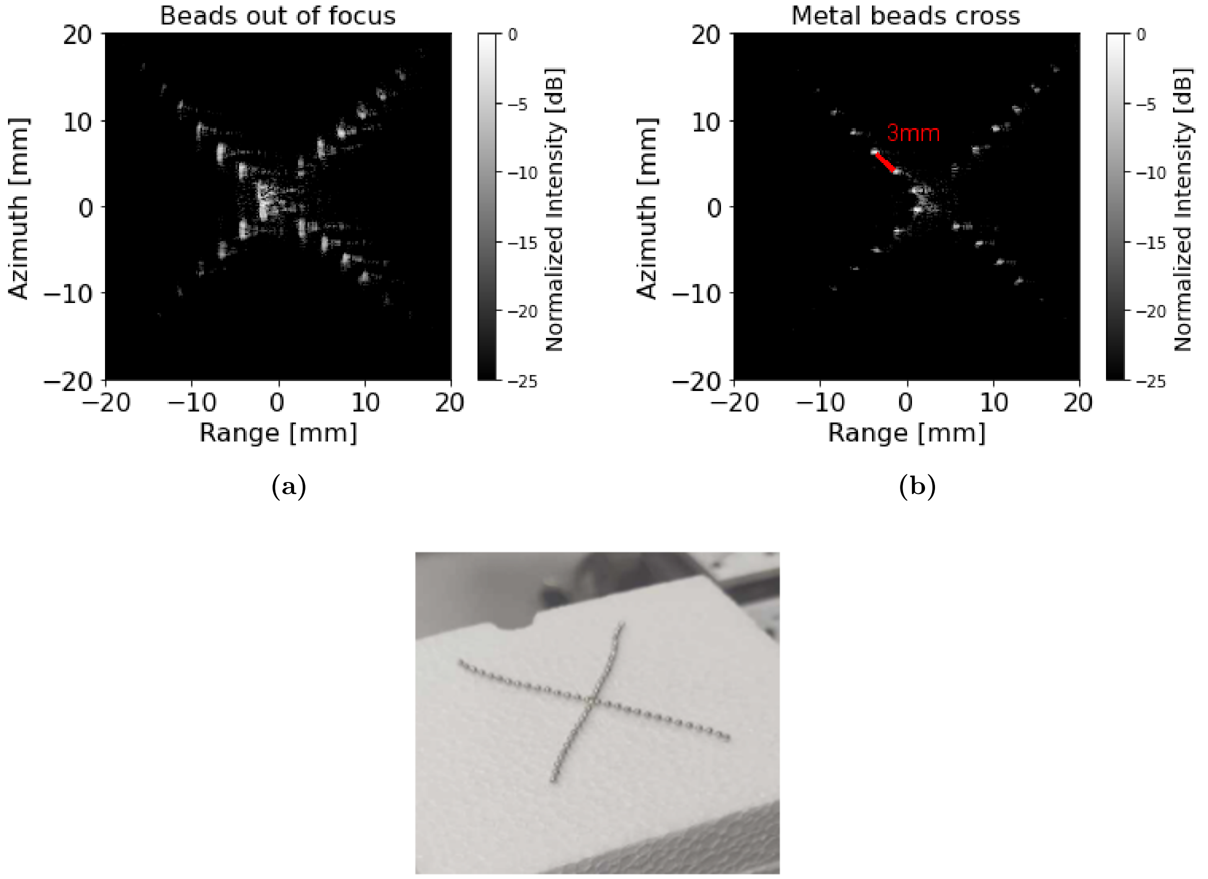
This section will contain experimental results related to optimizing the resolution of THz SAR images. The methods used for treating the data are presented in 3.4. All SAR images in this section have been generated by measurements from 0.5 to 1.5 THz, yielding a theoretical range resolution of 150 µm.

4.1.1 Resolution Testing

In the process of examining the resolution of our THz SAR setup, the first step was to obtain good image qualities when measuring objects with features on a cm to mm scale. For this, two strings of metal beads connected on wire were used as a target which can be seen in figure 4.1c. The beads have a diameter of 2.3 mm and are spaced 3 mm apart from center to center.

A requirement for good image quality when using the BPA is precise knowledge of the location of emitter and detector relative to the scene. Measuring these positions with mm or sub mm precision when they are located 80 cm away from the scene in a bi-static setup is a difficult task. Hence the auto focus technique described in section 2.1.3 comes in handy to iteratively optimize the precise emitter and detector position. In figure 4.1a the distance to emitter and detector is shifted with 3 mm in the range direction relative to the image after auto focus seen in figure 4.1b. The image quality after auto focus is significantly enhanced, and the beads appear more like points after auto focus, whereas before they were elongated in the azimuth direction.

Importantly, the distance between the dots in figure 4.1b matches the 3 mm distance between the beads. Furthermore, the beads furthest from the center of the cross are not visible due to the limited spot size of the THz beam. The distances on the axes in the presented SAR images are relative to the rotation center of the scene.



(a)

(b)

(c)

Figure 4.1. *(**a**) SAR image of metal beads with errors in position. (**b**) Metal beads after auto focus. (**c**) Optical image of measured beads.*

4.1.2 Microscopy Target

Several iterations of microscopy targets have been modeled and 3D printed. The challenge lies in printing small features of a couple hundred microns that are stable and print well. Ultimately targets consisting of a line of evenly spaced rods were used. The center of rods with a diameter d, are separated by a distance $2d$. As the diameter is decreased so is the gap between adjacent rods. The full model is shown in figure 4.2, for a diameter of $d = 400\,\mu\text{m}$. The model is made to be measured at a low slant angle. The base is therefore sloped to reflect incoming radiation away, and avoid imaging the base.

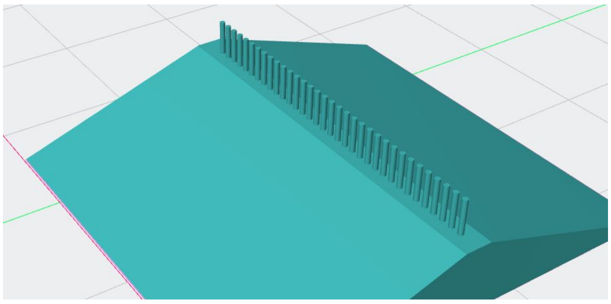
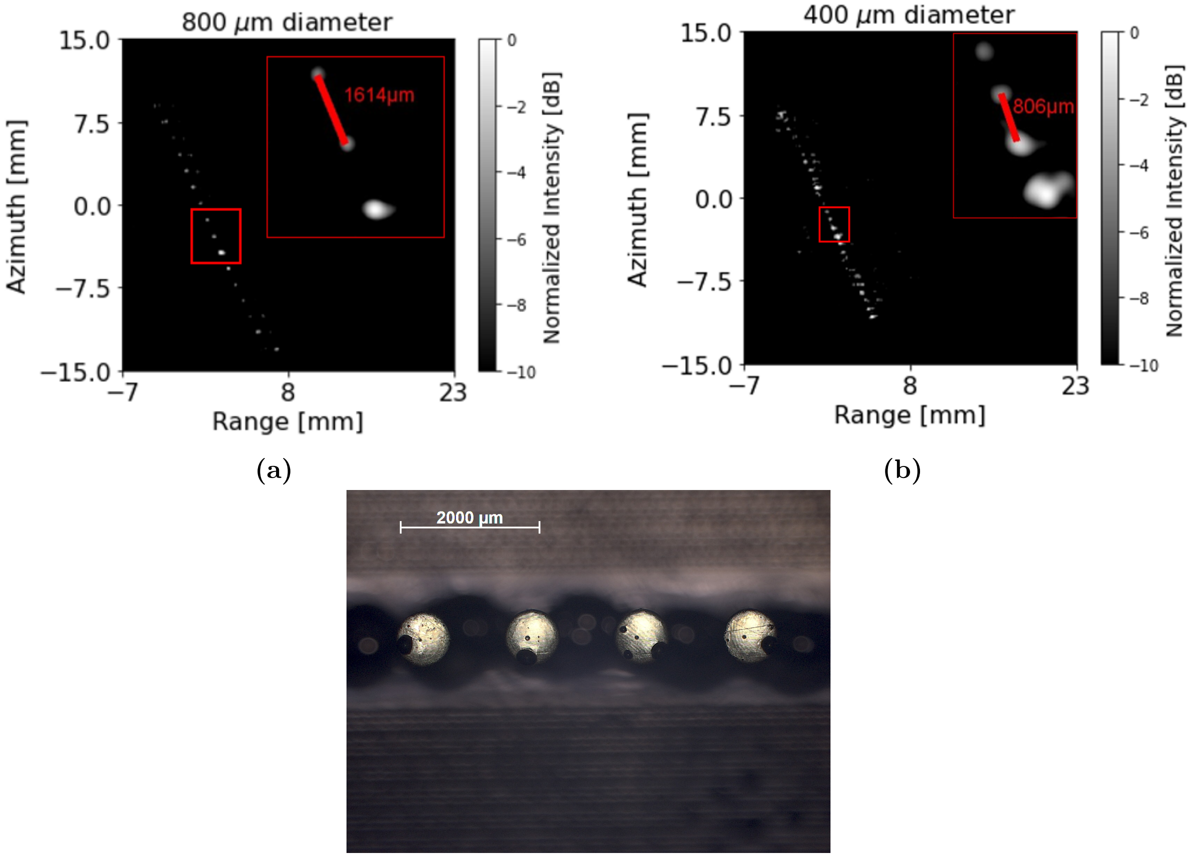


Figure 4.2. *3D model of the microscopy target. The rod diameter is* $400\,\mu\text{m}$.

The resulting THz SAR microscopy images of microscopy targets of rods with diameters of 800 µm and 400 µm are shown in figure 4.3. With cylinders, the part of the object that is imaged in SAR, is the small part of the cylinder pointing towards the detector. In both SAR images a full row of rods are visible by this reflective area. The distance measured in the THz SAR images between point reflections, match the distance $2d$ between rods. The resulting test target shown in figure 4.3c, was resin 3D printed and coated in gold.



(a)

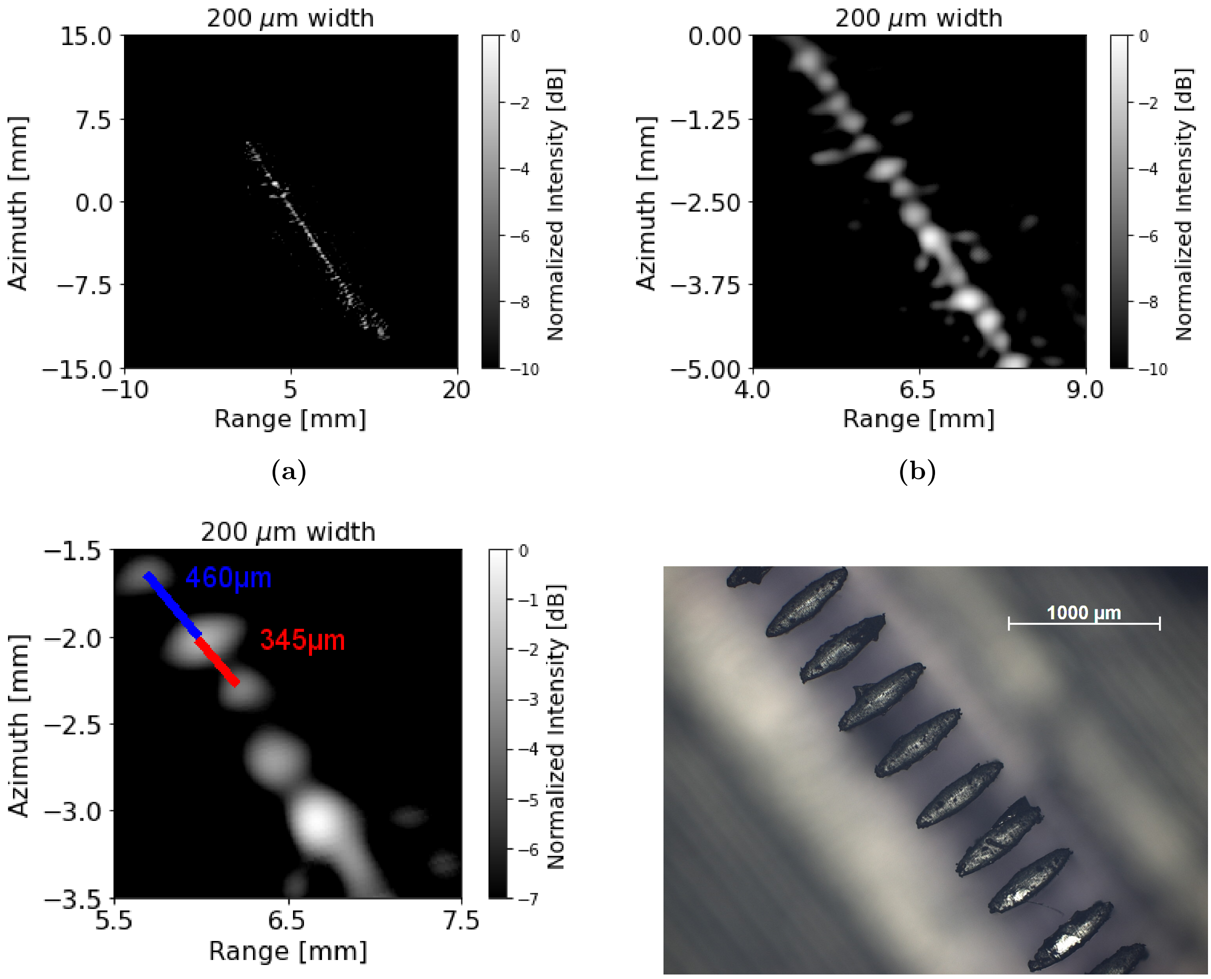
(b)

(c)

Figure 4.3. *SAR image of microscopy target of rods with* ***(a)*** *a diameter of* 800 µm *and* ***(b)*** *a diameter of* 400 µm*.* ***(c)*** *Top down light microscopy image of the microscopy target in* ***(a)****.*

The target design allows for precise printing of small features of at least 400 µm, while retaining its structural integrity and clear separation between rods. For smaller rods, the targets were elongated along the look direction for more structural support. In these cases the width is given, which is the minor axis of the elliptical rod.

The results of SAR microscopy of a microscopy target with a rod width of 200 µm is shown in figure 4.4. In the resulting SAR images individual rods can be distinguished. With a center frequency of 1 THz, the classical diffraction limit is 150 µm. In the resulting SAR microscopy images the resolution is shown to be comparable to the diffraction limit, with a working distance of 80 cm. In the 3D model the rods are separated by 400 µm, however as can be seen in both the SAR images and light microscopy image, the printed rods are slightly bent. The bending is caused by the 3D printing of such small features or during gold coating.



(a)

(b)

(c)

(d)

Figure 4.4. *SAR image of* 200 µm *wide rod microscopy target, at* ***(a)*** 3 × 3 cm, *at* ***(b)*** 5 × 5 mm *and at* ***(c)*** 2 × 2 mm. ***(d)*** *Top down light microscopy image of the microscopy target.*

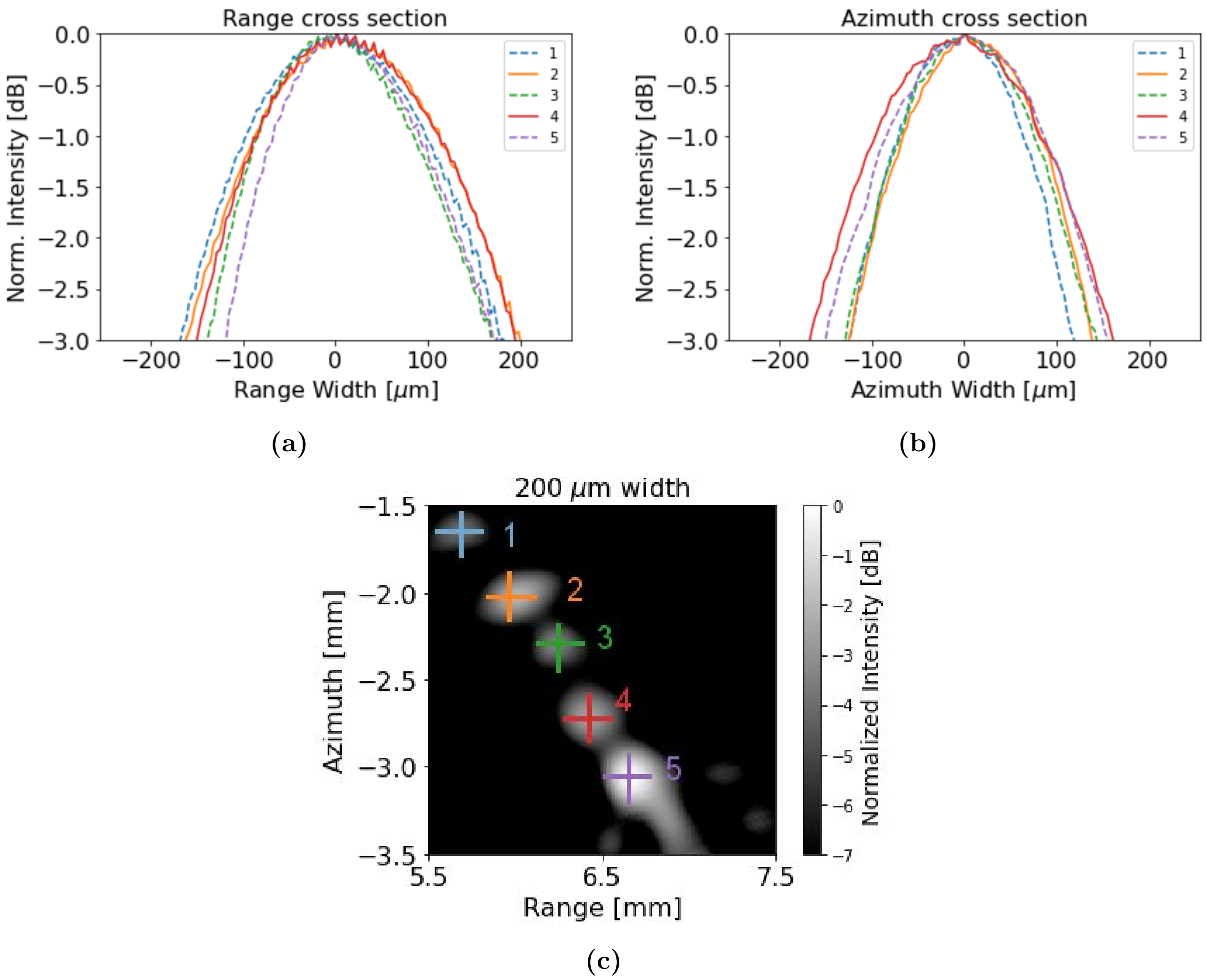


Figure 4.5. *(**a**) Range cross section across normalized SAR image peaks. (**b**) Azimuth cross section across normalized SAR image peaks. (**c**) The different peaks can be seen marked on the image.*

With a bandwidth of 1 THz an ideal range resolution would be a −3 dB pulse width of 150 µm from a perfect point reflector. However, as can be seen in figure 4.5a the −3 dB width in range is between 300 and 380 µm in the image. This difference from the theoretical range resolution comes partly from the application of range filter which widens the main lobes and that the measured rods are not perfect point reflectors. With an integration angle of 40° the theoretical azimuth resolution at a center frequency of 1 THz will be 215 µm. The −3 dB width in azimuth, which can be seen in figure 4.5b, is between 240 µm and 330 µm. The difference can again be attributed to the fact that the rods are not perfect point scatterers.

4.2 SAR Microscopy Discussion

In this section we have shown a THz SAR microscopy method for reflection microscopy, with a resolution capable of reaching the diffraction limit. Prior to the improvements presented in this project the maximum achievable resolution was around 1.3 mm [66]. However, with the addition of autofocus, filtering, and the increase in bandwidth and integration angle, the resolution has improved to <400 µm.

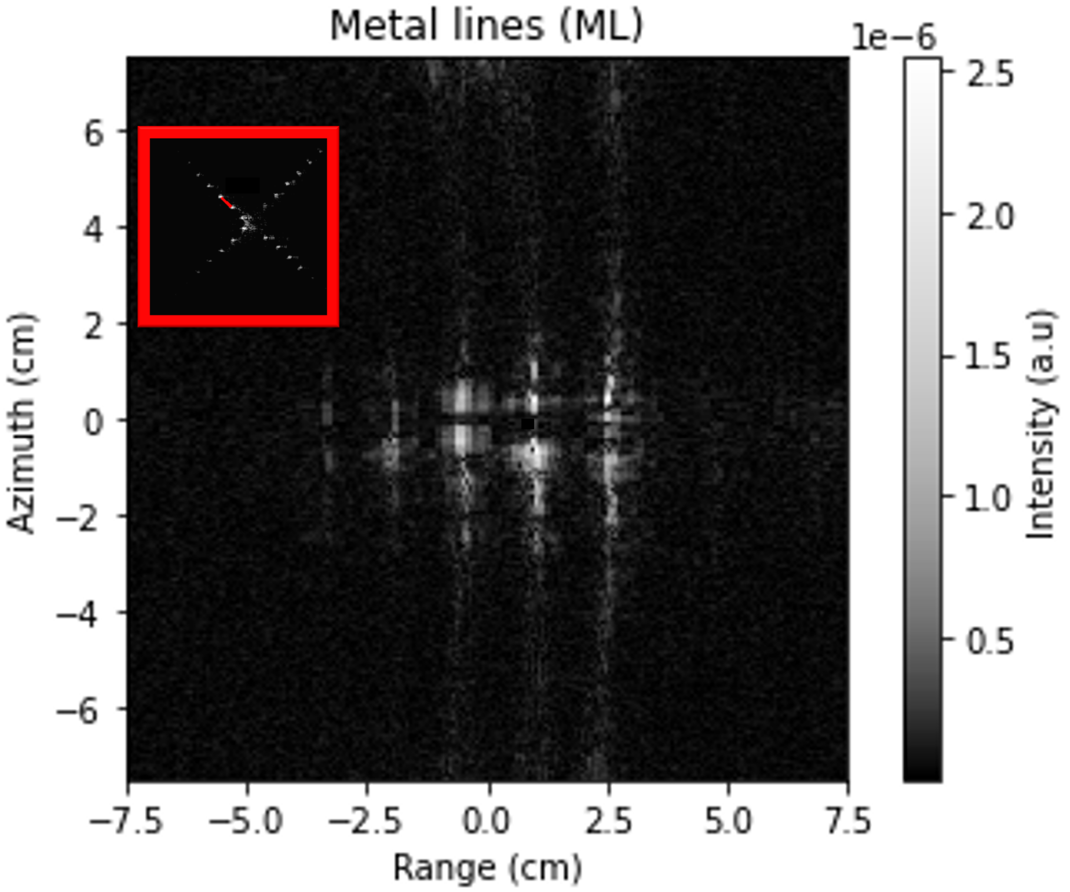


Figure 4.6. *SAR image of metal lines from Vang, Thomsen and Dyhre [66], with the SAR image of metal beads superimposed in correct scale.*

Figure 4.6 illustrates this improvement. Here, the a SAR image of the metal beads is superimposed onto the SAR image of the metal lines taken from previous work [66]. The difference can be seen as both the ability to image smaller features and lack of noise.

In general there are three factors to consider when evaluating a SAR method: resolution, standoff range, and measuring time. The achieved resolution is comparable to current work on THz SAR. In the works by Batra et al. [21] and Damyanov et al. [20], resolutions of respectively $\sim$ 700 µm and $\sim$ 100 µm were achieved with standoff ranges of 39 cm and 15 cm respectively. The longer standoff range was achieved with a frequency domain setup, and the shorter one with a time domain setup. What we generally see, is for lager standoff ranges it is advantageous to measure in the frequency domain, where a higher dynamic range can be achieved. By collimating the emitted THz radiation we are able to focus the beam onto a smaller area with higher intensity, which allows the larger standoff range of 80 cm. This is in contrast to many THz SAR applications using horn antennas producing slowly diverging beams.

The unique advantage of synthetic aperture, is the ability to image an entire object without the need of a raster scan. The raster scan is one of the most inhibiting factors in THz imaging systems. This is due to the long scan time caused by mechanical delay lines and the single pixel detectors used in the field[71]. Furthermore, regular THz-TDS and -FDS imaging methods are often also limited by the numerical aperture of the system of which they operate. Watanabe and Shimano [72] presents a THz-TDS system with an effective numerical aperture of 0.3-0.45, which will give a resolution of ≈1 mm. This is a transmission method and uses short working distances, which limits the use of the method. Redo-Sanchez et al. [73] uses a THz CW system, to raster scan for defects in sprayed-on foam insulation. They illustrate the low attenuation of THz radiation in certain materials. They used the operating frequencies of 0.2 THz and 0.38 THz yielding spot sizes, or resolution, of approximately 2 mm. They also claimed a spot size of 0.5 mm at 1.63 THz. THz SAR can thus achieve higher resolutions than its regular aperture counterparts, as it is not limited by the size of the achievable spot in the same way.

Like the raster scan, the SAR method can be time consuming. Due to the low output intensity and the fact that small targets reflect very little light, higher integration times are needed. Based on the reflections made in section 3.5, a general relation between THz SAR parameters and the total measuring time can be made

$$T_{tot} = \left\lceil \frac{\theta_{int} \cdot B}{\delta\theta \cdot df} \right\rceil \cdot t_{int} + T_{misc}. \tag{4.1}$$

Here, $\lceil x \rceil$ is a ceil function rounding up to the nearest integer, t_{int} is the integration time, and T_{misc} the time spent on mechanical motion for the system and resetting the scan frequency in between scans. Thus, increasing the integration angle or bandwidth to increase the resolution would increase the total time by the same factor. Increasing the integration time to get a better signal to noise of the raw signal, would also increase the total time by the same factor. Furthermore, decreasing the angle separation to allow for larger scenes, or decreasing the frequency step size to allow for a larger unambiguous range, would increase the total time as well. Therefore, it is important to consider each variable and its necessity to the actual purpose of the measurement. In microscopy, maximizing the parameters governing the resolution is of most importance, i.e. θ_{int} and B. To avoid long imaging times, this will come at a cost of the angular separation, which in turn restricts the scene size to a few centimeters. The frequency step size could also have been modified. However, to ensure the unambiguous range was large enough to accommodate the large standoff, $R = 80$ cm, without causing any artifacts, it was kept at 0.02 GHz. Thus, in total, the time for one measurement, where the resolution parameters were maximized, was approximately 20 hours. This can be shortened by increasing the frequency step size, risking artifacts in the SAR image and reducing the SNR in the SAR image, or decreasing the integration time reducing the SNR of the raw signal and thus the SAR image.

The move into the THz regime is seen in a few different SAR applications. The higher resolution increases the ability to resolve smaller items, such as knives, guns etc. while still being able to penetrate most non-metallic and non-polar materials. The theoretical work of Gui et al. [74] demonstrates the use of THz frequencies in detection of guns on moving targets using SAR imaging. The high resolution can also be utilized in non-destructive testing. Here Nezadal, Schür and Schmidt [75] investigated glass fiber reinforced plastics with THz SAR. They found it effective in detection of artificial defects.

4.3 Summary

In this section we showcase a THz SAR setup with a spatial resolution comparable to the diffraction limit, operating with a center frequency of 1 THz and a bandwidth of 1 THz. The setup measures reflections, with a working distance of 80 cm through air. The setup is based on a commercial THz-FDS system, making it suitable for applications in non destructive testing and characterization in industry.

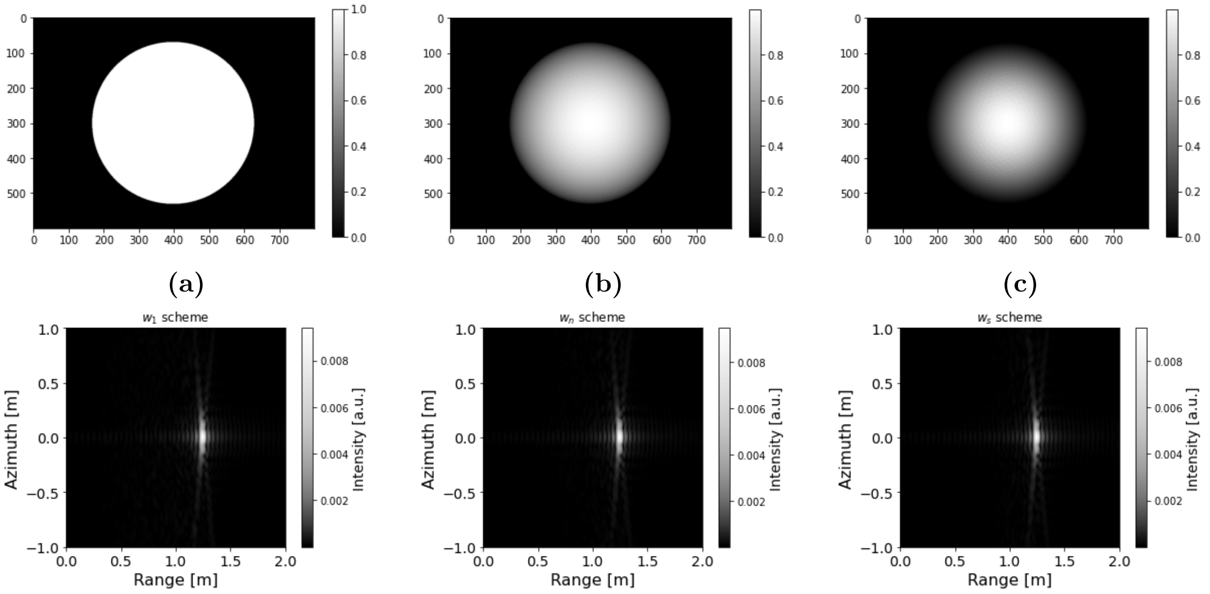
Computer Simulations 5

5.1 Rasterization

In this section rasterization is compared with ray-tracing. Initially, tests with the different shading schemes for the rasterization method have been carried out. Three schemes were tested, with weights calculated by

$$w_1 = 1, \qquad w_n = -\hat{\mathbf{n}} \cdot \hat{\mathbf{v}}, \qquad w_s = w_n^2. \tag{5.1}$$

From left to right, each weighting scheme gets more sensitive to the orientation of the target surfaces. The intensity of the backscattered ray from the surfaces of a sphere are shown in figure 5.1 for each scheme. The effects of each scheme on the measured intensity for differently orientated surfaces. Note how the schemes w_n and w_s are more specular, which is generally expected for metal surfaces.



(a) (b) (c)

(d) (e) (f)

Figure 5.1. *Sphere with its surface shaded according to the weights of the backscattered rays and the resulting SAR image for the (**a**, **d**) w_1 scheme, (**b**, **e**) w_n scheme and (**b**, **f**) w_s schemes.*

The resulting SAR images of a sphere calculated with rasterization for the different schemes are very similar. This is explained by the change in angle between rays and the sphere towards the poles. If we consider a sphere that is irradiated by parallel rays of radiation, then rays closer to the poles meet the sphere at smaller angles and therefor spreads the intensity over a larger area.

To test the validity of the rasterization method, it was tested against the ray tracer. The scene is a metal ball, and the resulting SAR images can be seen in figure 5.2.

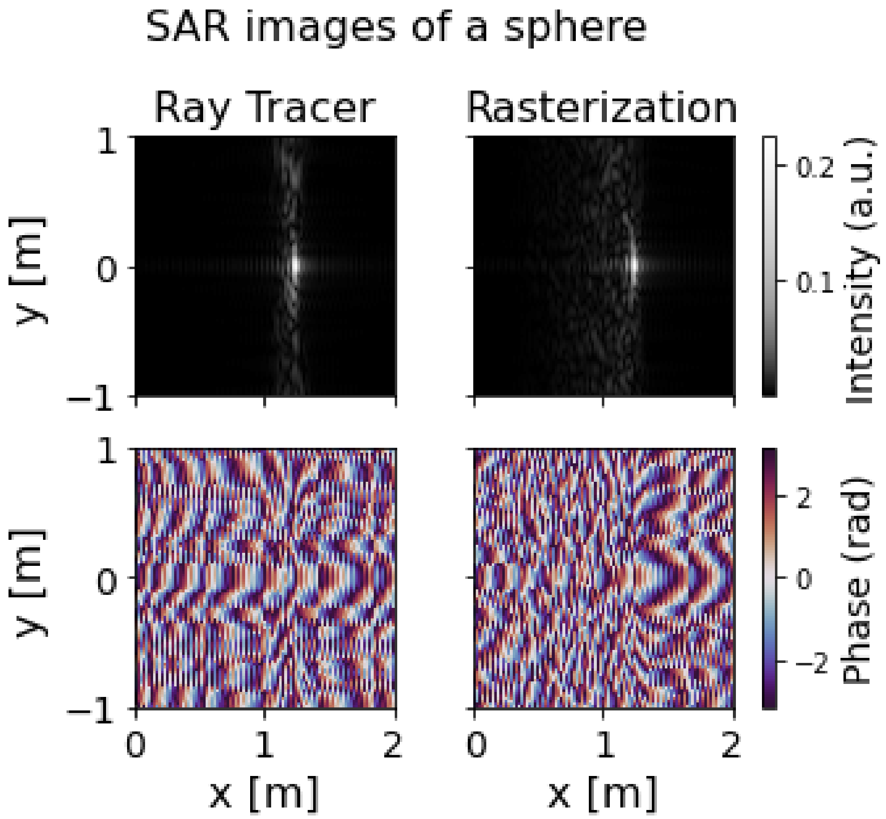


Figure 5.2. *SAR image and phase of metal sphere by ray tracing and rasterization.*

For simple scenes with minimal bounces ray tracing and rasterization can be seen to create similar radar data. In both cases the characteristic intensity hotspot is observed, corresponding to the part of the sphere perpendicular with the detector. With rasterization more of the sphere's curvature can be seen, as rays that are normally reflected away from the detector in a ray tracer are detected. The phases are generally comparable, however since the rasterization captures radiation from a larger area on the sphere, the return signal is also wider in time. This is what is responsible for the change in phase between the two methods. In figure 5.3 the difference between the SAR images and phases are calculated by rescaling the intensity of the SAR images with respect to peak amplitude, and subtracting the rasterization results from the ray tracer results.

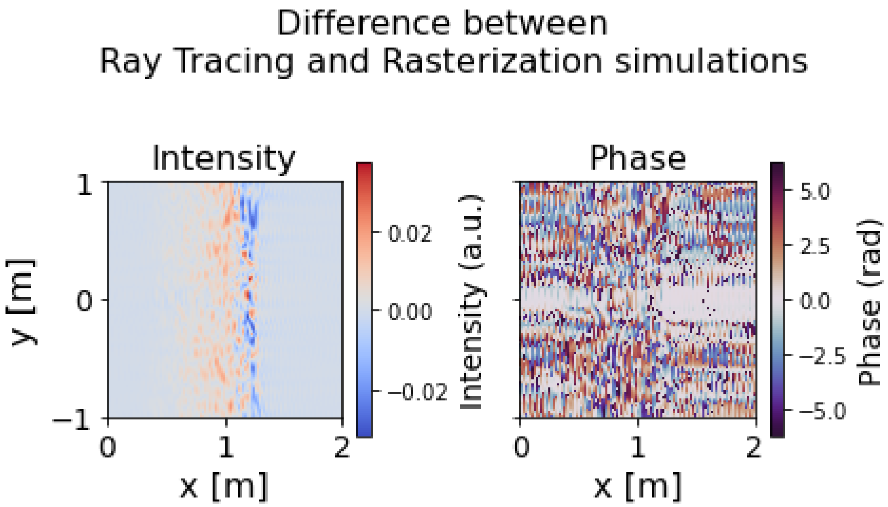
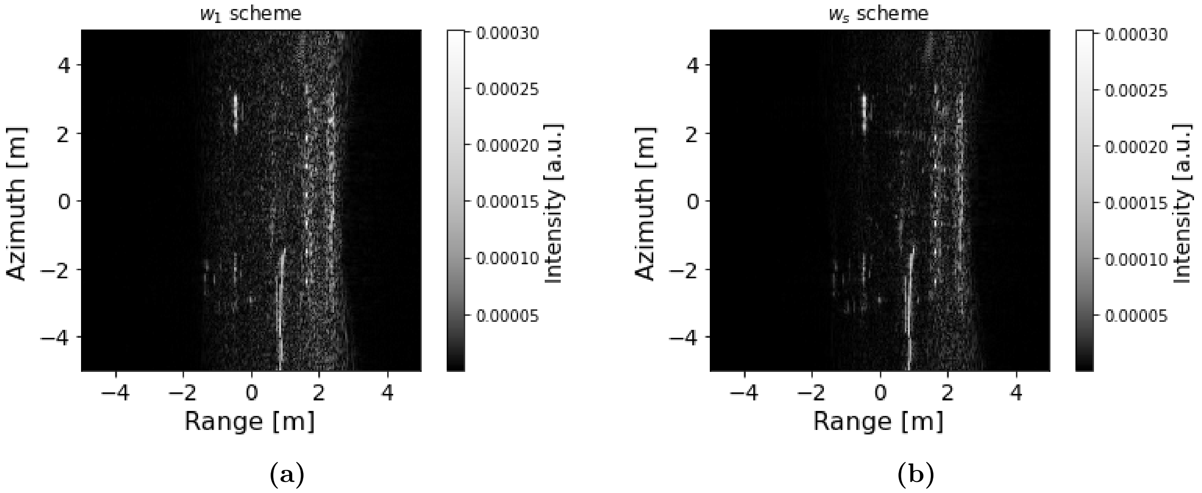


Figure 5.3. *Plots of the differences between ray-tracing and rasterization for* **(a)** *Intensity and* **(b)** *phase.*

In figure 5.3 the blue areas are areas in the SAR image where rasterization has higher intensity than ray tracing, and visa versa for red. When we subtract the phase, we find that the two methods agree best along the center line at $y = 0$.

To finish of with rasterization, a SAR image of a t90a tank has been simulated with the schemes w_1 and w_s, see figure 5.4. The SAR images have been simulated with 120 azimuth steps with 1M rays pr. step, a remarkably low number of rays as compared with a ray tracer, see section 5.4.



(a)

(b)

Figure 5.4. *SAR images of model t90a tank rasterization simulations with 1M rays pr. azimuth for* (**a**) w_1 *scheme and (***b**) w_s *scheme.*

For more complex models, the choice of weighting scheme can improve the quality of the final SAR image. With w_s the SAR image is more defined with less azimuth noise. However both schemes ultimately highlight the same areas of the model, i.e. faces perpendicular to the detector. In both images the barrel and treads are the most prominent features of the tank.

5.2 CPU Ray-Tracing

In this project a CPU ray-tracer has been coded from the ground up in Python using a BVH and mesh optimization, which is outlined in section 2.4. The CPU ray-tracer enables us to quickly implement new features, at the expense of slow computing when compared to GPU ray-tracing.

5.2.1 Optimization

In the previous project, a simple ray-tracer written in Python based on the work by Peter Shirley [76], was used for small scenes. However, with the increasing scope of the project the ray-tracer needed to be optimized. To this effect, a BVH was implemented with mesh optimization. Additional performance from python was also possible by utilizing Numba, a just in time compiler for translating python functions into fast machine code. Numba only works on a limited set of operations in python, mainly those in the Numpy package for scientific computing. However, with the rewrite, numba is able to speed up the code massively. In figure 5.5, the runtime of different ray-tracers of a large model (above 300000 triangles) are shown. With each ray-tracer additional optimization is added, with Numba, BVH and mesh optimization, see table 5.1.

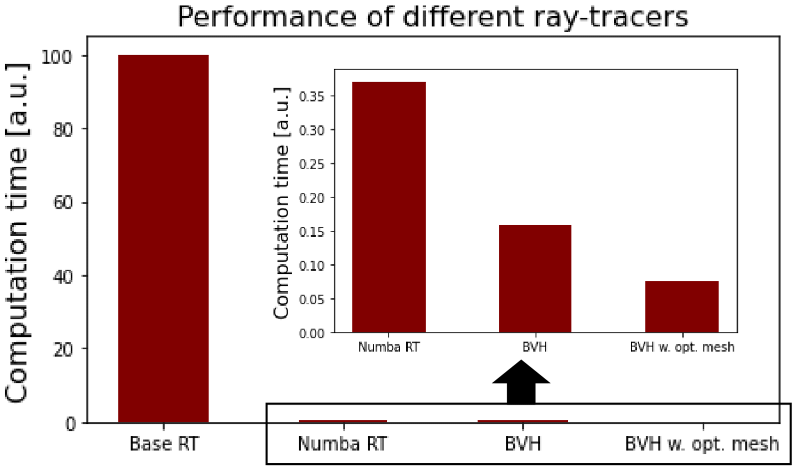


Figure 5.5. *Relative performance of different ray-tracers.*

RT name	Numba	BVH	Mesh optimization
Old RT			
Numba RT	x		
BVH	x	x	
BVH w. opt. mesh	x	x	x

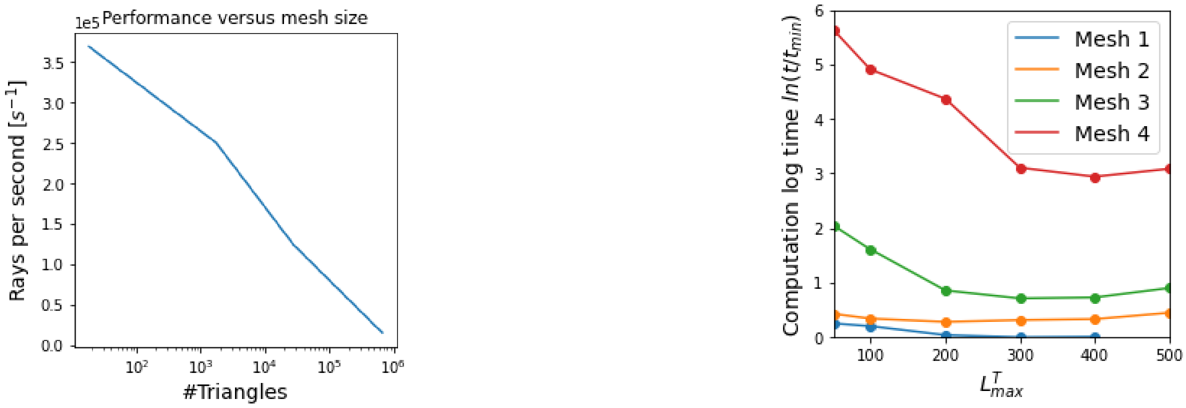
Table 5.1. *Overview of the different performance techniques applied to each ray-tracer. x marks the use of a technique.*

The biggest performance boost by far is the rewrite with Numba, a $\sim 270\times$ increase. Along with the fast functions, this step also includes CPU parallelization. The BVH is another $\sim 2.3\times$ increase, with better scaling for larger meshes. Optimizing the mesh with the method outlined in section 2.4.1, is another $\sim 2.1\times$ increase. The final performance boost from "Old RT" to "BVH w. opt. mesh" is a $\sim 1300\times$ speed increase. Even with this huge improvement, the CPU ray-tracer is limited by the number of CPU cores. Ray-tracing is a so called "embarrassingly parallel" process and therefore huge speed increases can be found with GPU implementations, as with the Optix ray-tracer. However the CPU ray-tracer allows for easy testing of new implementation before diving into CUDA. The performance for meshes with increasing number of triangles are shown in figure 5.6a. The number of triangles in each mesh, mesh 1-4, is given in table 5.2.

Name	Mesh 1	Mesh 2	Mesh 3	Mesh 4
Number of triangles	18	1728	27648	663552

Table 5.2. *Table with the number of triangles in each mesh used for performance testing.*

For each mesh the depth of the BVH has been optimized, by setting the maximum number of triangles in all leaf nodes, L^T_{max}. The resulting sweep of L^T_{max} for a single azimuth step is shown in figure 5.6b. For each mesh the value of L^T_{max} resulting in the fastest computation time was chosen. The resulting graph of rays pr. second as a function of the total umber of triangles is shown to be linear on the logarithmic axis.



(a)

(b)

Figure 5.6. *(a) Performance versus number of triangles in mesh. (b) Initial test results for finding optimized value of L^T_{max} for each mesh.*

5.2.2 BVH Characterization

During the creation of the CPU BVH, the mesh is split up into child nodes, based on the BVH splitting implementation. The following analysis is carried out on the 3D model of a t90a tank, model can be seen in figure 5.7. An example of the different child nodes of the BVH are shown in figure 5.8, with mesh optimization.

Figure 5.7. *t90a model*

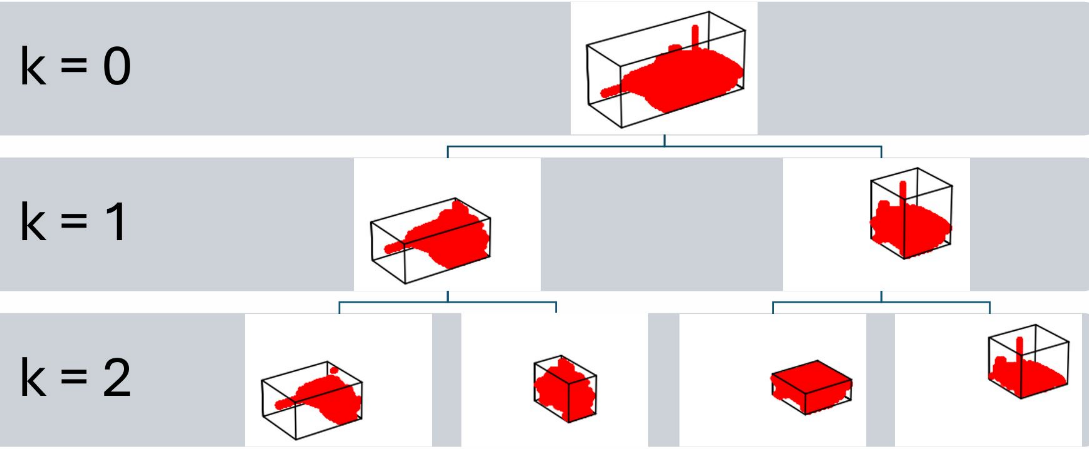


Figure 5.8. *BVH showcasing the first 3 layers. For each node the volumes (black) and mesh vertices (red) are shown.*

The figure immediately highlights the binary tree structure of the BVH. The BVH has nice tight bounding boxes around the node meshes. However, if we compare BVH nodes for the mesh with and without mesh optimization, it will become clear why mesh optimization has such a big impact on ray-tracing speed. In figure 5.9 every bounding volume at each depth k is plotted together for the original mesh and mesh optimized.

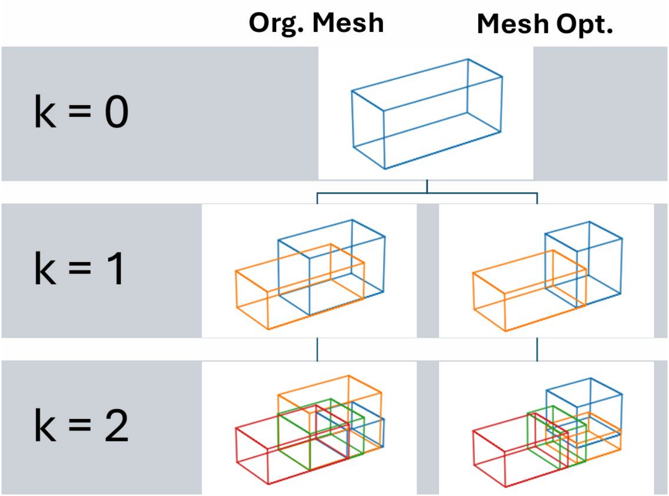


Figure 5.9. *Comparison of the bounding volumes created by the BVH for the original mesh and the optimized mesh. For each depth k all bounding volumes are plotted together.*

Note the difference in bounding volume overlap between the two BVH, particularly at mesh interfaces. These large overlaps are a result of large triangles in the mesh. For large triangles the centroid $\mathbf{C}$ can be far removed from its vertices $[\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3]$. These "dangling" vertices must be fully enclosed in its respective bounding volume, leading to the large overlap at the mesh interfaces. Generally lower BVH volume and overlap results in faster BVHs. By optimizing the mesh, the total volume and overlap of the BVH are thereby reduced.

The total volume and overlap of bounding volumes for each layer of the BVH with and without mesh optimization can be seen in figure 5.10.

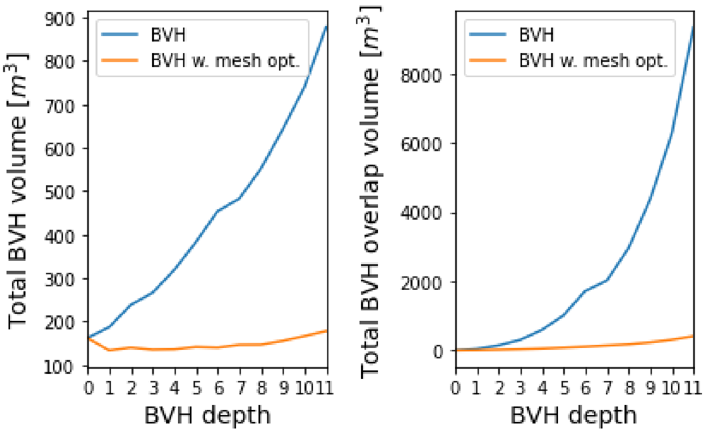


Figure 5.10. *Total BVH volume and overlap for each layer in the BVH tree of "BVH" and "BVH w. mesh opt.". The largest allowed centroid-vertex distance is* $D_{max} = 0.4$m.

Using mesh optimization hugely decreases the total BVH volume and overlap as a function of BVH depth. The total volume barely increases past the start volume with mesh optimization, and the overlap grows much slower. The result of lower total volume and overlap is evident when comparing the computation time. The "BVH w. mesh opt." was $\sim 2.1\times$ faster than the "BVH".

When doing mesh optimization by splitting large triangles into smaller ones, the total number of triangles obviously increases, which can have negatively effects on the computation time. In the above example the total number of triangles of the mesh went from 308444 to 337960, a 9.5% increase. Decreasing D_{max} further would result in an even larger increase. For a given mesh there is an optimal D_{max} where the upside related to the BVH construction is maximized compared with the downside of adding new triangles to the mesh. Similarly as outline in equation 2.53, there also exists a optimal BVH depth based on c_T and c_I the average cost of traversing a step and a ray-primitive intersection. In our case where every leaf is at the same depth, the BVH parameters depth k and centroid-vertex D_{max} can be optimized for a given mesh for the fastest computation time, see figure 5.11.

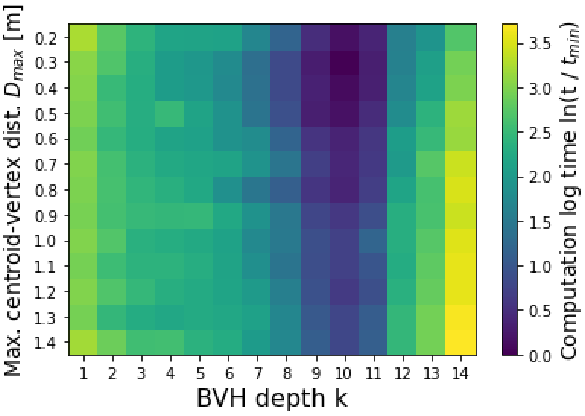


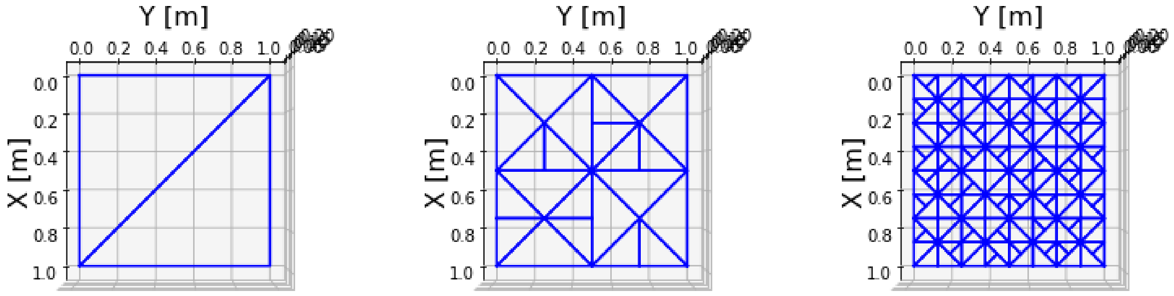
Figure 5.11. *Parameter sweep over BVH depth and maximum centroid-vertex distance showing the effect on computation time. The computation time is plotted logarithmically, normalized by the minimum computation time* t_{min}.

In the parameter sweep there is a strong minimum at BVH depth $k = 10$ for all D_{max} values.

A weaker dependence on D_{max} is also present, with computation time decreasing with smaller D_{max} before rising slightly again at $D_{max} = 0.2$. The optimal depth and centroid-vertex distance was found to be $k = 10$ and $D_{max} = 0.3m$.

Triangle Splitting

Given a $1m \times 1m$ plane consisting of 2 triangles, the resulting subdivision with the triangle splitting algorithm can be seen in figure 5.12, for $D_{max} = 0.3$m and $D_{max} = 0.1$m. The resulting meshes exhibit uniform distributions of triangles after mesh optimization, which is ideal, for this relatively simple splitting algorithm.



(a) **(b)** **(c)**

Figure 5.12. *Example of mesh optimization by triangle splitting for* ***(a)*** *original mesh,* ***(b)*** $D_{max} = 0.3m$ *and* ***(c)*** $D_{max} = 0.1m$.

5.3 Simulation Noise

With the aim of creating simulations which closely match real SAR measurements, different methods for adding noise to the simulation were investigated. Three different methods for different part of the simulation process were tested; in the ray-tracer, when calculating signals and in the SAR image. A totally ideal ray tracing simulation is given in figure 5.13, of the same model to which noise will be added.

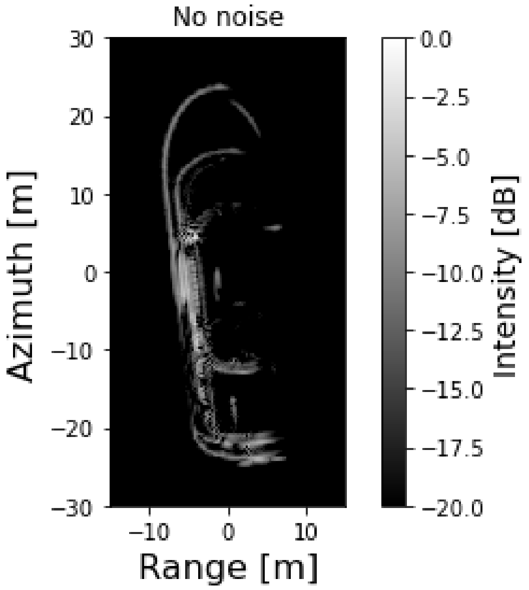


Figure 5.13. *Ideal SAR simulation of a boat.*

Ray-Tracer

We add noise to the ray-tracer simulation by giving each mesh triangle a reflectivity R from the normal distribution $\mathcal{N}(0, \sigma^2)$. When a ray, with an incoming ray intensity I_r^0, intersects a triangle with R, the outgoing ray intensity I_r^1 is given by

$$I_r^1 = RI_r^0. \tag{5.2}$$

With this method it is possible to mimic the signal gathered from rough surfaces and defects of real structures. The resulting SAR image is shown in figure 5.14.

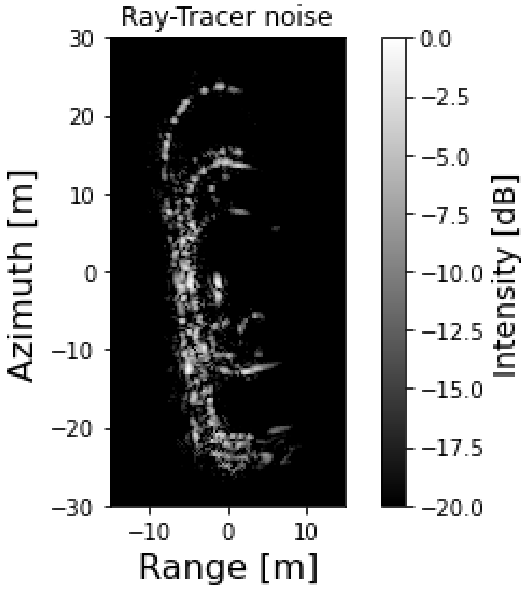


Figure 5.14. *Noisy SAR image, with ray-tracer noise.*

Signal

Adding noise to the signal is a three step process. Firstly, to emulate the small differences in path lengths from intersections with rough surfaces, the range histogram $R_H(r)$ is smoothed by a gaussian convolution. The smoothed range histogram $R_H^s(r)$ is then used to calculate the demodulated signal $\mathbf{S}_D(t)$ as usual. Secondly, multiplicative and additive noise is then applied to $\mathbf{S}_D(t)$ in the frequency domain. Both types of noise are generated from gaussian distributions. The multiplicative and additive noise are generated from

$$\mathcal{N}(1, \sigma_{mul}^2), \qquad \mathcal{N}(0, \sigma_{add}^2), \tag{5.3}$$

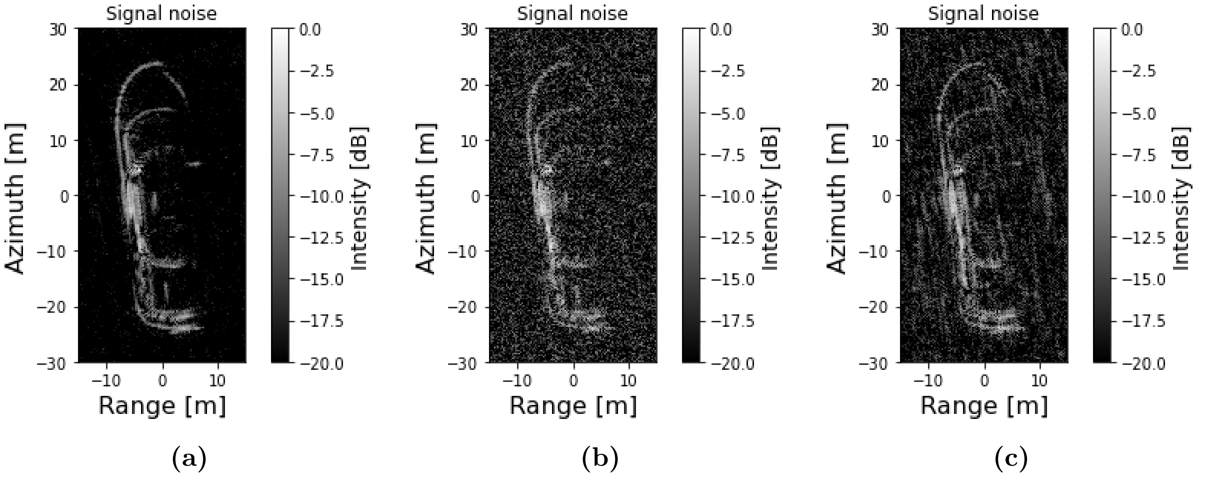
respectively. Thirdly, each azimuth signal is weighted by a random value from the gaussian distribution $\mathcal{N}(1, \sigma_{azi}^2)$, with small σ_{azi}. For a signal $\mathbf{S}_D(t)$ with N sample points in the frequency domain and M azimuth samples, the following 3 vectors are generated from the distributions above

$$\mathbf{v} = \begin{bmatrix} v_1^{mull} \\ \vdots \\ v_N^{mull} \end{bmatrix}, \qquad \mathbf{w} = \begin{bmatrix} w_1^{add} \\ \vdots \\ w_N^{add} \end{bmatrix}, \qquad \mathbf{u} = \begin{bmatrix} u_1^{azi} \\ \vdots \\ u_M^{azi} \end{bmatrix}. \tag{5.4}$$

For each azimuth step i the noise in the frequency domain is added by

$$\mathbf{S}_D^{noise}(f) = (\mathbf{S}_D(f) \odot \mathbf{v} + \mathbf{w} S_{abs}^{max})\, u_i^{azi}, \tag{5.5}$$

where S_{abs}^{max} is the maximum absolute value of $\mathbf{S}_D(f)$. Based on the range histogram of the ideal simulation, the resulting SAR images with signal noise are shown in figure 5.15.



(a) (b) (c)

Figure 5.15. *Noisy SAR images, with signal noise. Each figure has the following parameters* ***(a)*** *($\sigma_{add}^2 = 90$, $\sigma_{mul}^2 = 3$, $\sigma_{azi}^2 = 0.4$),* ***(b)*** *($\sigma_{add}^2 = 313$, $\sigma_{mul}^2 = 3$, $\sigma_{azi}^2 = 0.4$),* ***(c)*** *($\sigma_{add}^2 = 90$, $\sigma_{mul}^2 = 8$, $\sigma_{azi}^2 = 0.4$).*

Image

Speckle noise can also be added to the final SAR image by a simple multiplicate mask. This method does not allow us to gain any signal information which may be desirable for target recognition, as the noise is only added to the final SAR image. With gaussian noise the resulting SAR image is shown in figure 5.16.

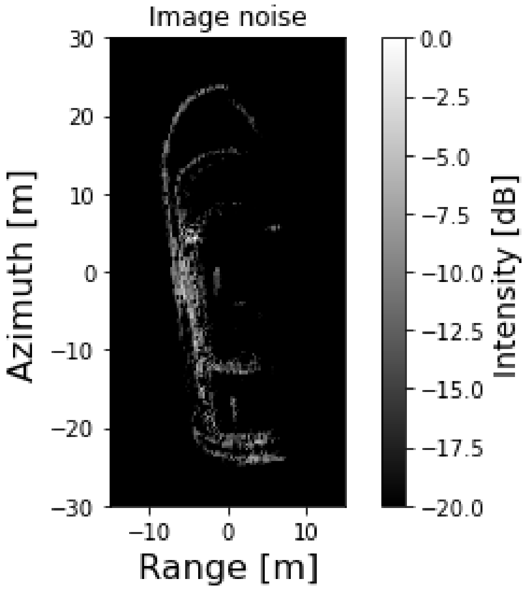
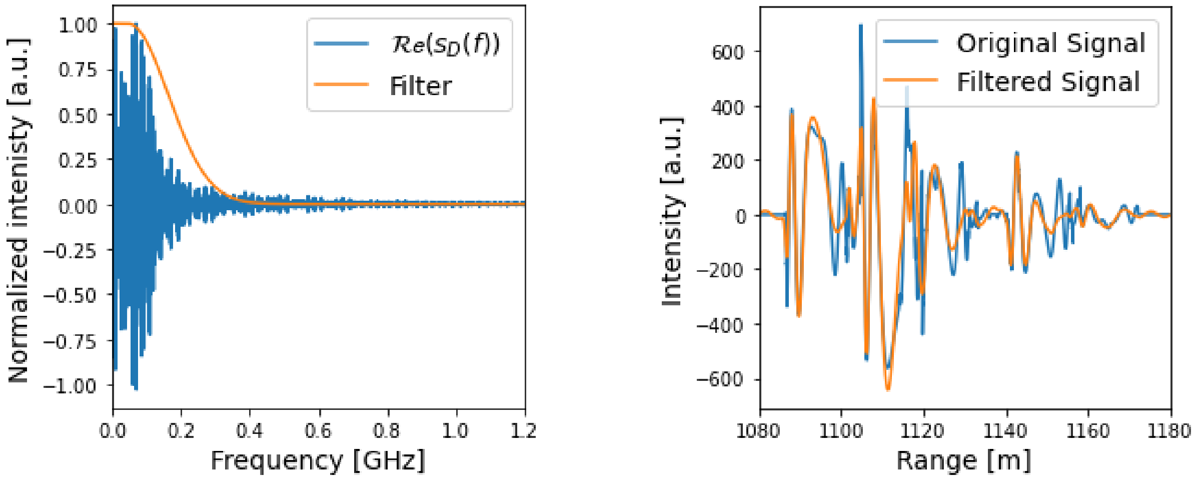


Figure 5.16. *Noisy SAR image, with image noise.*

Captured THz SAR Image

Here we compare a SAR image of the same model, measured with the THz setup with the appropriate scaled parameters. To get a simulation that closely matched the measured model, a combination of ray-tracing noise and signal noise is added. Additionally a simple smoothed step function filter in the frequency domain was also applied to the signal $s_D(f)$, see figure 5.17a. The result of the filter on the ideal signal $s_D(t)$ is shown in figure 5.17b.



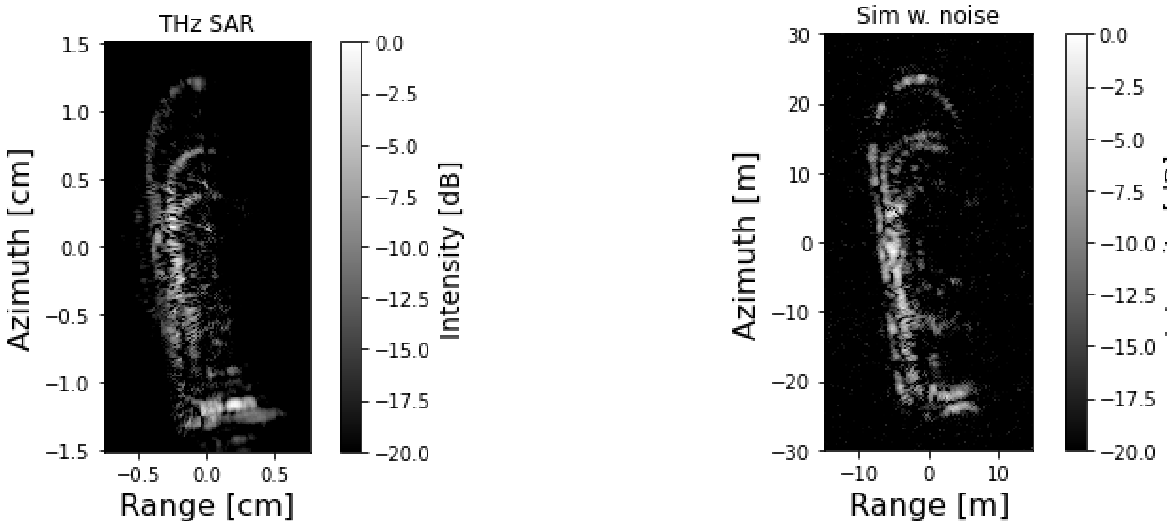
(a) **(b)**

Figure 5.17. ***(a)*** *Real value of the original normalized signal $s_D(f)$ plotted together with the filter.* ***(b)*** *Output signal in time domain with and without the frequency filter.*

To get a better outcome, we construct the new signal as a sum of the normalized ideal signal $s_i^n(t) = s_i(t)/s_i^{max}$ and the normalized signal with ray-tracing noise $s_{rt}^n(t)$, with the following weighting

$$s_{tot}(t) = 2s_{rt}^n(t) + s_i^n(t). \tag{5.6}$$

The signal noise is then added to the new signal $s_{tot}(t)$ with the frequency filter. The resulting SAR simulation is presented in figure 5.18 along with the experimentally measured SAR image of the same model printed in a scale of 1 : 2000.



(a) **(b)**

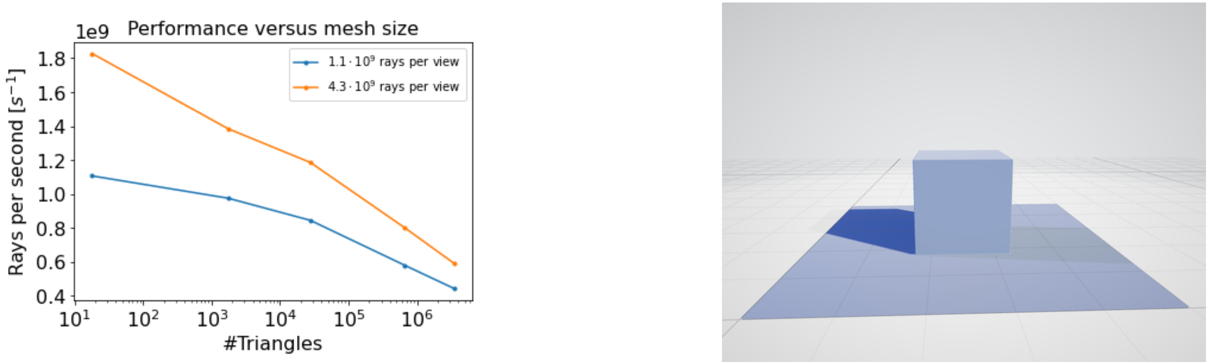
Figure 5.18. ***(a)*** *THz SAR measurement of boat.* ***(b)*** *Simulation with noise.*

The resulting SAR simulation has a lot of the same features as compared to the THz-SAR image, namely speckle and variations along smooth edges. A proper validation method is still needed to determine the best noise parameters, of which there are quite a few, across different models and radar parameters. Such a validation method would require the simulation of many SAR images with different parameters, which then are used to train neural network for image recognition of real SAR images.

5.4 GPU Ray-Tracing

5.4.1 Performance

One of the main goals for the project was to make a simulation engine which can handle meshes of realistic 3D models with many elements. To test how the OptiX GPU ray-tracer program performs with a increasing number of elements a test target, seen in figure 5.19b, was used. This mesh initially consisted of 18 triangles and was subdivided up to $3.5 \cdot 10^6$ triangles. The program was run with 100 different views over a 8° integration angle and with a slant angle of 30°. The α-model from section 2.4.1 with $\alpha = 0.85$ was used here, resulting in diffuse scattering, however for this mesh the scattering profile does not significantly affect the performance.



(a) **(b)**

Figure 5.19. *(**a**) Performance of the GPU ray tracer with increasing number of mesh elements. It can be compared directly to the CPU ray tracer in figure 5.6. (**b**) Scene used to make the performance plot. It has been discretized into various amount of triangles, the exact numbers can be seen in table 5.2*

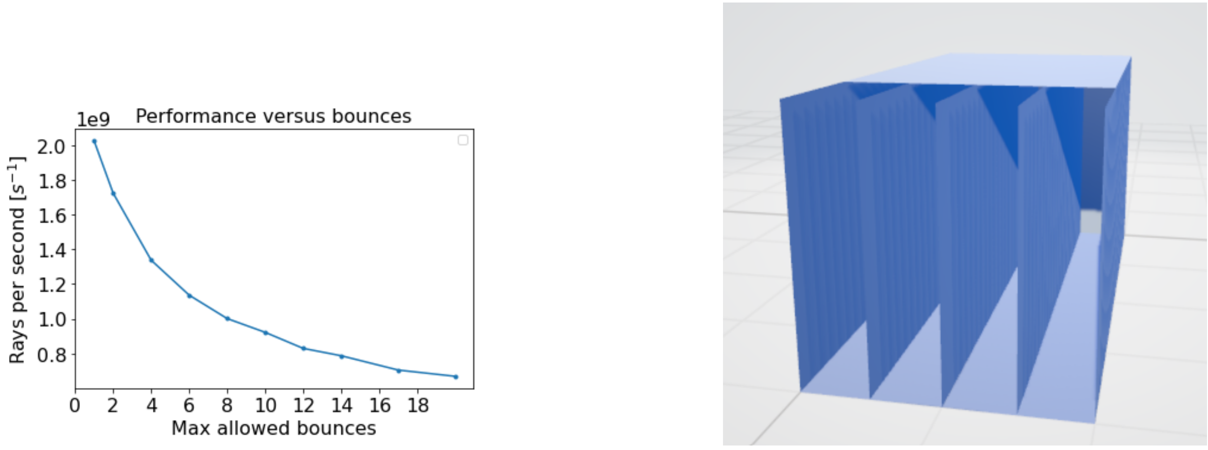
.

Figure 5.19a shows the number of new rays launched per second relative to the number of triangles. The rays per second was found by summing all rays launched over the 100 views divided by the total time for the program to run. When increasing the rays per view the program will use a higher proportion of time ray tracing compared to saving data, which can be time consuming. Hence, the discrepancy between the $1.1 \cdot 10^6$ and $4.3 \cdot 10^6$ rays launched per view. For the $4.3 \cdot 10^6$ rays launched per view, the following relationship can be seen

$$\frac{\text{Rays}}{\text{s}} = \left(\frac{\text{Rays}}{\text{s}}\right)_{\textit{one triangle}} - k \cdot \log\left(\#\text{Triangles}\right), \tag{5.7}$$

which is expected when using a BVH algorithm and makes it possible to compute large complex scenes within a reasonable time span. Furthermore, the GPU ray tracer is able to trace rays a factor of 10^4 faster than the CPU ray tracer.

Bounces can occur between the surface and box in figure 5.19b but the mesh is otherwise open and cannot trap rays. To investigate how bounces can affect the number of newly generated rays, a new test target is used seen in figure 5.20b with cavities to trap rays. Similar parameters were used as before, however, here $\alpha = 0.45$ for more specular reflections and mesh is rotated 20° in azimuth for more bounces between the vertical planes. $4.3 \cdot 10^6$ rays is launched per view and the max allowed bounces before termination is varied from 1 to 20. The result of this can be seen in figure 5.20a where the amount of newly generated rays per second will decrease when more bounces are allowed. This is a mean cavity for the ray tracer and would likely require more than 20 allowed bounces for a converged result. However, most meshes in this project will have a negligible amount of detected rays bouncing more than 6-8 times and therefore a termination of the rays will be set in this span.



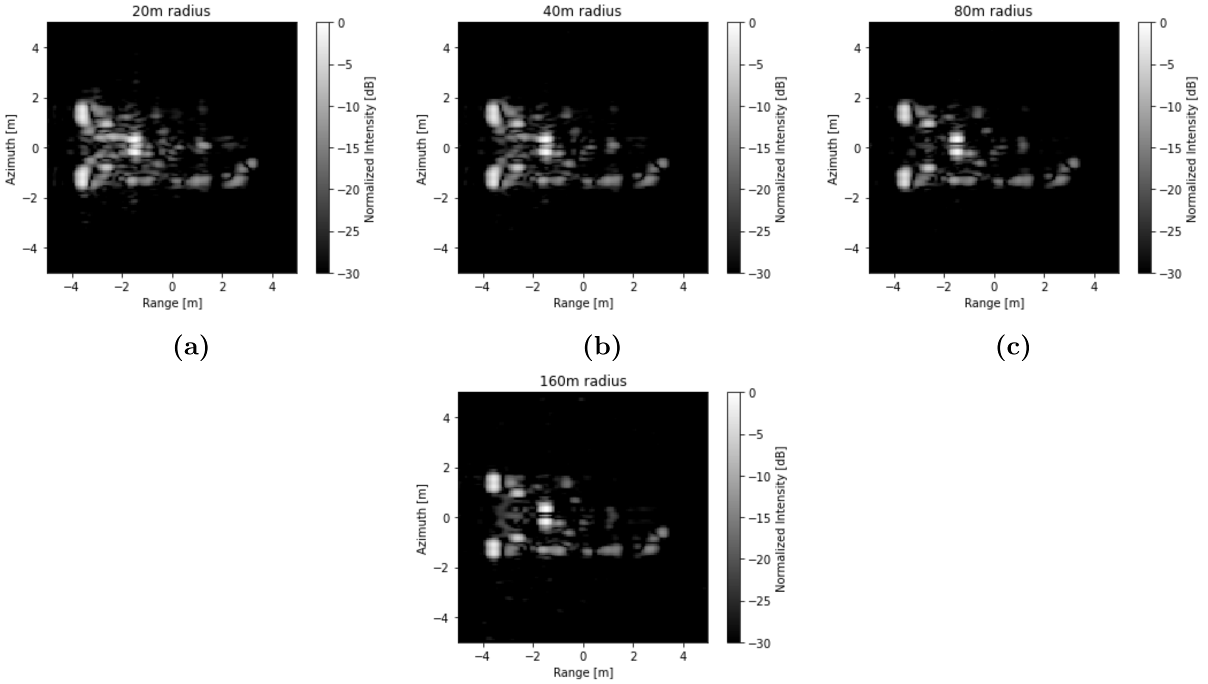
(a)

(b)

Figure 5.20. *(**a**) New rays generated per second at different max allowed bounces before termination. (**b**) The structure used to generate the plot. Rays were emitted at an azimuth and slant angle such that rays could get trapped inside the structure.*

5.4.2 Detector Size

A sphere is used as a detector and increasing the detector radius will result in more rays being cached. This makes it possible to emit fewer rays and still have a converged simulation. Hence, it is desirable to have a large detector for fast simulations but a large detector is not realistic. To observe the effect of an increasing detector four simulations are run at a slant range of 500 m with a detector radius of 20, 40, 80 and 160 m respectively. To keep the number of detected rays approximately constants the number of emitted rays per azimuth view are 2.4, 0.6, 0.15 and 0.04 billion rays respectively. This can be seen below in figure 5.21 where the target is a T72 tank and the α scattering model with $\alpha = 0.2$ is used.



(a)

(b)

(c)

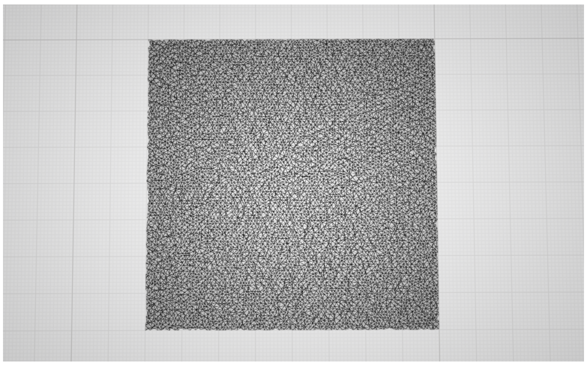
(d)

Figure 5.21. *Images of a T72 with different radii of detectors.*

Many details of the strongest intensities can be seen to be constant when going from a radius of 20 to 160m but some of the lower intensity details vanishes. A detector with a radius of 40m was settled upon and when the slant range is increased to 5000m in later sections, the detector will be increased to 400m. This is not realistic at all and one of the downsides of using ray tracing in this manner.

5.4.3 Simulated Clutter

Before simulations of objects and vehicles can be compared to real SAR data it is necessary to create a background with realistic reflectivity. For this, the clutter model described in section 2.4.3 is used. A 20 × 20 m plane surface is discretized into $20 \cdot 10^3$ triangles where the vertices have been randomly shifted. The mesh can be seen in figure 5.22a. When a ray intersects a triangle, the intersection point is moved to the centroid of the triangle, which will act as a point scatterer. Furthermore, each triangle will have a designated reflectivity which is found from a zero mean gaussian distribution.



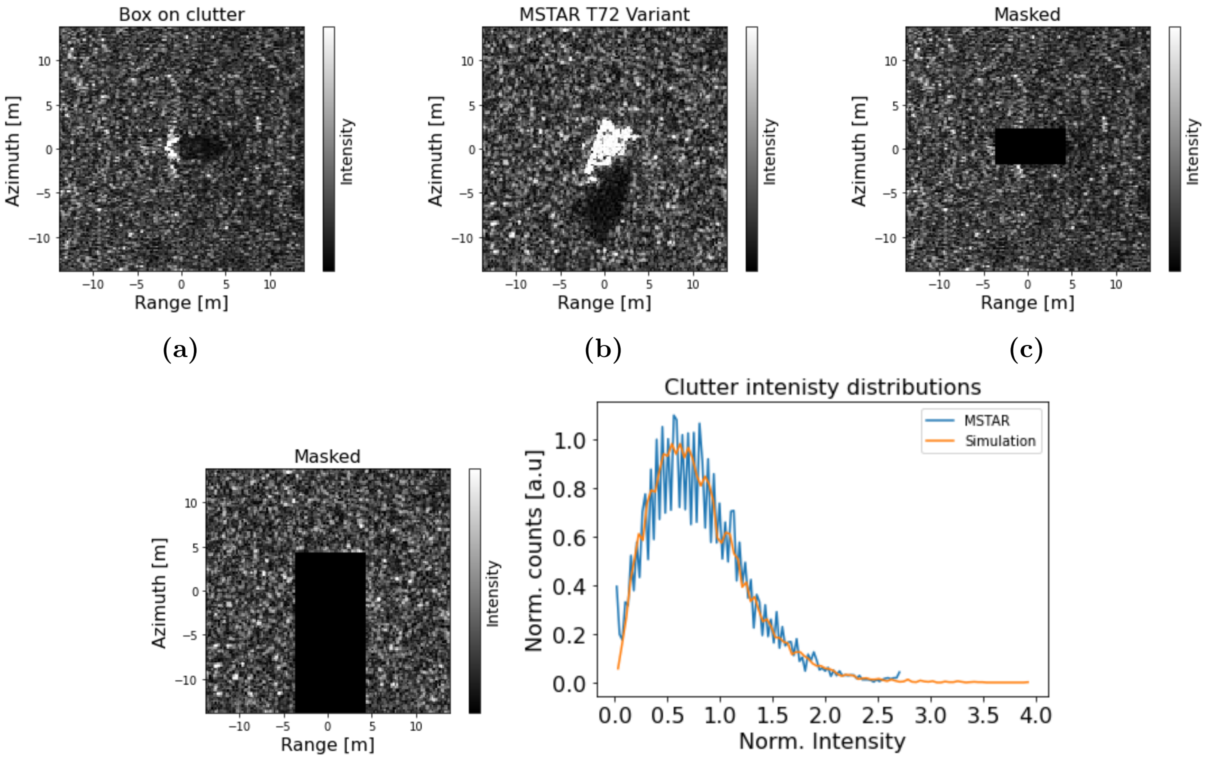
(a)

Simulation parameters	Values
Center frequency (f_c)	9.6 GHz
Bandwidth (B)	0.591 GHz
Pixel resolution	0.2 × 0.2 m
Integration angle	3.5°
Azimuth steps	240
Slant angle	15°

(b)

Figure 5.22. ***(a)*** 20×20 m *mesh used for clutter background.* ***(b)*** *Table of relevant simulation parameters, which matches the MSTAR dataset.*

To compare this method with real data, a 2m × 2m cube is placed on top of the plane to demonstrate the shadowing effect which can be seen in figure 5.23a. For comparison a real SAR image of a T72 tank from the MSTAR dataset can be seen in figure 5.23b. In the simulated image, the scene is illuminated from left to right while the MSTAR scene is illuminated from top to bottom. In both cases a shadow can be seen casted on the background in the direction of illumination. To compare the surrounding clutter in the two images, the objects and shadows have been masked which can be seen in figure 5.23c and 5.23d.



(a) **(b)** **(c)**

(d) **(e)**

Figure 5.23. *(**a**) Simulated image. (**b**) Real SAR image from the MSTAR dataset. (**c**-**d**) Same two images but masked. (**e**) Clutter distribution in simulated and MSTAR image.*

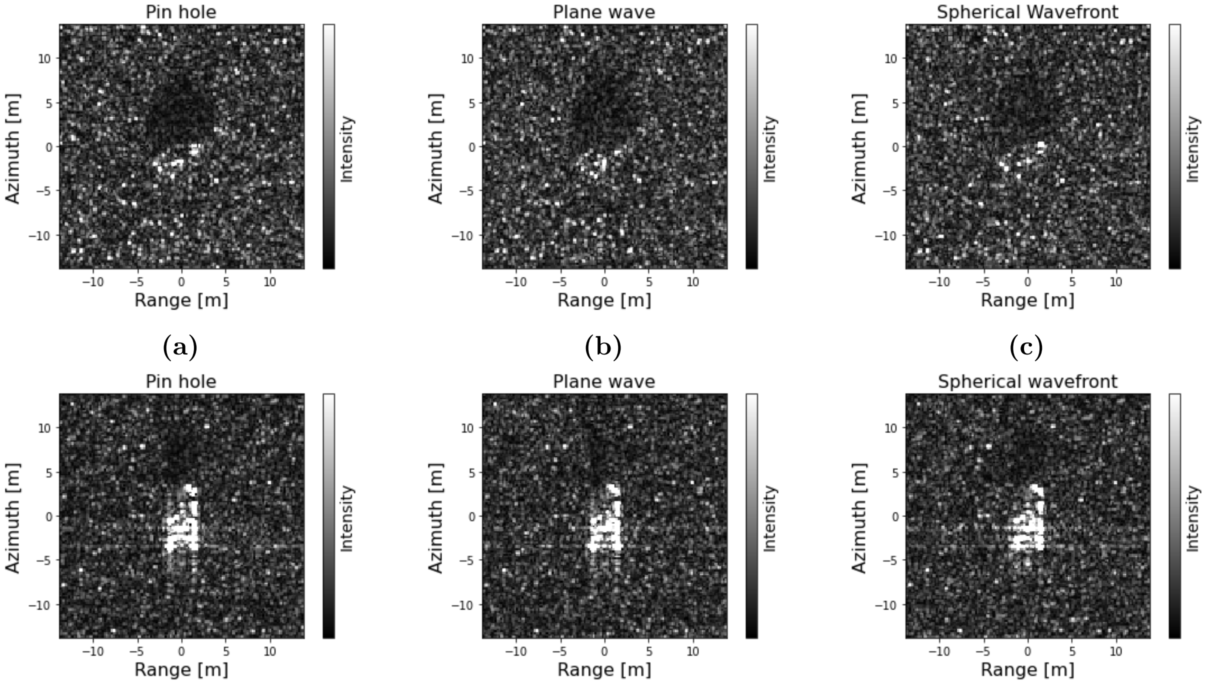
The unmasked pixels are then arranged into intensity bins to plot the intensity distributions which can be seen in figure 5.23e. The normalized intensity has been found by dividing with the average intensity and the normalized counts are found by

$$\text{Norm. counts} = \frac{\text{counts}}{(\text{Total counts}) \cdot (\text{bin size of Norm. intensity})}. \tag{5.8}$$

The simulated clutter pixel intensities can be seen to follow a similar distribution to that of the MSTAR image and the zero mean gaussian distribution for background reflectivity results in realistic clutter. The high intensity pixels in the MSTAR image can be seen to be more clustered together than in the simulation however, the clutter model is considered sufficient for this project.

5.4.4 Ray Emission

Three different ray emitting schemes have been implemented, which is a pin hole, a plane wave, and a spherical wave emitter. The pin hole emitter is often used in ray tracing in computer graphics, to obtain images similar to a pin hole camera. The rays here have a common origin and the direction of rays are found by points on a plane at a distance from the origin. In the plane wave emission the rays are emitted orthogonal to a plane and hence have different origins. In the spherical emitter the rays have the same origin and they are casted upon a unit sphere. The difference between the emitter types can be seen in figure 5.24 where the same scene of a tank at two different view angles have been simulated at a slant range of 500 m.



(a) (b) (c)

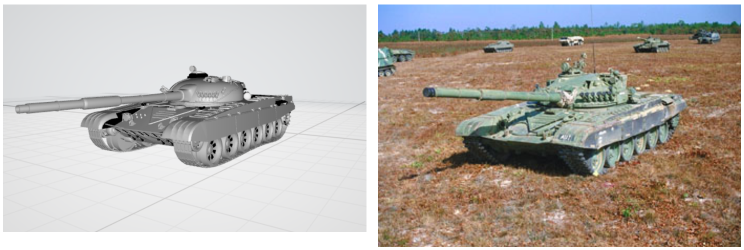
(d) (e) (f)

Figure 5.24. *Scene of a T72 tank simulated with different emiter types.* (***a-c***) *incidence at* 45° *in azimuth with respect to the front and* (***d-f***) *incidence from the front.*

Slight differences can be noticed between the emitter types however, most features in the images appears to be nearly identical. The simulations presented later in the project will use the spherical wave front but at a slant range of 5 km. This makes the difference between the plane and spherical front nearly indistinguishable since our scenes is on a scale of 20 × 20 m.

5.4.5 Optimizing Simulation Parameters to MSTAR Dataset

The GPU ray-tracer can simulate large meshes within a reasonable time span. Hence, the next step is to find simulation parameters that yield realistic images. For this, the three different scattering models mentioned in section 2.4.1 are compared to real SAR images which can be seen in figure 5.26. Here the most promising parameter within each scattering model is shown. The data for real SAR images comes from the MSTAR data set and has been downloaded from [77]. The intensities in these images are plotted linearly with a grayscale map and our simulated images are presented in a similar manner without an intensity bar. The vehicle used for comparison is a T72 and 5 different views in azimuth is used. The 3D model used in simulations, optical image of a T72 and SAR parameters matching the MSTAR data can be seen in figure 5.25. The 3D model of the tank and vehicles in later chapters has been obtained from the internet.



(a)

(b)

Simulation parameters	Values
Center frequency (f_c)	9.6 GHz
Bandwidth (B)	0.591 GHz
Pixel resolution	0.2 × 0.2 m
Integration angle	2.94°
Azimuth steps	153
Slant angle	15°

(c)

Figure 5.25. *(**a**) T72 3D model. (**b**) Optical image of MSTAR T72. (**c**) Radar parameters used in simulation.*

Taking a look at figure 5.26 from the top down, the first row is real images that our simulations ideally should match well. Below, two rows of the α-model can be seen, where a lower α-value will give more specular reflections. The $\alpha = 0.05$ parameter was found to yield the best results and converge faster than more diffuse reflections. Metallic objects on this scale are also expected to have specular reflections in the microwave regime. However, the real SAR data have relatively high intensities from places on the T72 that the specular α-model cannot achieve on its own.

The γ-model tries to overcome this problem by having a possibility of scattering backward, where the backward probability is $1-\gamma$. This makes the object appear more illuminated compared to the α-model. However, this method converges poorly and a black line can be seen horizontally across the images. This can be fixed by launching more rays, but there are already launched $92 \cdot 10^9$ rays per image, which is also the case for the α-model. Increasing the number of rays will affect the computational time and is not desired.

To improve the convergence of the γ-model, the intersection point of directly back scattered rays is shifted to the centroid of the triangles. This method is denoted as γ-model-2 in figure 5.26. Furthermore, the directly back scattered rays are weighted with the squared dot product between the incidence vector and normal vector. Additionally, the directly back scattered rays from the clutter is made to go more towards the detector for even better convergence. The images presented by this method are simulated with only $5.5 \cdot 10^9$ rays and are better converged than the previous γ-model.

The Λ-model presented in last row of figure 5.26 uses the same tricks for convergence as γ-model-2, but the rays will a probability of $1 - \Lambda$ to be fully diffuse. The results are here close to the α-model but with a blurry effect better resembling the real SAR data than the specular reflections in the α-model.

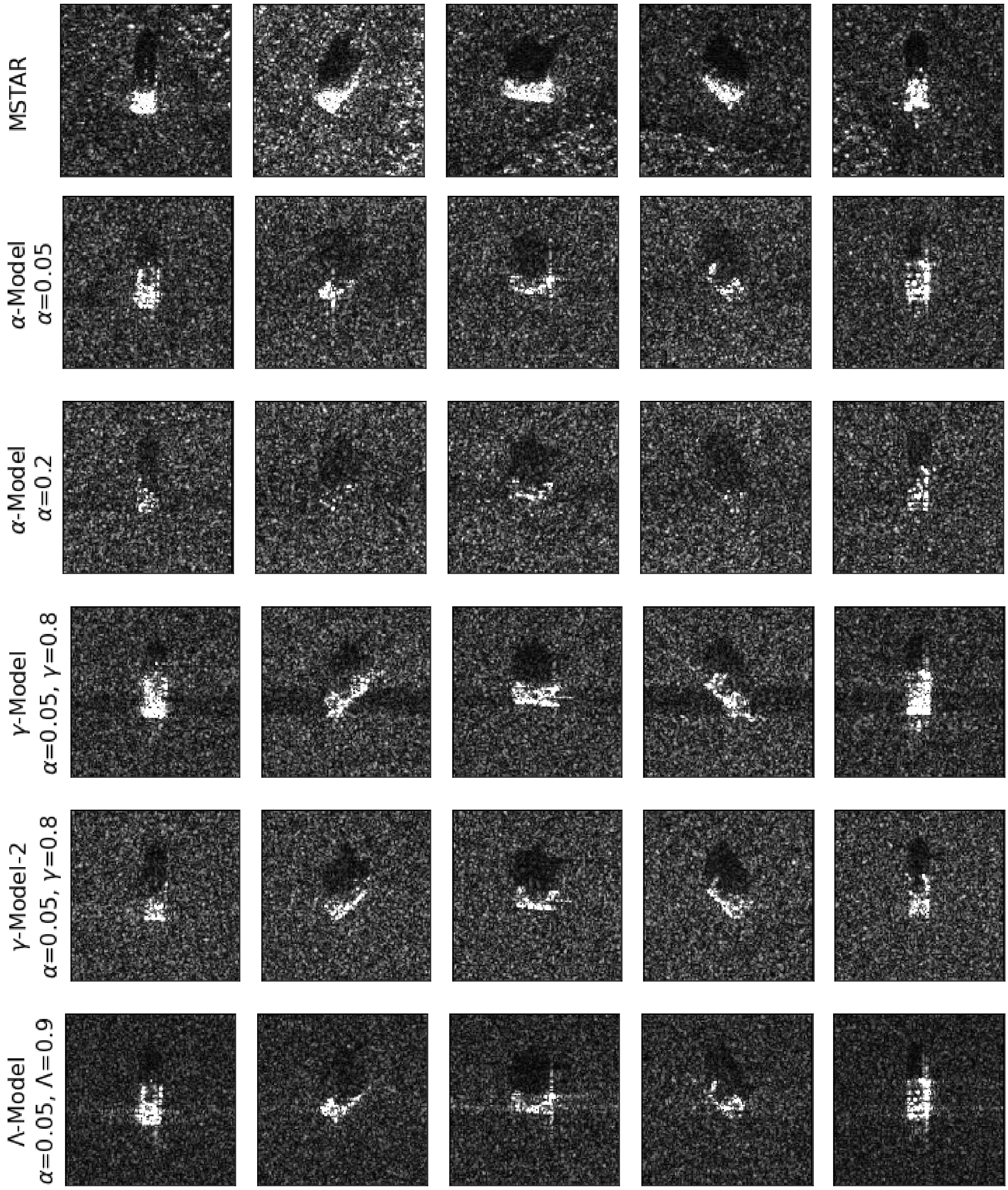
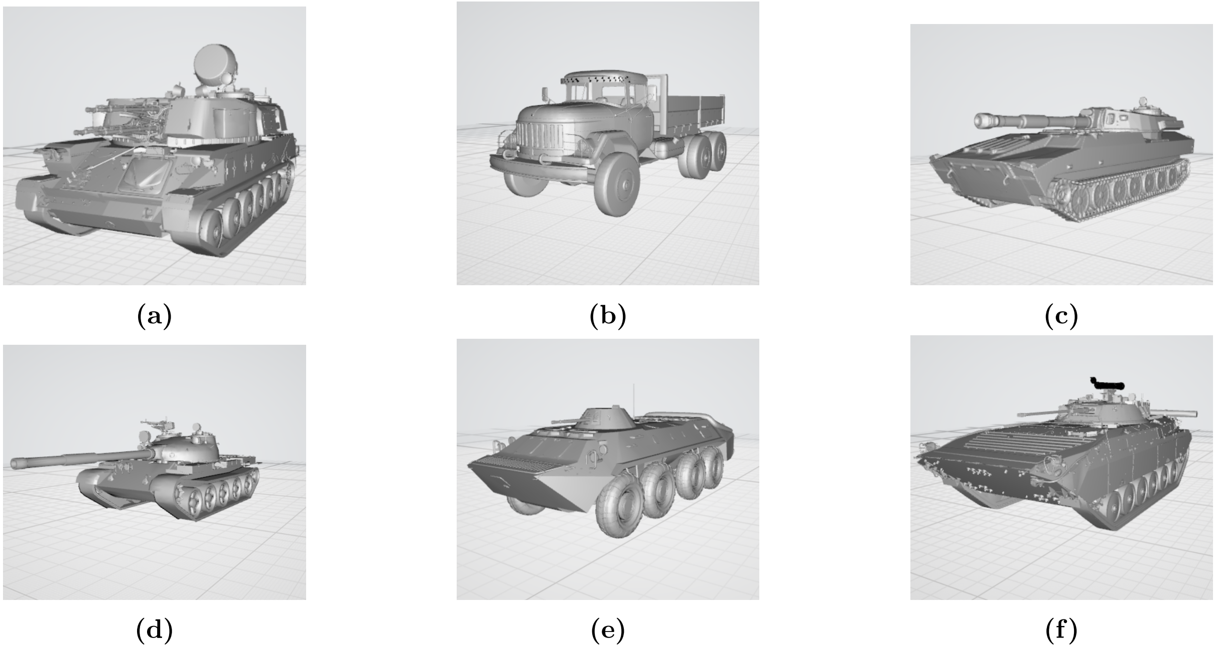


Figure 5.26. *Array of SAR images of a T72 tank. Each column have a different azimuth view. The top row is real SAR images from MSTAR while the rest of the rows are simulations with different scattering models.*

The Λ-model images were generated with $5.5 \cdot 10^9$ rays similar to the γ-model-2, but appears to converge slightly worse. The γ-model-2 was chosen for further simulations due to the fast convergence and reasonable similarity to the MSTAR T72 images. The time to generate an image with this model is ∼30s while the α and γ-model takes between 5-7 minutes. Ideally a more objective measurement than comparing the images with the human eye should be used.

5.4.6 Simulated Dataset of Different Vehicles

To further test our simulations six additional vehicles were simulated and the 3D models can be seen in figure 5.27. The γ-model-2 and the same parameters were used for all simulations, which can be seen in table 5.25c.

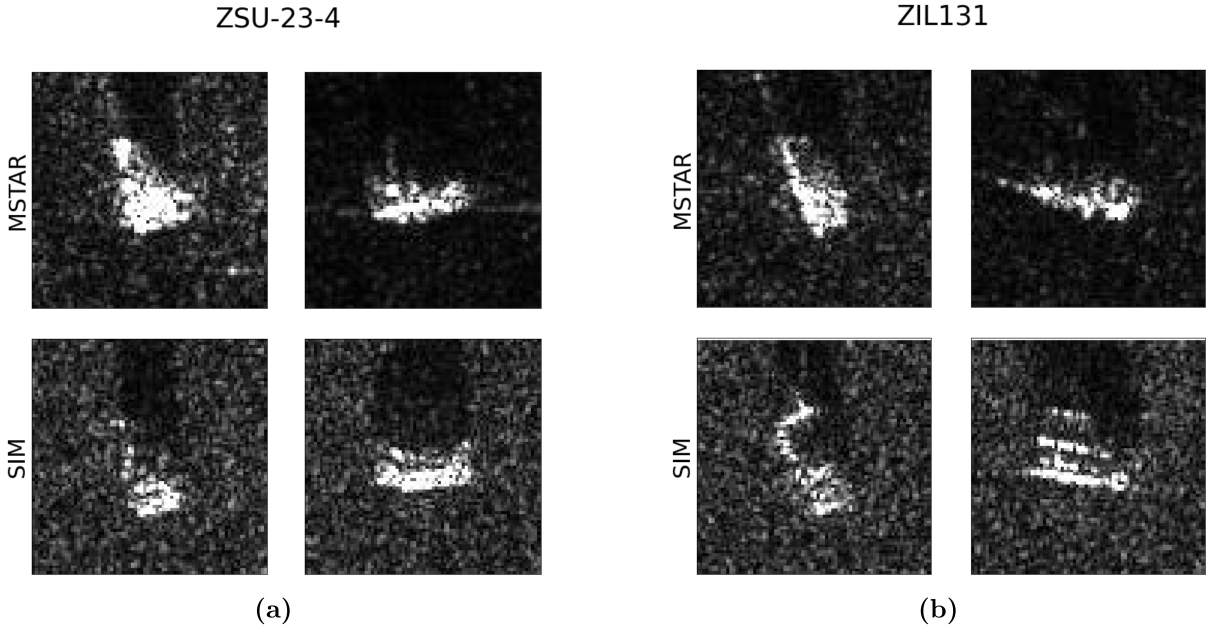


(a) (b) (c)

(d) (e) (f)

Figure 5.27. *3D models of vehicles used in simulations.* ***(a)*** *ZSU-23-4.* ***(b)*** *ZIL131.* ***(c)*** *2S1.* ***(d)*** *T62.* ***(e)*** *BTR70.* ***(f)*** *BMP2.*

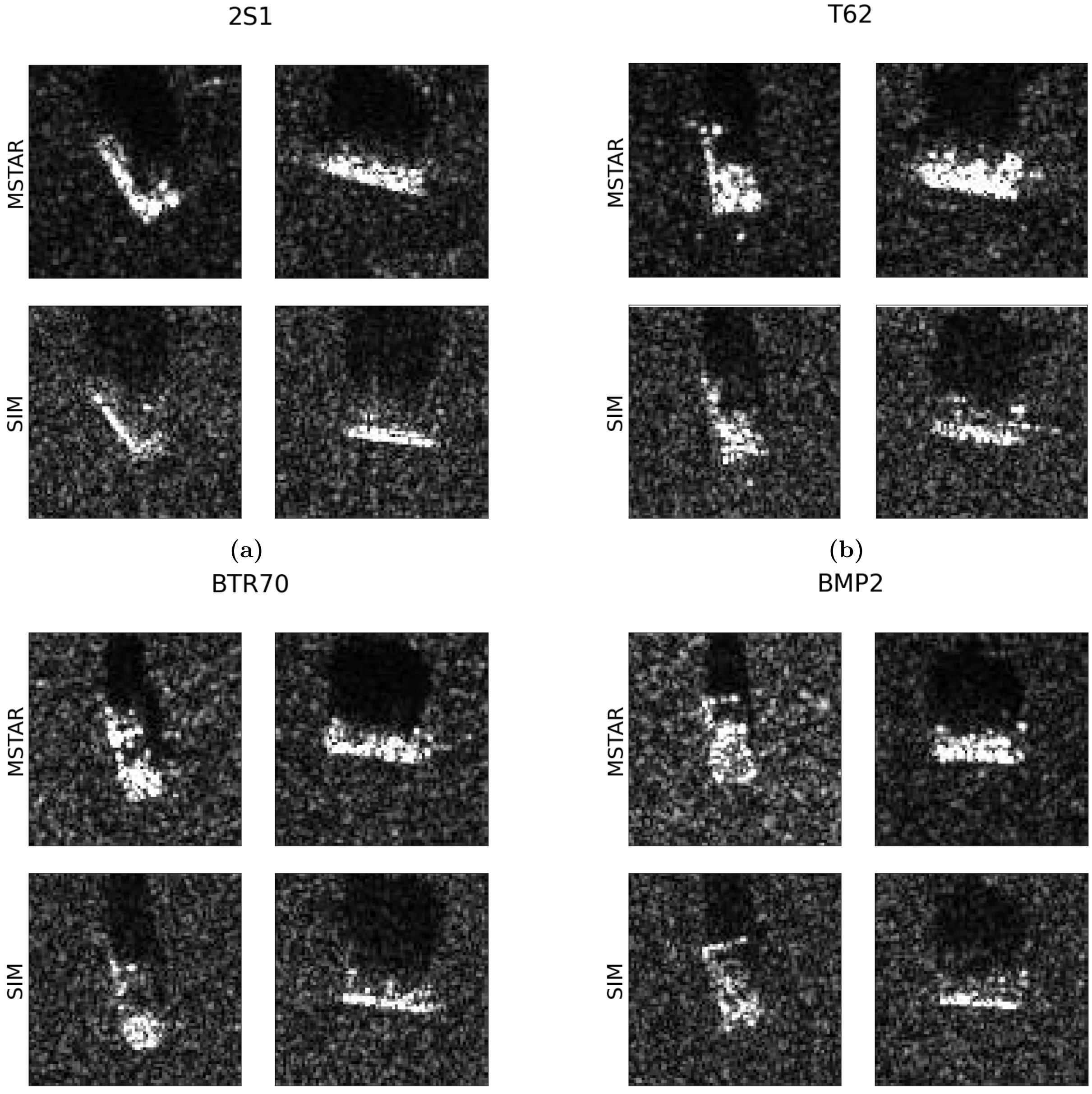
Two simulations from different azimuth views are presented for each vehicle together with similar MSTAR images in figure 5.28 and 5.29.



(a) (b)

Figure 5.28. *MSTAR and simulated images of* ***(a)*** *ZSU-23-4 and* ***(b)*** *ZIL131.*

When comparing the simulations and MSTAR data in figure 5.28a it can be seen that the areas of the illuminated target are similar but there is a difference between the smaller features and how much intensity different areas has. The ZIL131 model seen in figure 5.27 can have either an open or closed container in the back. The variant in MSTAR most likely has a closed container which is not the case for the 3D model used in our simulations. This can explain the different features and lack of bouncing effect in the MSTAR images. The bouncing effect can be seen as a line in the shadow region of the simulation.



(a) **(b)** **(c)** **(d)**

Figure 5.29. *MSTAR and simulated images of (**a**) 2S1, (**b**) T62, (**c**) BTR70 and (**d**) BMP2.*

In figure 5.29 the outline of the vehicles is similiar between simulations and MSTAR. However the smaller features and intensity distributions varies and better scattering profiles should probably be implemented to obtain better simulations.

5.4.7 Convolutional Neural Network Trial

We wanted to test whether or not a convolutional neural network (CNN) can be trained on our simulated data to recognize objects in real SAR images. For this 2100 simulations were run in total, where each vehicle from figure 5.27 and 5.25a was simulated from 150 different azimuth views from both a slant angle of 15° and 17°. Furthermore, for test and comparison 3576 MSTAR images of the same vehicles at slant angles of 15° and 17° were used and tables of the data can be seen in figure 5.30.

Simulated dataset		
Vehicle	15°	17°
T72	150	150
ZSU-23-4	150	150
ZIL131	150	150
2S1	150	150
T62	150	150
BTR70	150	150
BMP2	150	150

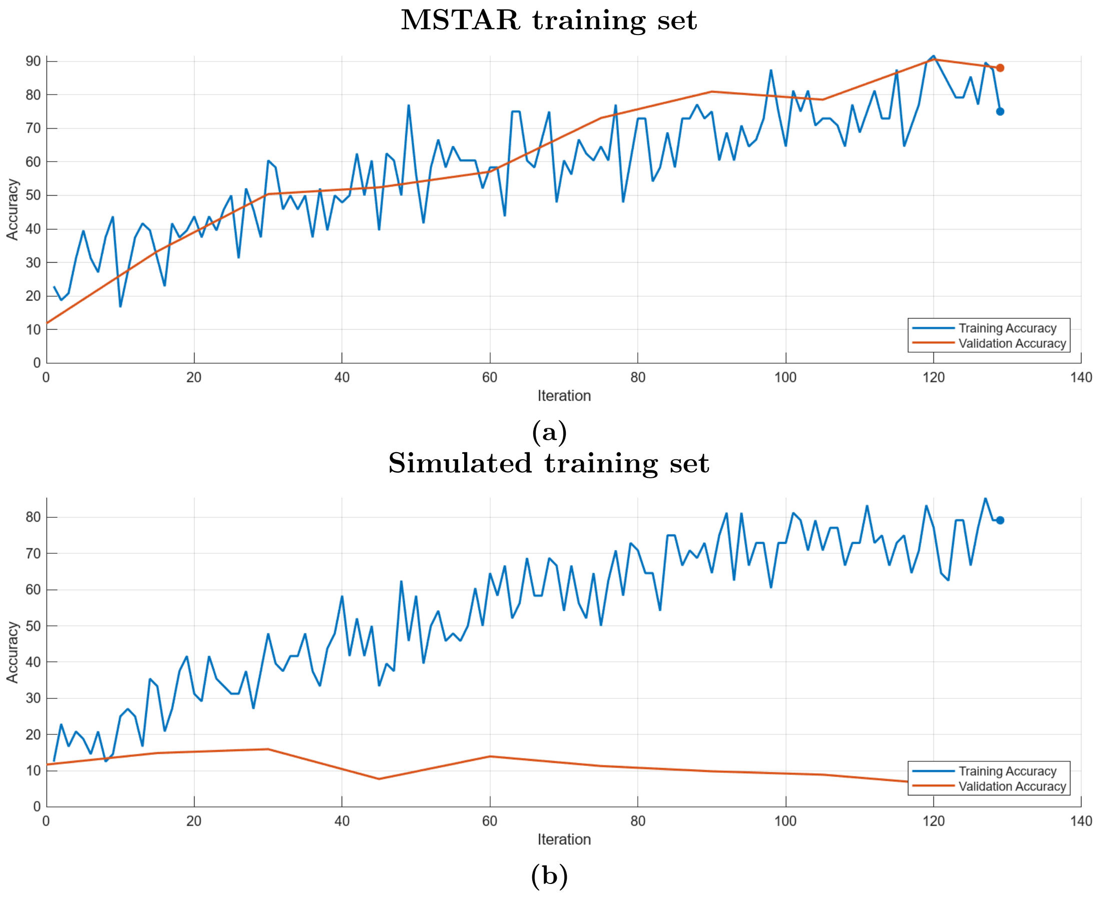
(a)

MSTAR dataset		
Vehicle	15°	17°
T72	232	196
ZSU-23-4	299	274
ZIL131	299	274
2S1	299	274
T62	299	273
BTR70	233	196
BMP2	233	195

(b)

Figure 5.30. *Number of images in the simulated and MSTAR dataset for different vehicles at slant angles of* 15° *and* 17°.

The CNN used is described in section 3.8 and first the model were trained and tested on pure MSTAR data. A training set with a fraction of 0.58 of each label in the MSTAR dataset was used to get a comparable size of data to the simulations. A fraction of 0.21 were used for both validation and test, respectively. The training and validation accuracy across iterations can be seen in figure 5.31a.



MSTAR training set

(a)

Simulated training set

(b)

Figure 5.31. *Accuracy throughout iterations of CNN on training and validation for (**a**) pure MSTAR images and (**b**) where the training set is simulations.*

The accuracy is here defined as

$$\text{Accuracy} = \frac{\#\ \text{Correct predictions}}{\#\ \text{Total predictions}}. \tag{5.9}$$

When training on pure MSTAR images the model can learn to distinguish between the different vehicles and since the validation dataset is similar to the training set the accuracy for both sets increases similarly across the iterations. The final test accuracy on the test set was 0.903. On the other hand when using the full simulated data set for training and 0.21 of the MSTAR set for validation and test respectively, the model are again able to classify the vehicles within the training set, but accuracy increases at a slower rate than the MSTAR training set. This can be seen in figure 5.31b. However, the validation accuracy are terrible which indicates than the model learns features in the simulations that are not present in the real data. The final accuracy on the test set was 0.16 which is similar to the random chance. This optimistic trial on the simulated data to predict real vehicles shows that the simulations should be optimized further with better parameters than what looks similar to the naked eye. However, this is out of the scope for this project.

5.5 Computer Simulations Discussion

The geometrical ray-tracing approach to radar simulations was chosen for this project, however, this method relies on geometrical optics which does not inherently account for diffraction effects and polarization effects. Solving Maxwells equations numerically with methods such as finite difference time domain (FDTD) or method of moments (MoM) would yield the most accurate results since wave phenomena are inherently included in the models. However, these approaches comes at a cost in computational complexity and simulation time. A hybrid approach of ray-tracing and FDTD was implemented by Kulpa et al. [78] where landscapes and buildings on a scale of square kilometers were simulated with geometrical optics. Meanwhile metallic objects of interest in the scene was simulated by FDTD and signals from both methods were combined and used to generate SAR images. The main downside of this approach was that calculating a object within a volume less than 1m^3 with FDTD would result in simulation times of up to ten days for a single SAR image. A similar approach could be combined with our simulation engine but an extreme price in speed would have to be paid for an increase in accuracy.

Another approach which is commonly used in calculating the scattering of microwaves is physical optics (PO). In PO a high-frequency assumption is used to make approximations that significantly reduces the computational complexity of solving Maxwells equations. Furthermore, it can be combined with ray-tracing in a method denoted as *shooting and bouncing rays* (SBR). Here rays are emitted and can bounce on a scene similar to our approach, however, each time the ray intersects a primitive, PO will be used to integrate over the hit surface and calculate the directly backscattered E-field to the observer. This was implemented in the Nvidia Optix engine by Xu et al. [79] and could also have been implemented in this project. The advantage of this method is that polarization of E-field is included in the PO model and that it does not require as much parameter optimization as our scattering profiles does in our simulations. The downside of SBR and PO is that the high-frequency approximation assumes that all surfaces are perfect electrical conductors and perfectly flat and smooth relative to the size of the wavelengths. These assumptions can cause very idealized simulations that can differ from real measurements. Our approach of adding custom scattering profiles to materials relies less on physical assumptions and allows for more room to tweak simulations. Importantly, we wanted to avoid the PO and SBR assumptions and have the possibility to add more diffuse scattering in our simulations.

One of the main goals for the project was to simulate life size complex objects in a timely manner. First the CPU ray-racer from a previous project was optimized significantly, hereafter NVIDA optix was used to implement ray-tracing on a GPU. The advantage of the GPU is that it is faster and we can simulate life size scenes with a reasonable throughput. However, it is easier to implement changes and modifications in Python on a CPU compared to C++ and a GPU, which is the reason why the polarimetry ray-tracer was only implemented on the python ray-tracer. Nevertheless, the throughput of the GPU ray-tracer allowed for simulating a dataset comparable to real SAR images from the MSTAR dataset. Taking these simulations directly and using them for training a CNN did not yield good results, which could also be expected. Cha et al. [80] tried to train a CNN directly on simulated data as well and test on real SAR images. They got an accuracy of 20% when using a simulated dataset of 3600 images of 10 vehicles. To improve the model training, they used a deep residual network to improve their simulated images and when used for model training they achieved an accuracy of 80%. Another study by Inkawhich et al. [81] showed that more advanced data augmentation and model construction can be used to more successfully train a CNN on purely simulated data. This was not within the scope of this project but could in the future be investigated together with improving the simulation engine if model training is of interest.

5.6 Summary

In this section we showcase the results on implementing a BVH on the performance of the CPU ray-tracer along with other optimizations. Additionally the Optix GPU implementation is characterized and tested against MSTAR data with different scattering profiles. A model was eventually selected, with a total computation time of ∼30 s pr. SAR image. The resulting simulated SAR images of different vehicles are shown to resemble MSTAR well. Taking thing a step further, a CNN was trained on simulations to identify different MSTAR objects, with poor results.

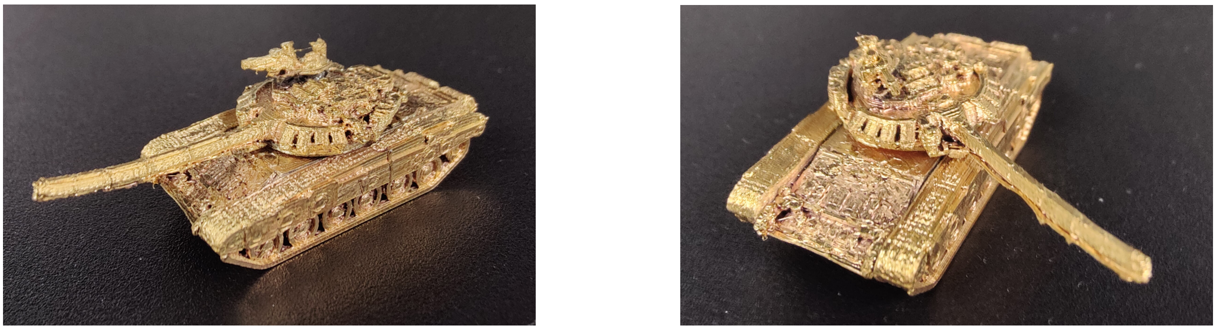
Reduced Scale Models 6

6.1 THz Images of Reduced Scale Objects

The ISAR setup can be configured with two different lens setups allowing for two different spot sizes of different diameters; 4 cm and 8 cm. Prior to production of the second lens setup, allowing for larger a spot size, different models have been imaged using the small spot size. The SAR images are plotted with a custom colorbar, starting out linear and turning logarithmic.

6.1.1 Reduced Scale Models

In this section the results of the THz setup for measuring reduced scale models are presented. Small scale models of a T72, a ford pickup and a boat have been 3D printed and gold coated. This is in part to define the reduced scale setups ability to mimic SAR environments including armored vehicles. Additionally, SAR images of two larger toy cars in metal have also been formatted. All the models are shown in appendix A.3. To showcase the detail and surface of the fabrication process the T72 model is also shown in figure 6.1. With 3D printing the details of the models can be captured well, such as the armor plating and individual wheels. The tank gun appears to bend upwards. This is caused by the gold deposition, which heated the plastic and as the gold deposited created stresses bending it upwards. However, this minor detail was not significant for overall images. Especially for extrusion printing, the surfaces can however be rough, as seen on the models including T72, ford and the first boat print. The surfaces of models printed with the resin printer were much smoother, and closer to reality, due to the increased printer resolution.



(a)

(b)

Figure 6.1. *Images of gold plated 3D printed T72 tank with rotatable turret rotated **(a)** 0 degrees, and **(b)** 45 degrees*

The T72 was measured in five scenes, which are given the names tank 1-5. Tank 1-3, are measured on a non reflective background. Tank 4-5 are measured on clutter. The different scenes, measure the tank at different orientations. The resulting SAR images are shown in figure 6.2.

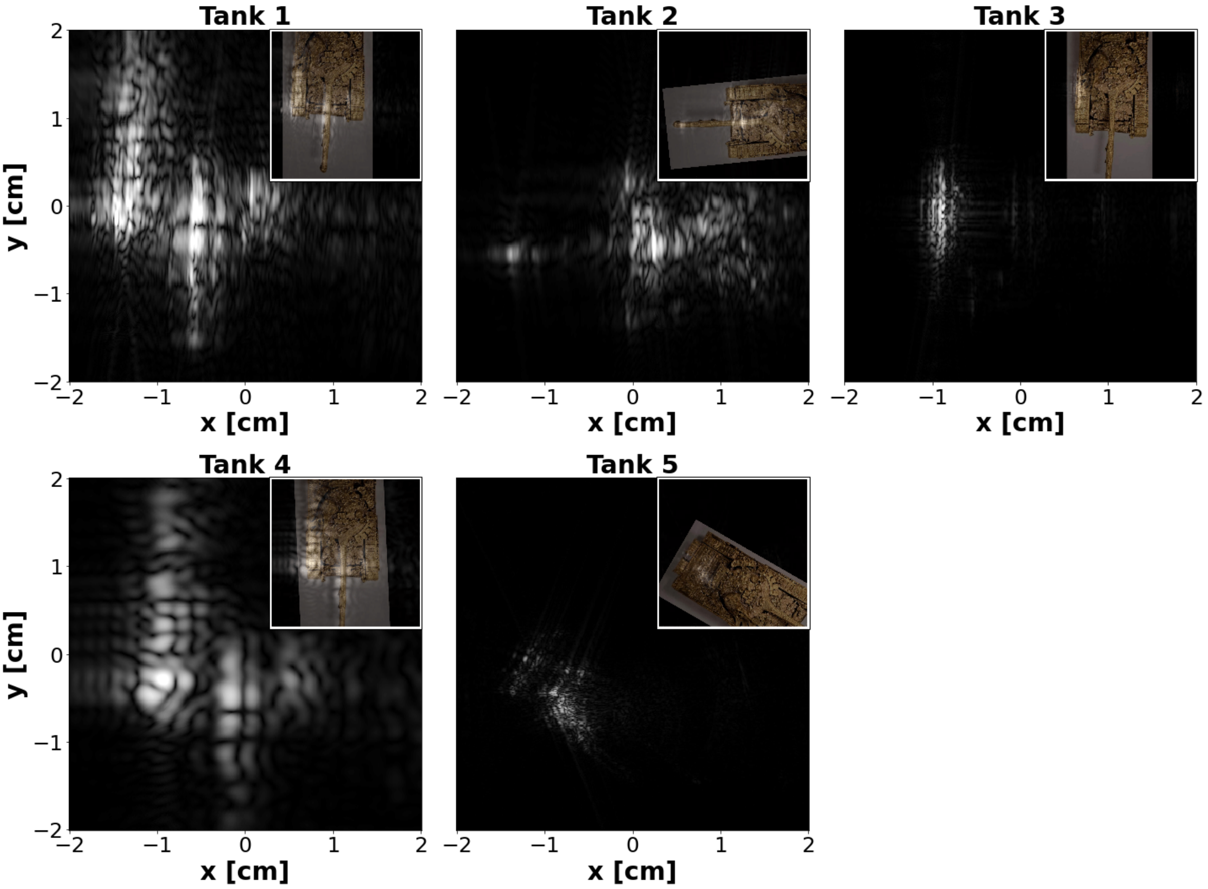


Figure 6.2. *THz SAR images of T-72 reduced scale model.*

Tank 1 and 4 are easily identified as the front of the T72 by the barrel and threads, as seen from the side. The clutter in tank 4 has a minimal effect on the final SAR image, likely due to the reflections from the tank drowning out the clutter. The orientation in tank 3 is identical to tank 1, but due to the limited spot size of the THz setup only the side of the T-72 is visible. Tank 2 offers a front on view, where both the barrel and main body of the tank are shown. Generally the tank model is quite rough, which results in somewhat grainy SAR images. However, the size and shape of the model is well captured in the SAR images.

Ford Pickup

A smaller model, in this case a pickup truck car was fabricated, which allowed us to image the entire model with the spot. The car model was measured in four scenes, which are given the names car 1-4. All measurements are done on a non reflective background. The resulting SAR images are shown in figure 6.3.

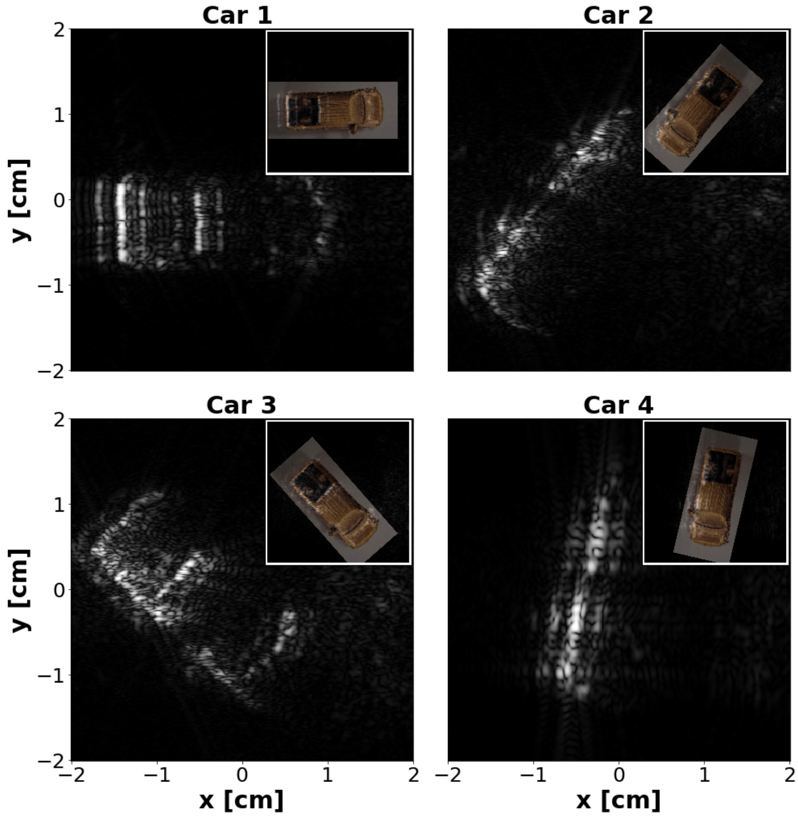


Figure 6.3. *THz SAR images of car reduced scale model.*

In all four SAR images, the car model is easily distinguishable. The size and shape of the model is captured well on the SAR images. Note the degree of detail, like in car 2, where the side mirror is visible. In these SAR images, there are some ghost images present just before and after smooth surfaces, like the artifact described in section 3.4. By eliminating these artifacts improved SAR images are obtained, see figure 6.4. In these SAR images the lines of the car model are much less blurry. In practice all the reduced SAR scale models could be improved by the same artifact suppression method. However, in the interest of time we have only shown the effect on the car model, which also have the most obvious ghost.

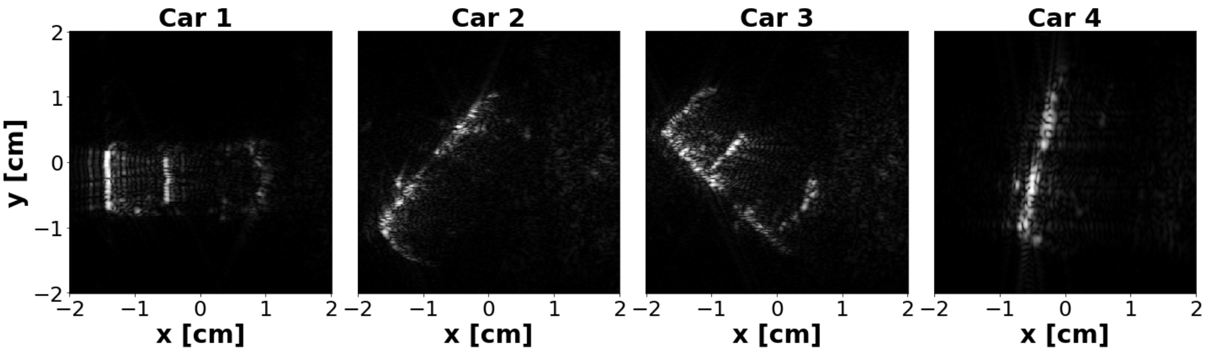


Figure 6.4. *THz SAR images of car reduced scale model with artifact suppression.*

Boat

Here a boat model was measured, at two very different integration angles, of 40° and 208°, named boat 1 and boat 2 respectively, see figure 6.5. By increasing the integration angle the azimuth resolution is increased. Furthermore, a larger area of the boat is also illuminated, resulting in a more detailed and complete image of the model. In both images the general shape of the boat is clear with good distinction of the different levels on the model.

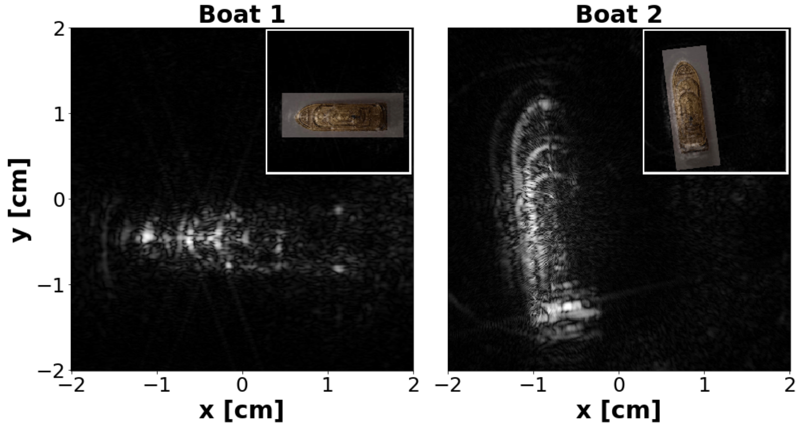


Figure 6.5. *THz SAR images of boat reduced scale model.*

Large Models

Two toy models, a blue car and a jeep, were also measured. As shown in figure 6.6, the scale is quite a bit bigger than the previous results. Particularly in the blue car SAR image the front of the car is well defined, and faint traces along the open door are also present. No distinct feature of the jeep was visible, this could be attributed to the limited spot size of the small lens setup.

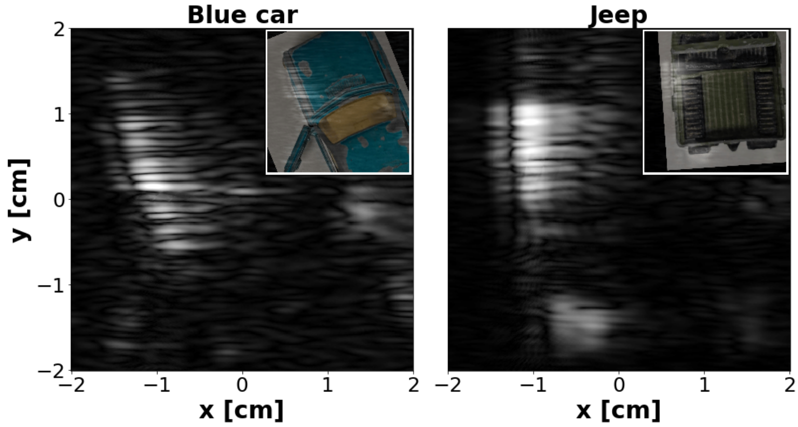


Figure 6.6. *THz SAR images of toy jeep and blue car.*

Simulated THz SAR

True scale simulations of the boat and ford car model have been computed with the Optix raytracer, see figure 6.7. The scale of the boat and car are respectively 1 : 2000 and 1 : 215. With the boat, that means that one 1 cm in the THz SAR corresponds with 20 m in the simulation.

The resulting simulated SAR images are in good agreement with the measured THz SAR images. Particularly the simulations of car 1 and car 2, respectively shown in figure 6.7b and 6.7c, closely match the THz SAR images after removing the ghost artifacts. In the simulation of the boat, shown in figure 6.7a, there is not any noise, in contrast to the reduced scale image in figure 6.5, where the roughness of the 3D print brought natural noise to the image. However, the most prominent reflection features of the simulation can be matched to areas of high reflection on the THz model. The simulation show just how nicely THz SAR can be utilized in reduce scale setups of radar systems, by accurately matching the experiment with simulation.

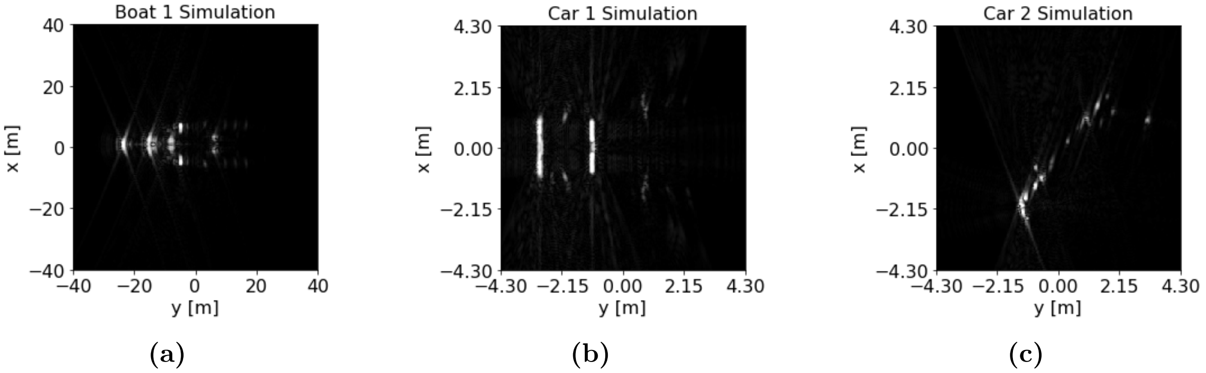


Figure 6.7. *(**a**) Simulation of boat which is simulated to be* ×2000 *larger than the reduced scale. (**b-c**) Simulation of a car, which is* ×215 *larger than the reduced scale car.*

6.2 Larger Scenes

To measured more complex scenes a larger spot size was needed in the THz setup. In this section the lens design is presented along with results of THz SAR with a larger spot size.

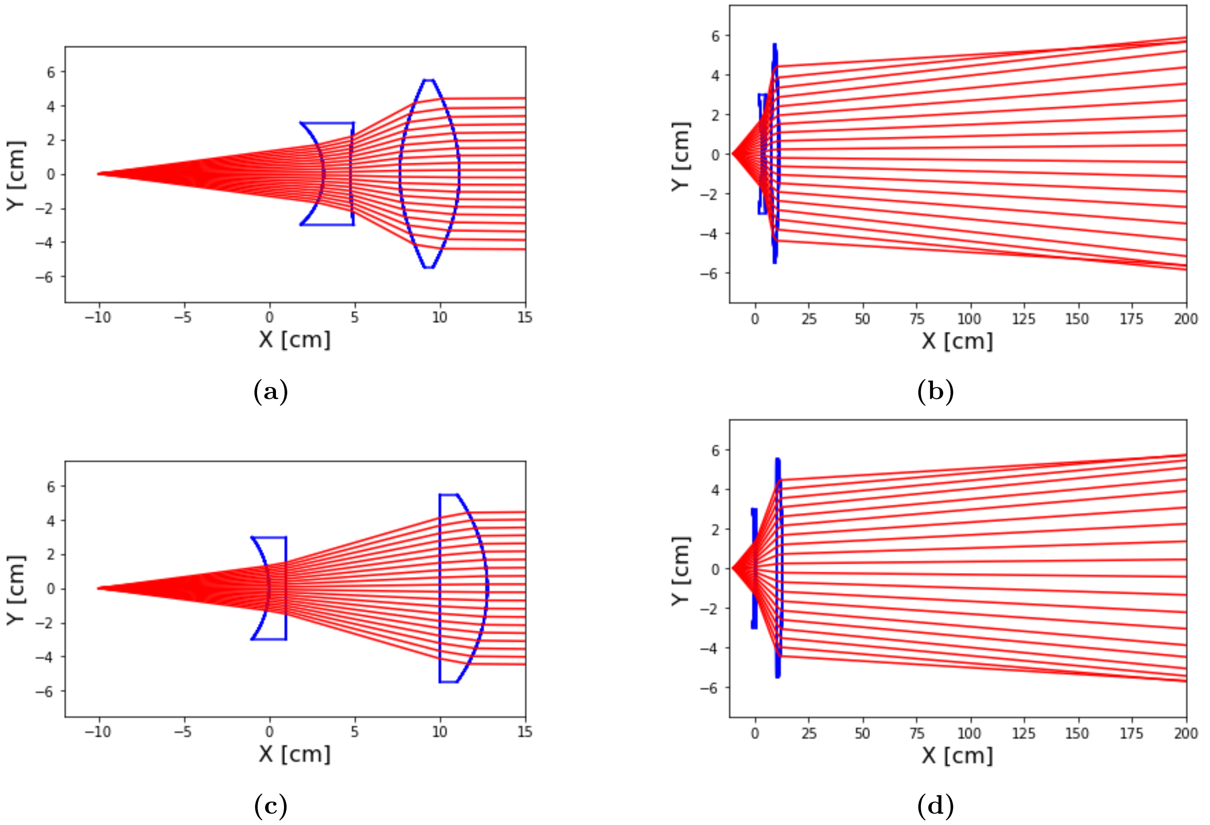
6.2.1 Lens Design

To get a bigger spot size a new lens system for collimating the emitted THz radiation was designed. The new lens setup consists of two lenses, a spreading lens and a bigger collimating lens. Both lenses are made from ultra high molecular weight polyethylene (UHMWPE), with a refractive index of $n = 1.5245$ in the THz region. To minimize aberration, originally two aspheric lenses were design. The general equation of am aspheric surface is given by

$$z(r) = \frac{r^2}{R\left(1+\sqrt{1-(1-\kappa)\frac{r^2}{R^2}}\right)} + \alpha_4 r^4 + \alpha_6 r^6 + \cdots , \tag{6.1}$$

where $x(r)$ is the displacement of the surface from the optical vertex plane, for an optical axis along $\hat{\mathbf{x}}$. R is the radius of curvature, κ the conic constant and α_n the n'th order coefficient.

To get the desired spread and collimating effects a 2d ray-tracer made for following refracted rays was used. See section 2.4.2 for details. In the ray-tracer each lens is defined by two surfaces, the front and back surface. A parameter sweep was then performed to get the optimal lens design. However, due to limitations in fabrication, we were forced to reconsider the design, and limit the lenses to simpler spherical or flat surfaces. The resulting lens designs, both spherical and aspherical, are shown in figure 6.8.



(a)

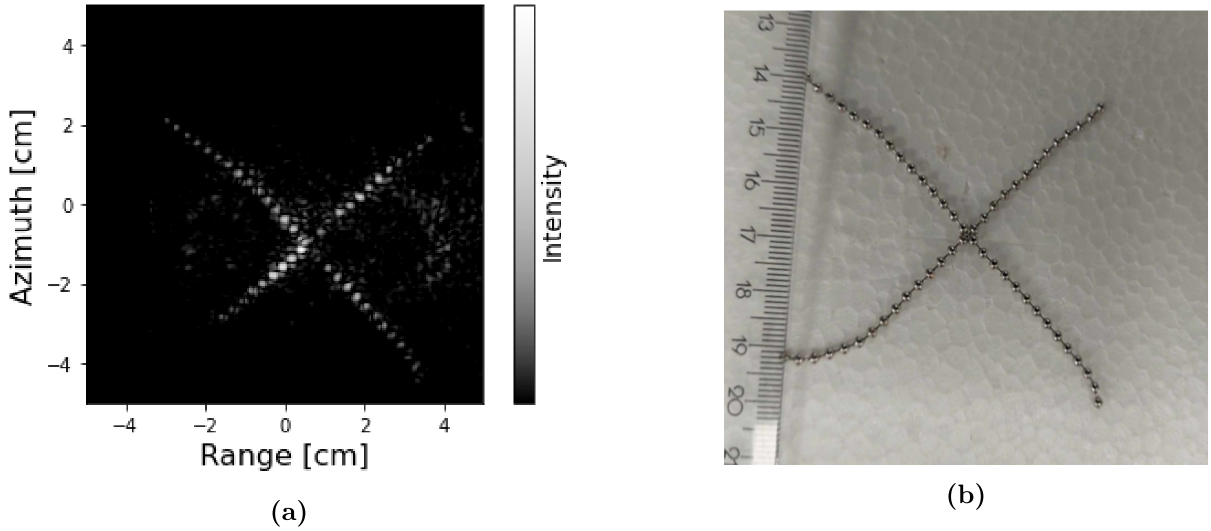
(b)

(c)

(d)

Figure 6.8. *The lenses are outlined in blue. Red rays are emitted from left to right. Aspheric lens setup **(a)** close up and **(b)** beam profile over a length of* $2m$*. Spherical lens setup **(c)** close up and **(d)** beam profile over a length of* $2m$*.*

Over the total length of $2m$ both setups have a tiny amount of spread, which is exaggerated by the uneven axes in the plots. With the new lenses, we get a spot size with a diameter of ≈ 8 cm. The result of a larger spot is shown in figure 6.9, where a much larger area of the target has been imaged as compared to figure 4.1b.



(a)

(b)

Figure 6.9. ***(a)*** *SAR image with the new lens setup of the scene presented in* ***(b)****.*

Here, almost all individual metal balls in the wires are shown in the SAR image. A consequence of the larger spot size is the lower intensity of the spot following the inverse-square law. This is also visible in figure 6.9a, where more background noise is visible compared to that in figure 4.1b, which was expected.

6.2.2 Large Spot Size Scenes

In this section the results of THz SAR with the large spot size are presented. The imaged scenes include some larger toy models as well as more complex scenes. Initial comparison of the two lens setups, show a decrease in the SNR when using the bigger lenses. This is expected, as the diameter of the spot size is double, the power of the THz radiation is distributed on an area four times the original size. However, even with the limitation we are still able to identify the model in the THz SAR images.

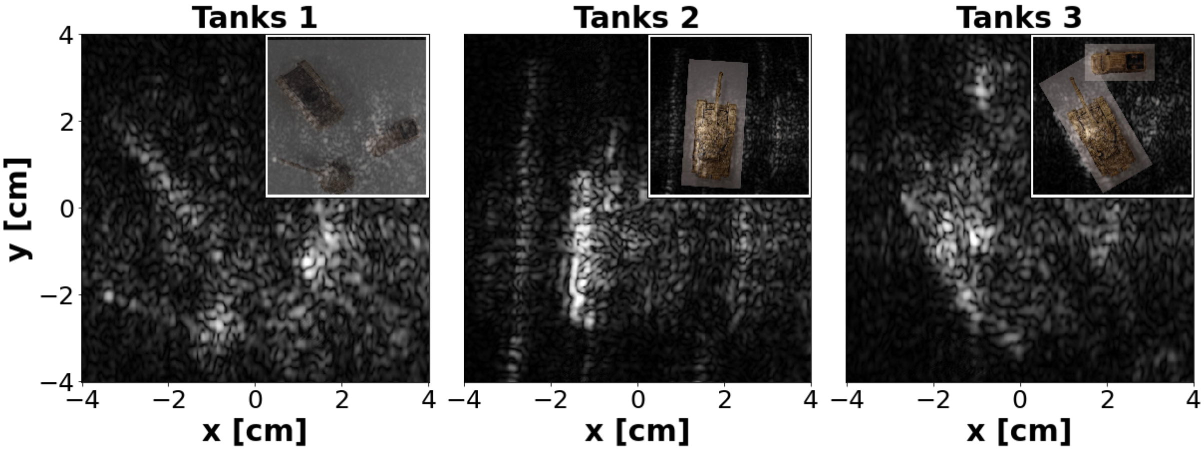


Figure 6.10. *THz SAR images of large scenes of the T72 model with other models and clutter.*

In the first figure, 6.10, the tank model is imaged in different scenarios and next to other vehicles. In tanks 1, three objects are imaged, the tank turret, tank body and the ford. The background noise is quite a bit higher on these images, which can make it hard to identify the reflective surfaces of the models. However it is still possible to make out the turret and body of the tank. The shape of the ford is impossible to make out on the SAR image, but its location is visible by the area of high intensity to the right of the tank body. In tanks 2, the tank is much easier to identify, measured from the side we see the tracks and the turret clearly among the clutter. In the last image, tanks 3, the body of the tank and front of the car show up quite well.

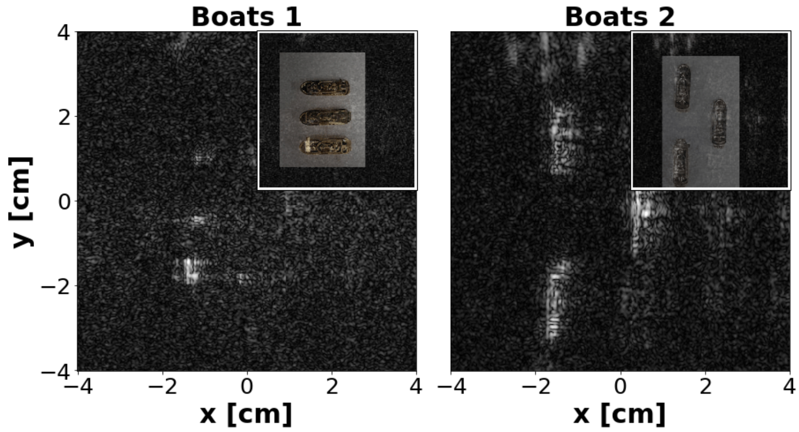


Figure 6.11. *THz SAR images of large scenes of different boat models.*

In figure 6.11, the THz imaging method is tested on three models with only slight variations, namely the addition of a antenna tower and a crane on the original boat model, see appendix A.3. In boats 1, the boats are measured front on, where the crane boat is identifiable by the large area of reflectivity at the front. The two other boats only reflect minimal radiation back in the front on configuration, with no clear differences between them. In boats 2, the boats are measured form the side, with no major differences which can be attributed to the crane or the antenna.

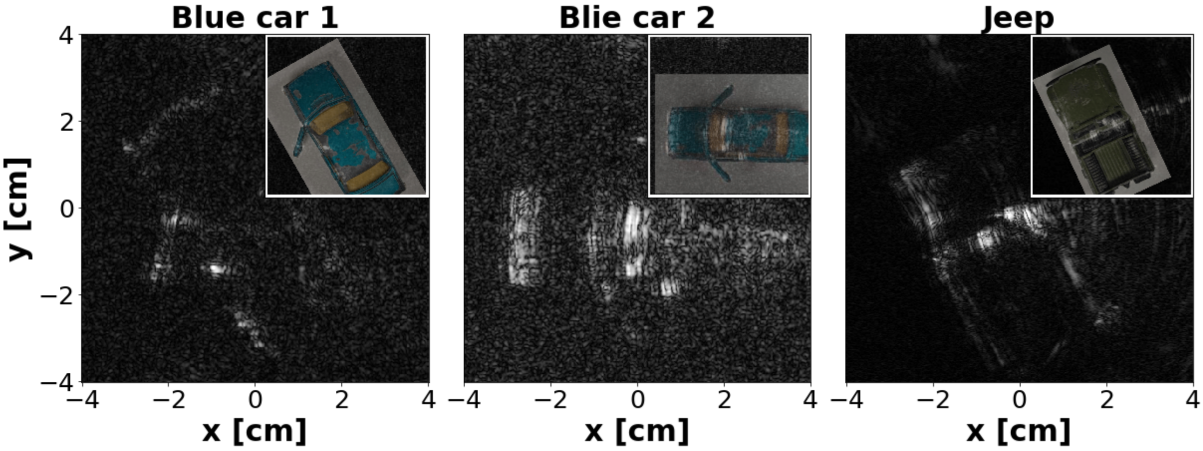


Figure 6.12. *THz SAR images of larger scenes of toy jeep and blue car.*

Lastly some larger toy models were measured with the large spot size. The resulting THz SAR images capture the models very well. The shape and orientation of the models are nicely shown in the SAR images. Most of the intensity comes from the edges of the model, with some highlights on features oriented perpendicular to the emitter/detector setup.

6.3 Reduced Scale THz SAR Discussion

The initial results of THz SAR are shown to capture the general size and shape of the reduced scale models well, along with bright reflective features, as is expected of SAR. When comparing the resulting THz SAR images with MSTAR, the THz images generally seem more specular. The same dominant features are however present in the THz SAR images. Additionally the implementation of nice MSTAR like clutter was hard to recreate in the THz region, making it hard to image the vehicle shadows.

Tests involving similar models with slight variations, figure 6.11, show it is possible to discern between some models, if the new feature is not located in shadow. The boat with the crane mounted at the front of the boat, was clearly identifiable in the front on THz SAR image. Results with the antenna, showed little to no difference when compared to just the boat. When measured from the front, the antenna is mainly in shadow and from the side, the surface area of the antenna is quite small. Both effects impact the amount of reflected radiation form the antenna and drowns the feature in the noise.

A similar method with a THz ISAR setup for measuring reduced scale vehicles has been tested by Danylov et al. [82]. The resulting resolution and image quality is comparable to what was achieved in this report. The main advantage of our setup is the comparable simplicity for generating THz radiation, lowering the barrier of entry into THz ISAR. Other results with THz reduced scale, show good agreement between the THz response and true radar for radar cross section measurements [83]. Additionally, comparison of THz SAR and Optix simulation also show good agreement, further validating reduced scale THz SAR for modeling radar signals.

The current limitation of imaging area is related to the beam spot size. By making new larger lenses the energy pr. area is reduced, reducing the SNR. For larger spot sizes, a more powerful THz emitter is needed along with new collimating lenses or a diverging setup. The most powerful THz sources are currently based on vacuum electronic sources[84]. Another approach involve generating THz radiation by up-converting the signal from a vector network analyzer by a frequency extender, reaching frequencies of 200-500 GHz [21, 23].

The new lens system introduced a lot of noise along the range direction. These artifacts in range are believed to be related to THz radiation bouncing around inside of the lens setup before leaving the lens setup towards the target.

6.4 Summary

The results presented above demonstrates the ability to capture real SAR data in a lab setting, in this case by utilizing THz radiation with much shorter wavelengths as compared to traditional radar. The THz SAR images are shown to accurately capture the size and features on the models. The setup is shown to produce good scale representations of true radar wave scattering and detection, when compared with true scale radar simulations.

By enabling real data accusation on custom models in a lab setting, the data can easily and effectively be utilized in computer simulation validation. For most scenes, new models can be fabricated and measured in the span of 24 hours, depending on the measuring parameters. This allows for fast testing, which is otherwise hard to come by.

SAR Techniques 7

In this section different SAR techniques are presented, including polarimetry and interferometry. In all cases simulations and experimental measurements are pursued. Theory and initial simulations on hyperspectral SAR is shown in appendix B.

7.1 Polarimetry

This section will contain polarimetry simulations at radar frequencies as well as THz polSAR measurements. Initially we will go through the polarimetry simulations to validate the simulation method. We then follow up with measurements acquired from the THz setup and the formatting of THz SAR.

7.1.1 Initial Ray Tracing Simulations

In the following section a refractive index of $n = 300 + i300$ was used, as this roughly correlates to metals in the THz region. Specifically the refractive index of gold at 1.2 THz is $n = 345 + i379$[85]. For lower frequencies, the effective refractive index of metals tends to increase. Increasing the refractive index past the current one, has little impact on the Fresnel amplitude coefficients, and even with the current refractive index the fresnel amplitude coefficients from air to metal are essentially constant, see figure 7.1.

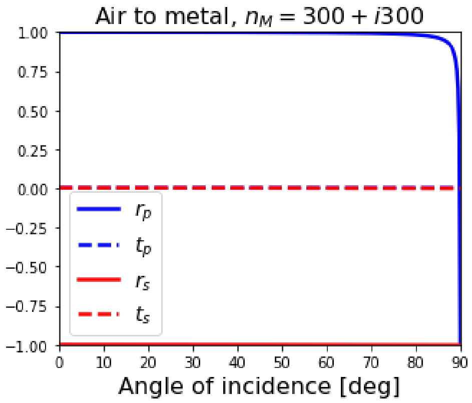


Figure 7.1. *Fresnel amplitude coefficients of air to metal interface with a refractive index of* $n = 300 + i300$.

The method for simulation polSAR images, as described in section 2.4.4, was originally tested on a box with side lengths of 3 µm. The resulting SAR image of HH polarization is shown in figure 7.2 for different multilooks N.

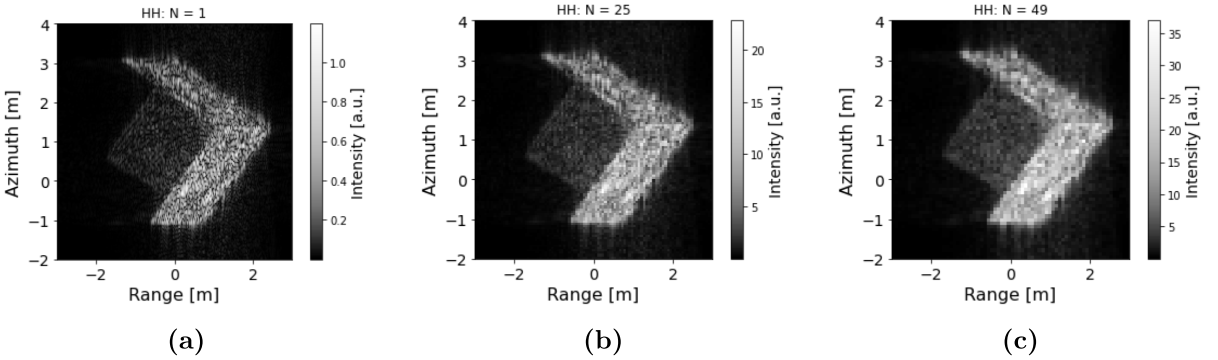


Figure 7.2. *HH polSAR image with different number of look in multilook* ***(a)*** $N = 1$, ***(b)*** $N = 25$ *and* ***(c)*** $N = 49$.

By increasing the number of looks, the speckle of the SAR image is reduced, as expected. The benefit of which is more defined faces on the box. For $N = n^2$, a $n \times n$ look window is averaged for each new pixel. For $N = 7$ the effect on image resolution is becoming quite clear, with limited improvements in speckle noise as compared to $N = 25$. Moving forward multilook with $N = 25$ are used on the SAR images.

The box was oriented in such as way as to showcase the ability of polSAR to identify the orientation of faces in SAR images. The primary output of a polSAR simulation is 3 SAR images, one for each polarization, HH, HV and VV. The resulting images with multilook are shown in figure 7.3.

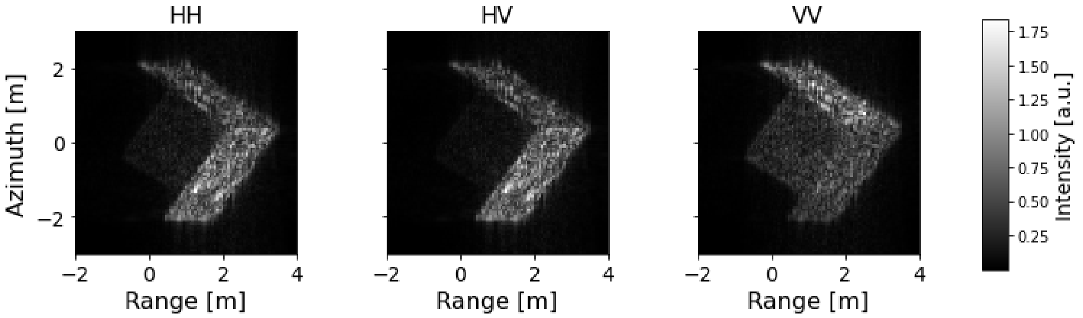


Figure 7.3. *SAR images of a* $3 \times 3 \times 3$ *m box, measured at different polarizations, HH, HV and VV with* $N = 25$ *looks.*

All 3 polarizations capture the expected SAR image of a box, but the intensity of the different faces of the box change depending on the polarization. To format the final polSAR image, each SAR image is given a color. In the following polSAR images the polarization channels have been colored according to,

$$HH \rightarrow blue, \qquad HV \rightarrow green, \qquad VV \rightarrow red. \tag{7.1}$$

The above definition is used in a all polSAR images presented during the project. The individual color channels are shown in figure 7.4. The final polSAR image of the box, can seen in figure 7.5.

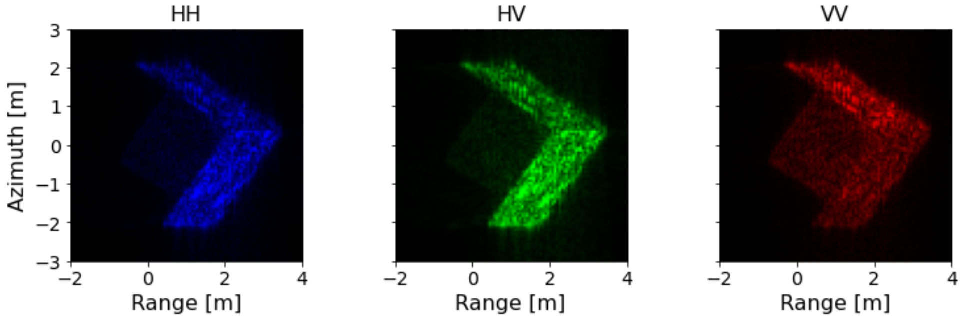


Figure 7.4. *SAR images of a $3 \times 3 \times 3$ m box, measured at different polarizations, HH, HV and VV. The SAR images are colored according to 7.1.*

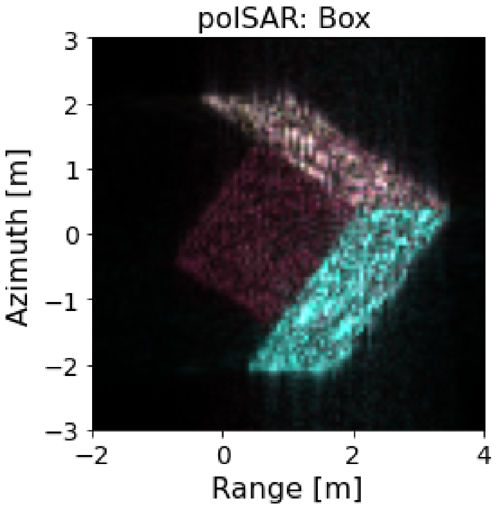


Figure 7.5. *Final polSAR images of the box.*

Being able to identify the orientation of certain features in a SAR image is one of the many advantages polSAR has on ordinary SAR. In the polSAR image the different faces of the box are easily distinguishable by their color. To showcase this on a larger scene, a collection of houses situated in a small village was modeled in blender and simulated. The scene is shown in figure 7.6. The HH polarization SAR image is shown in figure 7.7 along with the final polSAR image.

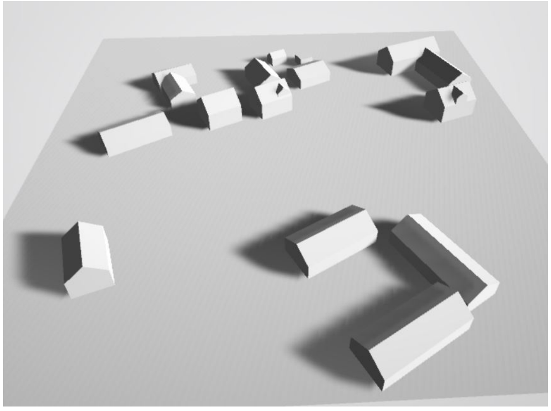


Figure 7.6. *Small scale model of a village.*

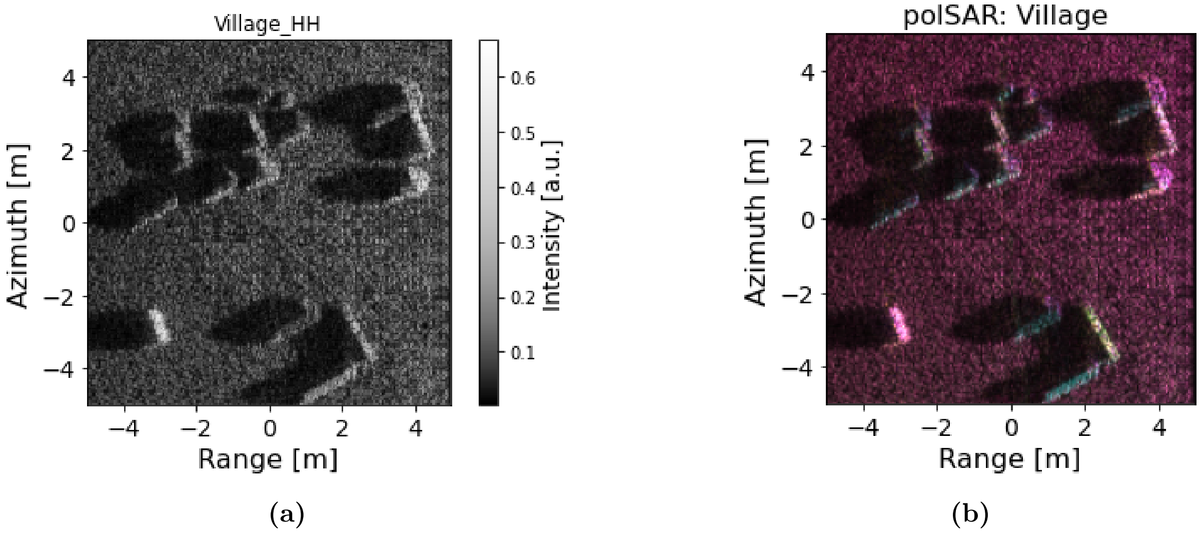


Figure 7.7. *HH polarization SAR image* ***(a)*** *and polSAR image* ***(b)*** *of the village scene measured at a slant angle of* $\theta = 25°$.

The resulting polSAR image makes it easy to identify the pink roofs from the cyan illuminated facades of houses in the village scene. With the houses being quite short compared to the box, the straight faces appear quite a bit thinner in the village simulation.

7.1.2 THz Polarimetry

With modifications presented in section 3.3, the following four polarizations have been measured with the THz setup, HH, HV, VH and VV. The results of each polarization for clutter is presented in figure 7.8.

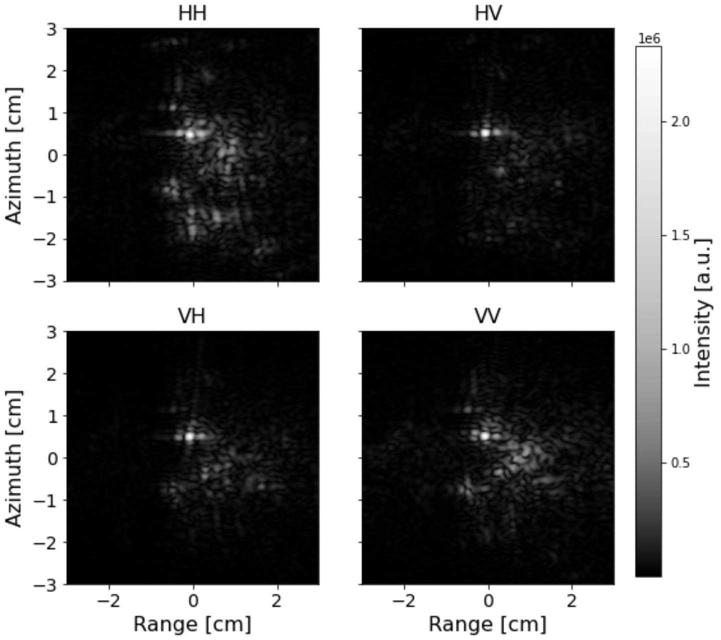
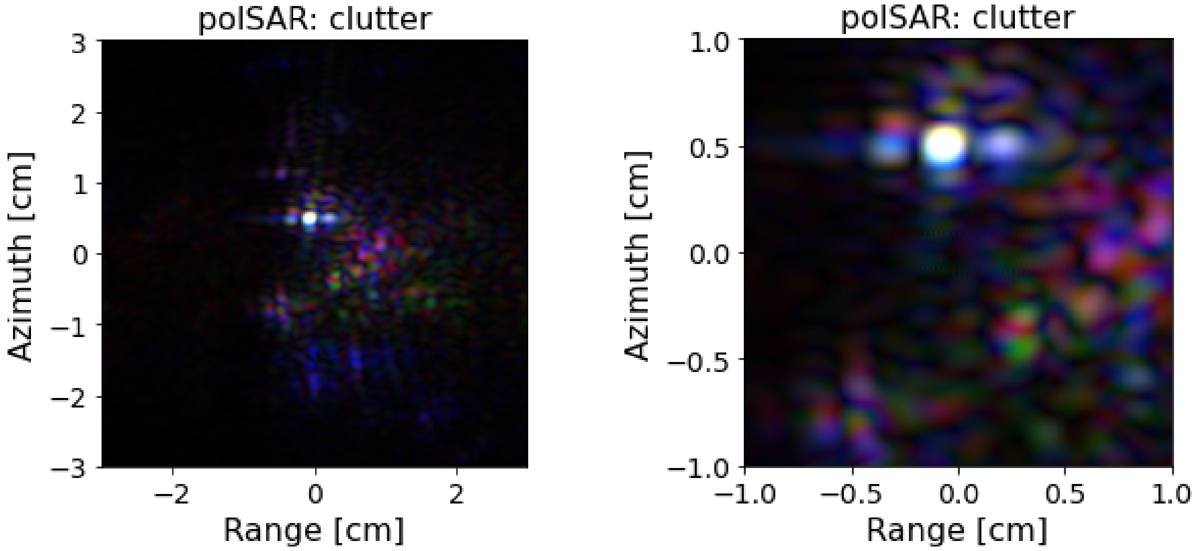


Figure 7.8. *THz SAR images of clutter, measured for the four different polarizations.*

A bright reflector can be seen in all polSAR images, but depending on the polarization different areas of the clutter are observed. It is worth noting that some cross polarization is expected due to the bowtie structure of the THz antennas. But even so, each polarization captures different scatterers. Furthermore the THz setup is a bistatic setup with a small distance from the emitter to the detector, thereby breaking reciprocity, which is also seen in the differences from HV and VH. To format the polSAR image, see figure 7.9, let VH be the green color channel.



(a) (b)

Figure 7.9. *THz polSAR image of clutter **(a)** at full size **(b)** and cropped to show the area of most intensity.*

It is worth noticing that, there is some slight misalignment between the different polarizations, which is mostly visible on small scatterers. The displacement is under 1 mm, measured over a total travel distance of ≈ 2 m. Image alignment is a definite next step in trying to combat the displacement, which is not pursued in this project.

Next a scene consisting of the car model placed on clutter, was measured with a larger integration angle. All four polarizations are shown in figure 7.10.

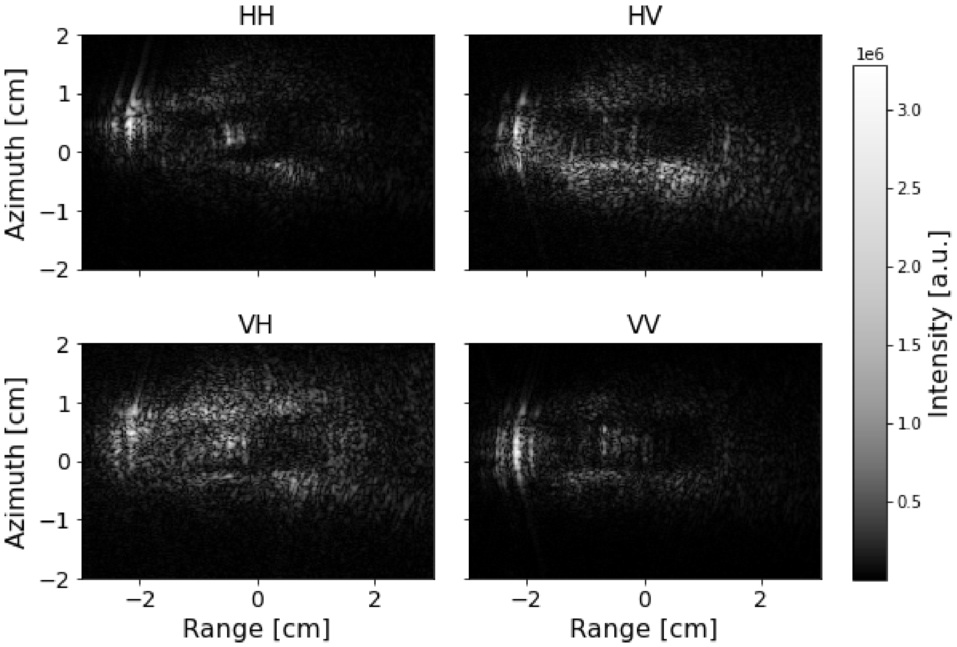


Figure 7.10. *THz SAR images of car on clutter, measured for the four different polarizations*

The car is oriented with the front towards the emitter, resulting in the high intensity at the front of the SAR images. The resulting polSAR image is shown in figure 7.11, with respectively HV and VH used for the green color channel.

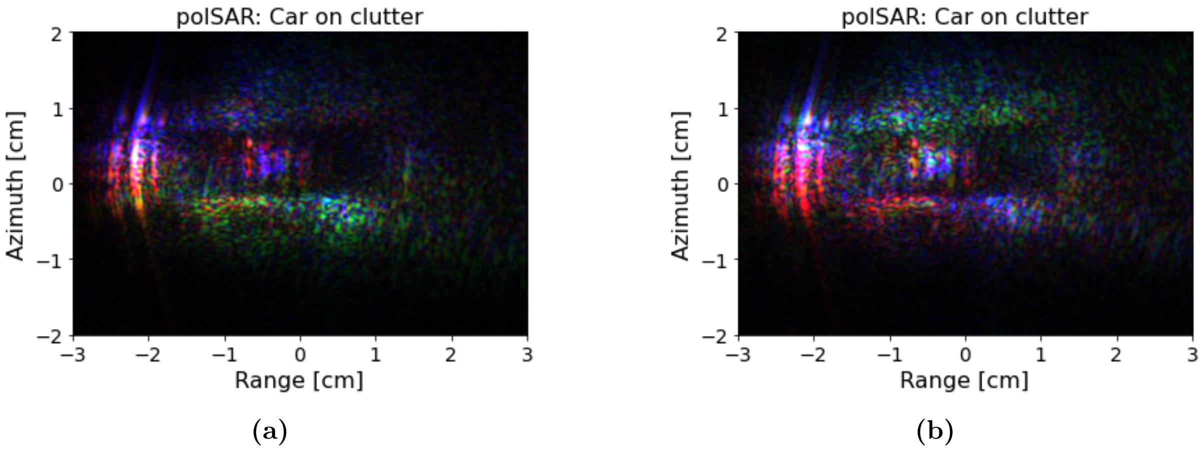


Figure 7.11. *THz polSAR image of car on clutter **(a)** HV → green**(b)** VH → green.*

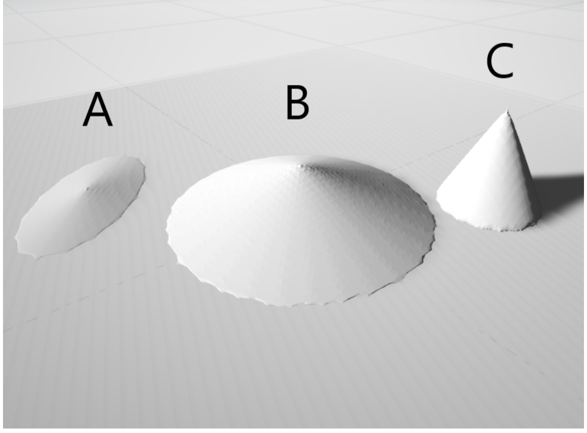
The resulting polSAR image of the car on clutter has a very distinct shadow, which was missing on many of the experiments in section 6. Studying the different polarizations, it becomes clear that the HV and VH polarizations reflect much more intensity back from the clutter, relative to the model. As all the reduced scale models are captured in either HH or VV, it might be possible to better capture the clutter in the HV or VH polarization instead.

7.2 Interferometric SAR

This section will contain interferometry simulations at THz and radar frequencies as well as THz InSAR measurements. Initially we will go through the interferometry simulations at radar frequencies to validate the simulation method. We then follow up with measurements of THz SAR.

7.2.1 Simulated Interferometry

To validate the interferometric SAR (inSAR) data processing described in section 3.7 three different cones (A,B,C) with varying height and slope were meshed on top of a plane and simulated in the Optix ray-tracer, see figure 7.12a. The mesh was modeled like clutter as described in section 2.4.3. Relevant SAR parameters for the simulations can be found in table 7.12b.



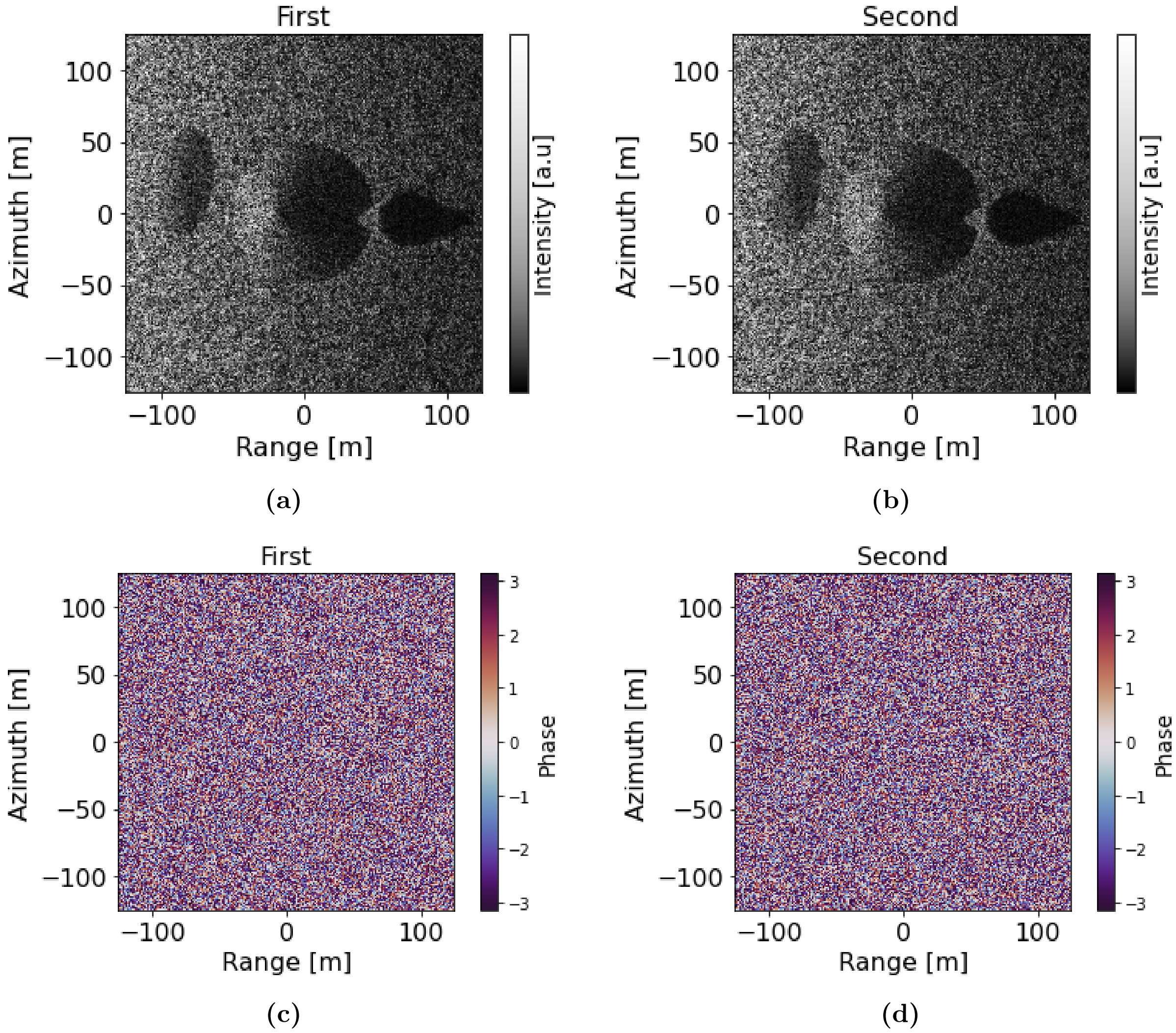
(a)

Simulation parameters	Values
Center frequency (f_c)	9.6 GHz
Bandwidth (B)	0.2 GHz
Range resolution	0.75 m
Azimuth resolution	0.90 m
Standoff range	720 m
Integration angle	1°
Azimuth steps	300
Slant angles	45° and 45.2°

(b)

Figure 7.12. ***(a)*** *Image of the scene used for inSAR, with the indexing of each cone by A,B,C.* ***(b)*** *Table of relevant simulation parameters.*

Two simulations were carried out with slant angles of 45° which will be labeled as the first image and 45.2° which is the second image. The resulting SAR images and phase are shown in figure 7.13. For cones with higher slant angles, there are increased effects of layover and foreshortening in the SAR images. The resulting SAR images are almost identical due to the very small change in slant angle between the two images. But as we will show below the difference is enough do interferometry and get height information of the measured model.



(a)

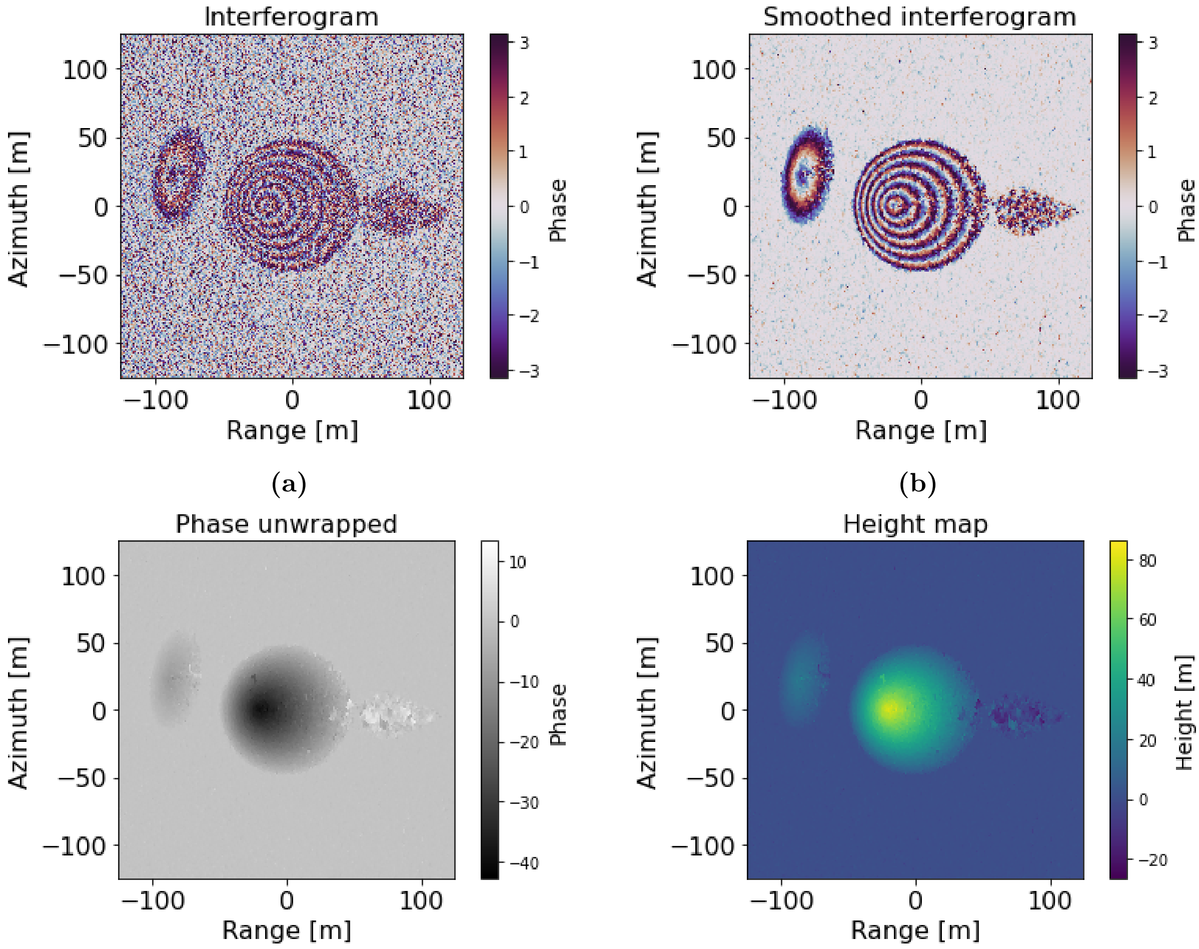
(b)

(c)

(d)

Figure 7.13. ***(a-b)*** *complex amplitude SAR images of the first and second image.* ***(c-d)*** *complex phase SAR images of the first and second image.*

Individually the phase images are essentially random, but by taking the interferogram, see figure 7.14a, the cones become visible by the fringes. The frequency and number of fringes is related to the slope and height of the measured targets. By a simple smoothing, see figure 7.14b, the noise in the interferogram is reduced substantially, which allows for better phase unwrapping. The unwrapped phase image is shown in figure 7.14c. Note how cone A was badly unwrapped due to its high slant angle. Through equation 2.36, the height map can then be calculated, see figure 7.14d. The resulting heights for cone A and B are then $H'_A \approx 20$ m and $H'_B \approx 85$ m given by the height map. In the mesh the actual heights of the cones are only $H_A = 4$ m, $H_B = 20$ m and $H_C = 40$ m. This discrepancy stems from the incorrect z-coordinates of the pixels used in BPA when formatting the first and second image for interferometry. Initially it is assumed that all scatterers in the image stem from a plane, which is incorrect. To fix this we need to construct a digital elevation map of our scene, as outlined in the flow diagram 3.19. The resulting height map of the DEM is shown in figure 7.15a, with much more accurate height information for cone A and cone B. The true height information of the mesh used for testing can be found in figure 7.15b. Additionally the peak of cone A is also correctly placed in the DEM height map. As the iterative method for creating a DEM is running, the location of the peak can slowly be seen moving into the correct position by constructing the digital elevation map used in image formatting.



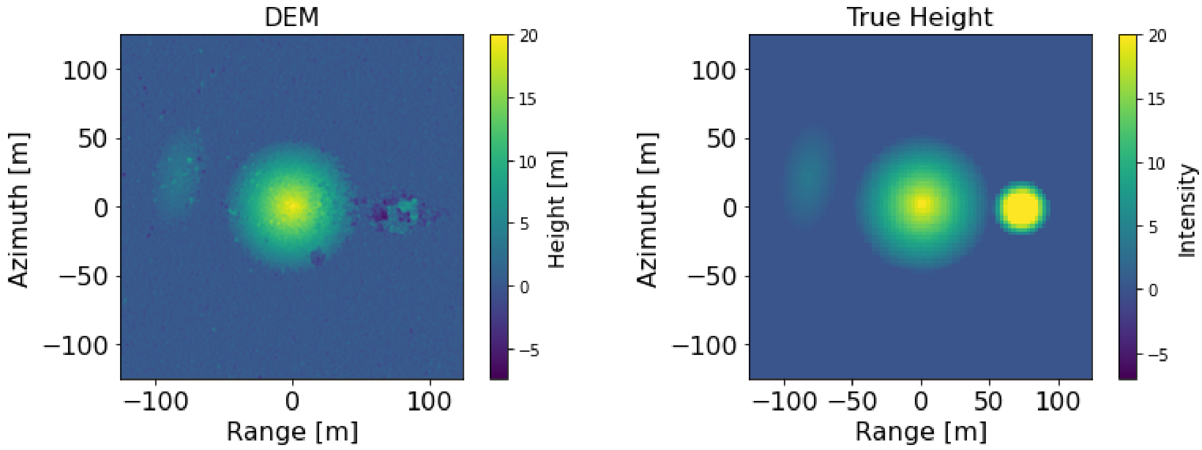
(a)

(b)

(c)

(d)

Figure 7.14. ***(a)*** *SAR interferogram based on the first and second SAR images in figure 7.13.* ***(b)*** *Smoothed interferogram by averaging.* ***(c)*** *Unwrapped phase plot of smoothed interferogram.* ***(d)*** *Height map of the scene.*



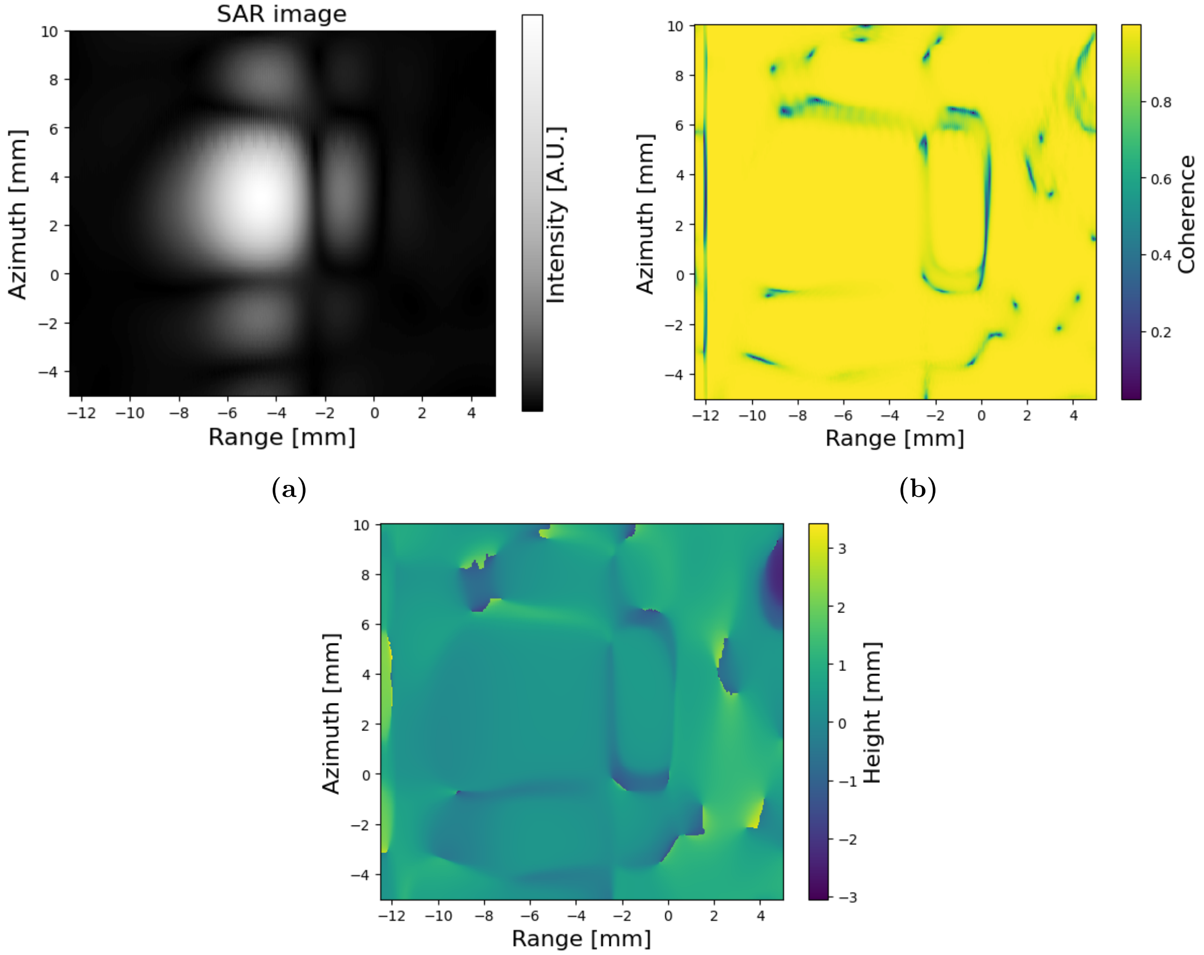
(a)

(b)

Figure 7.15. ***(a)*** *Height map of cones after DEM.* ***(b)*** *True height map of the cone mesh.*

7.2.2 THz InSAR

With the addition of the linear stepper motor to control the slant of the THz SAR setup, it is possible to apply the InSAR technique. However, when examining equation 2.36 a problem becomes apparent. Inserting values relevant to our setup into the equation, we can calculate the change in height across a single fringe. Using a center frequency of 1 THz, slant of 10° and 10.5° and $\Psi = 2\pi$, we get a height difference of ≈ 3.5 cm. This is about the same the size as the spot. This shows that for the change in heights relevant to this setup ≈ 2 cm, there will little no discernible differences in the phase. This problem is clearly seen in figure 7.16c, showing the InSAR image of a Ø5 mm ball bearing. Here, the change in height at the edge of the ball is seen to be 0.3 mm. This can be true as only parts of the ball is illuminated at certain slant angle, and thus we only see a fraction of the actual height of the ball. However, this change is drowned out by the surrounding phase noise from low coherence spots.



(a)

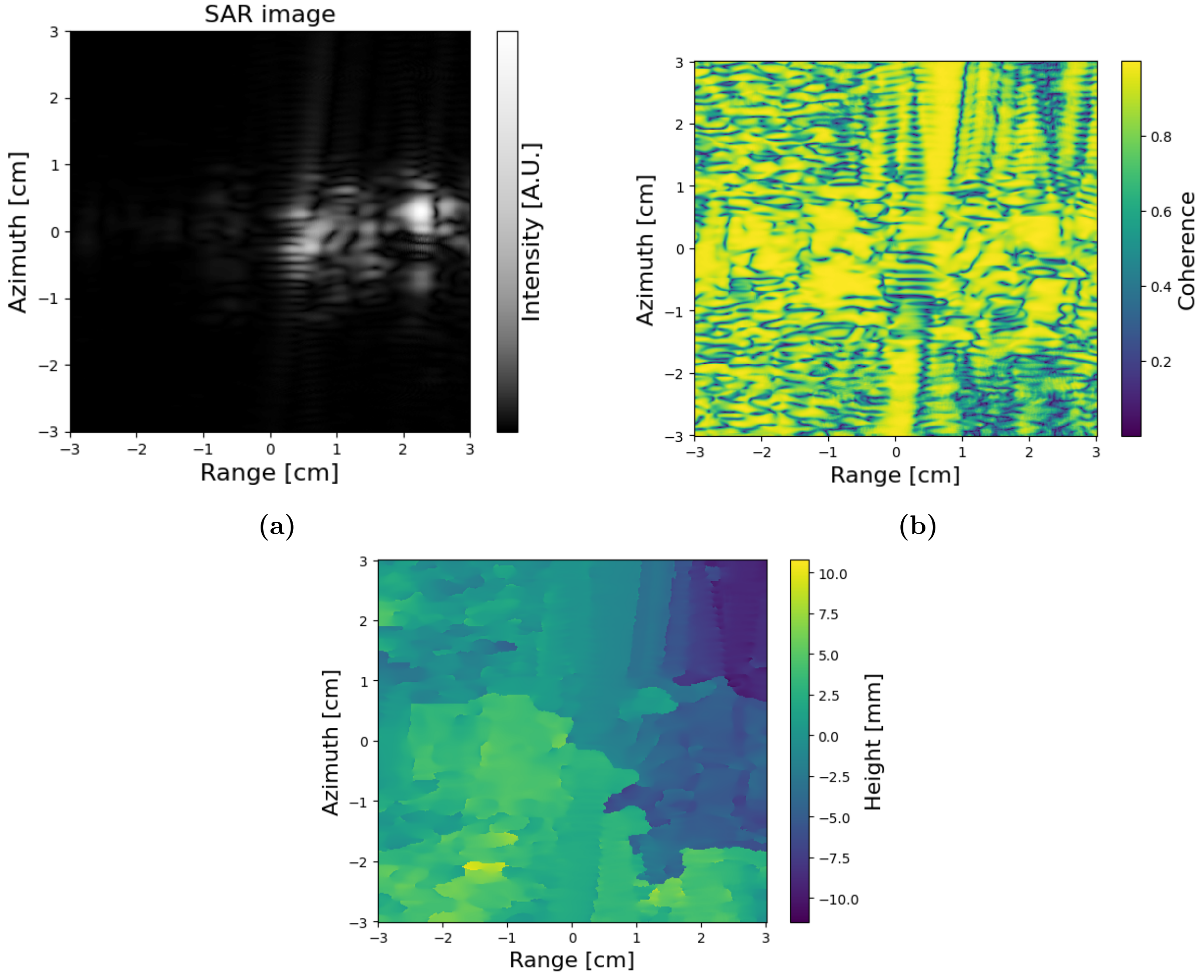
(b)

(c)

Figure 7.16. ***(a)*** *SAR image of Ø5* mm *ball bearing.* ***(b)*** *Coherence map of the two SAR images of the ball.* ***(c)*** *InSAR image of the ball.*

An important factor is the coherence of the image, calculated by equation 2.39. The coherence of two images can at maximum be 1, indicating a perfect match in pixels of the SAR images. At 0 or near 0 values there is no coherence. For unwrapping the coherence need to be high. This is also seen in figure 7.16b, where most of the image exhibits high coherence.

Another issue for InSAR images is how well the object is imaged. If the SAR image has low intensity or contains artifacts, it will affect the coherence, and thus the unwrapping. Figure 7.17a shows a poor SAR image of the blue toy car. This is immediately seen in the coherence map of the car, figure 7.17b, where less of the scene shows high coherence, compared to that of the ball. We can identify an artifact of the image in both the SAR image and the coherence map, which is an almost vertical fringe located at a range of [0; 1] cm. The poor image quality yields a poor InSAR image as seen in figure 7.17c.



(a)

(b)

(c)

Figure 7.17. *(a) SAR image of the blue toy car. (b) Coherence map of the two SAR images of the car. (c) InSAR image of the car.*

Figure 7.17c also suffers the same issue as the ball, which is the low height difference compared to phase noise. Here, the maximum and minimum of the height is located off the target and is attributed to the noise.

7.3 Discussion of SAR Techniques

7.3.1 Polarimetry

The results of the polSAR simulations nicely capture the polarization response from different surface orientations, which allows for easy detection of surface orientations in the final SAR image. The polSAR implementation is very simple, which limits the use cases to domains dominated by double bounces or surface scattering. No effort was put in to model volume scattering. However, In the limited use cases the model produces nice polSAR images.

The initial experiment with THZ-SAR polarimetry are promising. By utilizing the photo conductive antennas we are able to capture and map the reflection polarizations response of different pars of the scene. The degree of linear polarization from the bowtie antennas was shown to be sufficient for capturing and distinguishing the different polarizations. Current THz polarimetry setups mainly rely on scanning the subject on an xy-stage and measuring either reflection or transmission of each polarization. To measure the polarization, either polarizers or polarization sensitive antennas are often used. These setups usually measure with a high resolution of $200 - 50$ µm. Increasing the resolution either increases the acquisition time or decreases the image area.[86, 87] For large area images with high resolution, even with the current setup, polSAR has the time advantage. And with faster scan technologies, such as a time domain system, polSAR could conceivably be the faster option as a single image consists of 100 to 500 time domain signal measurements, compared to several thousands steps needed with an xy-stage.

With the current THz polarimetry setup, we are able to measure and map surfaces and their polarimetric response from a distance of 80 cm, with prospects in non-destructive sensing and characterization.

7.3.2 Interferometry

The InSAR simulations illustrate the need for orthorectification in which after the postprocessing steps the InSAR images match the initial mesh in both height and position thereof. However, cone C from figure 7.12a is not unwrapped correctly, as its slope is too steep for unwrapping. This illustrates the limitation of InSAR, as the phase change from too large changes in the height can not be resolved.

The experimental approach to InSAR was unsuccessful in producing InSAR images. This is due to the low change in height across fringe of 2π, and the low reflected intensity from the object. The low reflected intensity essentially makes it difficult to achieve high coherence of the SAR images. When a high coherence is achieved the fringe to height ratio then serves as another limitation in calculating the actual height difference. However, the InSAR method should be possible. Yang et al. [88] presented an InSAR experimental setup based on horn antennas, which accurately could measure the height of different corner-reflectors. With a successful InSAR setup the next step could be tomoSAR, which requires numerous images with different baselines, and allows for the ability to create high quality 3D representations of objects. In lab settings these multiple baselines are acquired by a planar aperture in a square grid[20–23] or a cylindrical plane aperture[22], based by whether a SAR or an ISAR setup is used. This method can be utilized for non-destructive testing in security applications for package and person inspection.

7.4 Summary

Both polSAR and InSAR have been investigated experimentally and through ray-tracing simulations. Simulations showed the essence of both methods, with polSAR enabling the capture of polarization specific effects, and InSAR being able to create DSMs of SAR images. PolSAR was also successfully implemented into the lab setup, where THz polSAR images were made. However, THz InSAR did not yield results of sufficient quality due to system restrictions.

Conclusion 8

This project was divided into four main parts: THz SAR microscopy, SAR simulations, reduced scale SAR measurements, and SAR techniques. All have been thoroughly examined for their strengths and limitations.

In THz SAR microscopy we achieved spatial resolutions near the diffraction limit around 300 µm. This was determined by examining the 3dB width of the range and azimuth cross sections of the rods on the microscopy targets. This is on par with current THz SAR literature and other THz imaging methods. However, the measuring time and SNR is relatively low. This would be variables of interest for further development. Additionally, the polSAR method could be applied to THz SAR microscopy to allow for polarization specific measurements on small targets, for applications in the security sector or for non-destructive testing.

Ray-tracing simulations for radar were optimized to simulate complex meshes of vehicles within a reasonable time frame. The computational time of the CPU ray-tracer was reduced by a factor of ×1300 compared to a previous project. The implementation of GPU support in the simulations using Nvidia optix further reduced computational time. Simulations comparable to real SAR images from the MSTAR dataset were achieved. The resulting simulation time for a single SAR image was ∼30s on non-optimized hardware and hence opens the possibility for generation of large data sets. The simulation engine is not optimized for use in CNN model training, but could possibly in the future be improved for this.

In reduced scale THz SAR two lens setups have been tested both producing fine quality SAR images of reduced scale objects such as tanks, cars and boats. This served as an easy and fast validation method for SAR simulations, due to the simple process of 3D printing and gold coating 3D models used in simulations. Both the boat and the pickup truck were in good accordance with the simulations after the implementation of noise in simulations. With the implementation of the larger lenses, the spot size increased, at the cost of average intensity in the scene. This allowed for the imaging of larger scenes and objects, at the cost of longer measuring times and lower SNR. For further development, other THz systems can be examined for the use in THz SAR, to either optimize the SNR or the measuring time of the setup. This could be electronic sources such as vector network analyzers combined with horn antennas, or pulsed THz-TDS systems.

With a relatively simple ray-tracing polarization model, we were able to simulate polarimetry SAR images of metallic scenes. The simulations lack support for volumetric scattering, limiting the use case to domains dominated by double bounding and surface scattering. By utilizing polarization sensitive photo conductive antennas, polarimetric THz SAR was preformed. The resulting polarimetric SAR images map the reflections of different polarizations of a scene. The degree of linear polarization from the bowtie antennas was shown to be adequate in measuring the different polarization responses.

Ray-tracing based simulations of InSAR were performed and used in the iterative process of constructing a digital elevation map with good similarity with the original topology of the measured scene. Additionally THz InSAR was tried, but due to the limitations of our setup, particularly the spot size limiting the illuminated area, it was hard to measure good fringes in the interferogram. The result was poorly unwrapped images. Better experimental results for height mapping of scenes would require using the method of tomoSAR, which requires additional measurements.

Bibliography

[1] Van Rossum, G. and Drake Jr, F. L. *Python reference manual.* Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[2] Parker, S. G. et al. "Optix: a general purpose ray tracing engine". In: *Acm transactions on graphics (tog)* 29.4 (2010), pp. 1–13.

[3] Sarkar, T. K., Salazar Palma, M. and Mokole, E. L. "Echoing Across the Years: A History of Early Radar Evolution". In: *IEEE Microwave Magazine* 17.10 (2016), pp. 46–60. DOI: `10.1109/MMM.2016.2589200`.

[4] Bhatta, N. and .M, G. P. "RADAR and its applications". In: 10 (Jan. 2017), pp. 1–9.

[5] Hull, A. W. "The magnetron". In: *Journal of the American Institute of Electrical Engineers* 40.9 (1921), pp. 715–723. DOI: `10.1109/JoAIEE.1921.6594005`.

[6] Doviak, R., Zrnic, D. and Sirmans, D. "Doppler weather radar". In: *Proceedings of the IEEE* 67.11 (1979), pp. 1522–1553. DOI: `10.1109/PROC.1979.11511`.

[7] Muehe, C. E. et al. "New techniques applied to air-traffic control radars". In: *Proceedings of the IEEE* 62.6 (1974), pp. 716–723.

[8] Ender, J. et al. "Radar techniques for space situational awareness". In: *2011 12th International Radar Symposium (IRS).* 2011, pp. 21–26.

[9] Richards, M. A. et al. *Principles of Modern Radar: Basic Principles.* Vol. 1. Citeseer, 2010.

[10] Griffiths, D. J. *Introduction to electrodynamics.* Cambridge University Press, 2023.

[11] Richards, M. A. et al. *Principles of Modern Radar: Advanced Techniques.* Vol. 2. IET, 2012.

[12] Van Zyl, J. J. *Synthetic aperture radar polarimetry.* John Wiley & Sons, 2011.

[13] Fornaro, G. "Tomographic sar". In: *NATO SET-191 Lecture Series, Bucharest* (2014), pp. 10–11.

[14] Duersch, M. I. *Backprojection for synthetic aperture radar.* Brigham Young University, 2013.

[15] Ashry, M. M., Mashaly, A. S. and Sheta, B. I. "Improved SAR Range Doppler Algorithm Based on The Stretch Processing Architecture". In: *2022 International Telecommunications Conference (ITC-Egypt).* 2022, pp. 1–6. DOI: `10.1109/ITC-Egypt55520.2022.9855729`.

[16] Li, Z., Wang, J. and Liu, Q. H. “Frequency-Domain Backprojection Algorithm for Synthetic Aperture Radar Imaging”. In: *IEEE Geoscience and Remote Sensing Letters* 12.4 (2015), pp. 905–909. DOI: 10.1109/LGRS.2014.2366156.

[17] Ma, Z. T. et al. “Modulators for Terahertz Communication: The Current State of the Art”. In: *Research (Wash D C)* (May 2019).

[18] Zhang, J., Li, S. and Le, W. “Advances of terahertz technology in neuroscience: Current status and a future perspective”. In: *iScience* 24.12 (Dec. 2021).

[19] Yi, L. et al. “Ultra-Wideband Frequency Modulated Continuous Wave Photonic Radar System for Three-Dimensional Terahertz Synthetic Aperture Radar Imaging”. In: *Journal of Lightwave Technology* 40.20 (2022), pp. 6719–6728. DOI: 10.1109/JLT.2022.3193397.

[20] Damyanov, D. et al. “3D THz-TDS SAR Imaging by an Inverse Synthetic Cylindrical Aperture”. In: *IEEE Access* 11 (2023), pp. 9680–9690. DOI: 10.1109/ACCESS.2023.3240101.

[21] Batra, A. et al. “Sub-mm Resolution 3D SAR Imaging at 1.5 THz”. In: *2021 Fourth International Workshop on Mobile Terahertz Systems (IWMTS)*. 2021, pp. 1–5. DOI: 10.1109/IWMTS51331.2021.9486780.

[22] Batra, A. et al. “Short-Range SAR Imaging From GHz to THz Waves”. In: *IEEE Journal of Microwaves* 1.2 (2021), pp. 574–585. DOI: 10.1109/JMW.2021.3063343.

[23] Batra, A. et al. “Object Recognition in High-Resolution Indoor THz SAR Mapped Environment”. In: *Sensors* 22.10 (2022). ISSN: 1424-8220. DOI: 10.3390/s22103762. URL: https://www.mdpi.com/1424-8220/22/10/3762.

[24] Huang, Y., Shen, Y. and Wang, J. “From Terahertz Imaging to Terahertz Wireless Communications”. In: *Engineering* 22 (2023), pp. 106–124. ISSN: 2095-8099. DOI: https://doi.org/10.1016/j.eng.2022.06.023. URL: https://www.sciencedirect.com/science/article/pii/S2095809922006361.

[25] Es-Saqy, A. et al. “Terahertz VCO Design for High-Speed Wireless Communication Systems”. In: *Terahertz Wireless Communication Components and System Technologies*. Ed. by El Ghzaoui, M. et al. Singapore: Springer Singapore, 2022, pp. 1–16. ISBN: 978-981-16-9182-9. DOI: 10.1007/978-981-16-9182-9_1. URL: https://doi.org/10.1007/978-981-16-9182-9_1.

[26] Ikeda, T. et al. “Investigation of inflammable liquids by terahertz spectroscopy”. In: *Applied Physics Letters* 87.3 (2005), pp. 034105-1-034105–3. DOI: https://doi.org/10.1063/1.1999847.

[27] Yamamoto, K. et al. “Noninvasive Inspection of C-4 Explosive in Mails by Terahertz Time-Domain Spectroscopy”. In: *Japanese Journal of Applied Physics* 43.3B (Mar. 2004), p. L414. DOI: 10.1143/JJAP.43.L414. URL: https://dx.doi.org/10.1143/JJAP.43.L414.

[28] Federici, J. F. et al. “THz imaging and sensing for security applications—explosives, weapons and drugs”. In: *Semiconductor Science and Technology* 20.7 (June 2005).

[29] Baxter, J. B. and Guglietta, G. W. “Terahertz Spectroscopy”. In: *Analytical Chemistry* 83 (2011).

[30] Siegel, P. H. “Terahertz Pioneer: David H. Auston”. In: *IEEE TRANSACTIONS ON TERAHERTZ SCIENCE AND TECHNOLOGY* 1.1 (Sept. 2011).

[31] Federici, J. F. et al. “Chapter 11 - Detection of Explosives by Terahertz Imaging”. In: *Counterterrorist Detection Techniques of Explosives.* Ed. by Yinon, J. Amsterdam: Elsevier Science B.V., 2007, pp. 323–366. ISBN: 978-0-444-52204-7. DOI: `https://doi.org/10.1016/B978-044452204-7/50030-4`. URL: `https://www.sciencedirect.com/science/article/pii/B9780444522047500304`.

[32] Jepsen, P. U. “Phase Retrieval in Terahertz Time-Domain Measurements: a “how to” Tutorial”. In: *Journal of Infrared, Millimeter, and Terahertz Waves* 40.4 (2019), pp. 395–411. DOI: `10.1007/s10762-019-00578-0`.

[33] Kutz, J. et al. “A Terahertz Fast-Sweep Optoelectronic Frequency-Domain Spectrometer: Calibration, Performance Tests, and Comparison with TDS and FDS”. In: *Applied Sciences* 12.16 (2022). ISSN: 2076-3417. DOI: `10.3390/app12168257`. URL: `https://www.mdpi.com/2076-3417/12/16/8257`.

[34] Coutaz, J.-L., Garet, F. and Wallace, V. *Principles of Terahertz Time-Domain Spectroscopy.* Pan Standford Publishing, 2018.

[35] Deninger, A. J. et al. “2.75 THz tuning with a triple-DFB laser system at 1550 nm and InGaAs photomixers”. In: *Journal of Infrared, Millimeter, and Terahertz Waves* 36 (2015), pp. 269–277.

[36] Vogt, D. W., Erkintalo, M. and Leonhardt, R. “Coherent continuous wave terahertz spectroscopy using Hilbert transform”. In: *Journal of Infrared, Millimeter, and Terahertz Waves* 40 (2019), pp. 524–534.

[37] Hoshina, H. et al. “Noninvasive Mail Inspection System with Terahertz Radiation”. In: *Appl. Spectrosc.* 63.1 (Jan. 2009), pp. 81–86. URL: `https://opg.optica.org/as/abstract.cfm?URI=as-63-1-81`.

[38] Shen, X. et al. “Detection and Segmentation of Concealed Objects in Terahertz Images”. In: *IEEE Transactions on Image Processing* 17.12 (2008), pp. 2465–2475. DOI: `10.1109/TIP.2008.2006662`.

[39] Wang, Y. et al. “Terahertz absorbance spectrum fitting method for quantitative detection of concealed contraband”. In: *Journal of Applied Physics* 102.11 (2007), p. 113108.

[40] Naftaly, M., Vieweg, N. and Deninger, A. “Industrial applications of terahertz sensing: State of play”. In: *Sensors* 19.19 (2019), p. 4203.

[41] Vieweg, N., Deninger, A. and Leisching, P. "True OEM terahertz systems for industrial applications". In: *Photonic Instrumentation Engineering VI*. Ed. by Soskind, Y. G. Vol. 10925. International Society for Optics and Photonics. SPIE, 2019, 109250U. DOI: 10.1117/12.2511268. URL: https://doi.org/10.1117/12.2511268.

[42] Vieweg, N. et al. "Terahertz-time domain spectrometer with 90 dB peak dynamic range". In: *Journal of Infrared, Millimeter, and Terahertz Waves* 35 (2014), pp. 823–832.

[43] Afsah-Hejri, L. et al. "A Comprehensive Review on Food Applications of Terahertz Spectroscopy and Imaging". In: *Comprehensive Reviews in Food Science and Food Safety* 18.5 (2019), pp. 1563–1621. DOI: https://doi.org/10.1111/1541-4337.12490. eprint: https://ift.onlinelibrary.wiley.com/doi/pdf/10.1111/1541-4337.12490. URL: https://ift.onlinelibrary.wiley.com/doi/abs/10.1111/1541-4337.12490.

[44] Wang, K., Sun, D.-W. and Pu, H. "Emerging non-destructive terahertz spectroscopic imaging technique: Principle and applications in the agri-food industry". In: *Trends in Food Science & Technology* 67 (2017), pp. 93–105. ISSN: 0924-2244. DOI: https://doi.org/10.1016/j.tifs.2017.06.001. URL: https://www.sciencedirect.com/science/article/pii/S0924224417301565.

[45] Navale, A. and Dipanwita, H. "Evaluation of machine learning algorithms to Sentinel SAR data". In: *Spatial Information Research* 28.3 (2020), pp. 345–355. DOI: 10.1007/s41324-019-00296-8.

[46] Auer, S., Bamler, R. and Reinartz, P. "RaySAR - 3D SAR simulator: Now open source". In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2016, pp. 6730–6733. DOI: 10.1109/IGARSS.2016.7730757.

[47] Nishino, R. and Loomis, S. H. C. "Cupy: A numpy-compatible library for nvidia gpu calculations". In: *31st confernce on neural information processing systems* 151.7 (2017).

[48] Xi, L., Guosui, L. and Ni, J. "Autofocusing of ISAR images based on entropy minimization". In: *IEEE Transactions on Aerospace and Electronic Systems* 35.4 (1999), pp. 1240–1252.

[49] Ghighlia, D. and Pritt, M. *Two-dimensional phase unwrapping: Theory, algorithms, and software*. JohnWiley & Sons, 1998.

[50] Zhang, T. et al. "Rapid and robust two-dimensional phase unwrapping via deep learning". In: *Opt. Express* 27.16 (Aug. 2019), pp. 23173–23185. DOI: 10.1364/OE.27.023173. URL: https://opg.optica.org/oe/abstract.cfm?URI=oe-27-16-23173.

[51] Herráez, M. A. et al. "Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path". In: *Applied optics* 41.35 (2002), pp. 7437–7444.

[52] Liang, R. et al. "A Novel Coherence Estimation Method for InSAR". In: *2023 8th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR)*. 2023, pp. 1–5. DOI: 10.1109/APSAR58496.2023.10388925.

[53] Hajnsek, I. and Desnos, Y.-L. *Polarimetric synthetic aperture radar: principles and application.* Vol. 25. Springer Nature, 2021.

[54] Shirley, P., Ashikhmin, M. and Marschner, S. *Fundamentals of computer graphics.* AK Peters/CRC Press, 2009.

[55] Schüßler, C. et al. "A realistic radar ray tracing simulator for large MIMO-arrays in automotive environments". In: *IEEE Journal of Microwaves* 1.4 (2021), pp. 962–974.

[56] Meister, D. et al. "A survey on bounding volume hierarchies for ray tracing". In: *Computer Graphics Forum.* Vol. 40. 2. Wiley Online Library. 2021, pp. 683–712.

[57] Shirley, P. *Ray tracing: The next week.* 2020.

[58] Williams, A. et al. "An efficient and robust ray-box intersection algorithm". In: *ACM SIGGRAPH 2005 Courses.* 2005, 9–es.

[59] Malmgren-Hansen, D. et al. "Improving SAR automatic target recognition models with transfer learning from simulated data". In: *IEEE Geoscience and remote sensing Letters* 14.9 (2017), pp. 1484–1488.

[60] T. Ulaby, F. and Dobson, M. C. *Handbook of Radar Scattering Statistics for Terrain.* Artech House Publishers, 1989.

[61] Yun, G., Crabtree, K. and Chipman, R. A. "Three-dimensional polarization ray-tracing calculus I: definition and diattenuation". In: *Applied optics* 50.18 (2011), pp. 2855–2865.

[62] Vasiou, E. et al. "A detailed study of ray tracing performance: render time and energy cost". In: *The Visual Computer* 34 (2018), pp. 875–885.

[63] Nordmark, R. and Olsén, T. *A Ray Tracing Implementation Performance Comparison between the CPU and the GPU.* 2022.

[64] Nickolls, J. et al. "Scalable parallel programming with cuda: Is cuda the parallel programming model that application developers have been waiting for?" In: *Queue* 6.2 (2008), pp. 40–53.

[65] NVIDIA. "Nvidia Optix 8.0 Programming Guide, Version 1.16". In: (2023).

[66] Vang, M. C. H., Thomsen, G. M. and Dyhre, L. "Synthetic Aperture Radar: Modeling and Reduced Scale THz Setup". Semester project at the department of Materials and Production at Aalborg University. 2024.

[67] Alkemade, C. et al. "A review and tutorial discussion of noise and signal-to-noise ratios in analytical spectrometry—I. Fundamental principles of signal-to-noise ratios". In: *Spectrochimica Acta Part B: Atomic Spectroscopy* 33.8 (1978), pp. 383–399. ISSN: 0584-8547. DOI: `https://doi.org/10.1016/0584-8547(78)80049-4`. URL: `https://www.sciencedirect.com/science/article/pii/0584854778800494`.

[68] *Spraymaling, kobber.* ver. 2. BILTEMA. Aug. 2023.

[69] Blender Online Community. *Blender - a 3D modelling and rendering package.* Blender Foundation. Blender Institute, Amsterdam. URL: http://www.blender.org.

[70] *SAR target classification using Deep Learning. MATLAB & Simulink.* https://se.mathworks.com/help/radar/ug/sar-target-classification-using-deep-learning.html. Accessed: 2025-05-16.

[71] Wang, Q., Xie, L. and and, Y. Y. "Overview of imaging methods based on terahertz time-domain spectroscopy". In: *Applied Spectroscopy Reviews* 57.3 (2022), pp. 249–264. DOI: 10.1080/05704928.2021.1875480. eprint: https://doi.org/10.1080/05704928.2021.1875480. URL: https://doi.org/10.1080/05704928.2021.1875480.

[72] Watanabe, S. and Shimano, R. "Compact terahertz time domain spectroscopy system with diffraction-limited spatial resolution". In: *Review of Scientific Instruments* 78.10 (Oct. 2007), p. 103906. ISSN: 0034-6748. DOI: 10.1063/1.2796941. eprint: https://pubs.aip.org/aip/rsi/article-pdf/doi/10.1063/1.2796941/15974563/103906_1_online.pdf. URL: https://doi.org/10.1063/1.2796941.

[73] Redo-Sanchez, A. et al. "Damage and defect inspection with terahertz waves". In: *The 4th International Workshop on Ultrasonic and Advanced Methods for Nondestructive Testing and Material Characterization.* Vol. 11. 7. 2006, pp. 67–77.

[74] Gui, S. et al. "Analysis of Security Imaging Method for Walking Human Screening With Single Channel Synthetic Aperture Radar". In: *IEEE Access* 7 (2019), pp. 111363–111374. DOI: 10.1109/ACCESS.2019.2931413.

[75] Nezadal, M., Schür, J. and Schmidt, L.-P. "Non-destructive testing of glass fibre reinforced plastics with a synthetic aperture radar in the lower THz region". In: *2012 37th International Conference on Infrared, Millimeter, and Terahertz Waves.* 2012, pp. 1–2. DOI: 10.1109/IRMMW-THz.2012.6380301.

[76] Peter Shirley Trevor David Black, S. H. *Ray Tracing in One Weekend.* Apr. 2025. URL: https://raytracing.github.io/books/RayTracingInOneWeekend.html.

[77] SDMS. *MSTAR data set.* 2025. URL: https://www.sdms.afrl.af.mil/index.php?collection=mstar (visited on 05/09/2025).

[78] Kulpa, K. S. et al. "An advanced SAR simulator of three-dimensional structures combining geometrical optics and full-wave electromagnetic methods". In: *IEEE Transactions on Geoscience and Remote Sensing* 52.1 (2013), pp. 776–784.

[79] Xu, G. et al. "Acceleration of shooting and bouncing ray method based on OptiX and normal vectors correction". In: *Plos one* 16.6 (2021), e0253743.

[80] Cha, M. et al. "Improving SAR automatic target recognition using simulated images under deep residual refinements". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE. 2018, pp. 2606–2610.

[81] Inkawhich, N. et al. “Bridging a gap in SAR-ATR: Training on fully synthetic and testing on measured data”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021), pp. 2942–2955.

[82] Danylov, A. A. et al. “Terahertz inverse synthetic aperture radar (ISAR) imaging with a quantum cascade laser transmitter”. In: *Optics express* 18.15 (2010), pp. 16264–16272.

[83] Iwaszczuk, K., Heiselberg, H. and Jepsen, P. U. “Terahertz radar cross section measurements”. In: *Optics Express* 18.25 (2010), pp. 26399–26408.

[84] Lewis, R. A. “A review of terahertz sources”. In: *Journal of Physics D: Applied Physics* 47.37 (2014), p. 374001.

[85] Hagemann, H.-J., Gudat, W. and Kunz, C. “Optical constants from the far infrared to the x-ray region: Mg, Al, Cu, Ag, Au, Bi, C, and Al2O3”. In: *Journal of the Optical Society of America* 65.6 (1975), pp. 742–744.

[86] Hossain, M. M. S., Nahar, N. K. and Sertel, K. “Resolution-Enhanced Polarimetric Terahertz Imaging”. In: *IEEE Transactions on Terahertz Science and Technology* (2024).

[87] Watanabe, S. “Terahertz polarization imaging and its applications”. In: *Photonics*. Vol. 5. 4. MDPI. 2018, p. 58.

[88] Yang, Z. et al. “Photonic THz InISAR for 3D Positioning With High Resolution”. In: *Journal of Lightwave Technology* 41.10 (2023), pp. 2999–3006. DOI: 10.1109/JLT.2023.3240507.

Appendices

Appendix A

A.1 BVH DATA Management

After the BVH is built, all the necessary information of the BVH data structure is stored in three matrices, box_interval_mat $\bar{\bar{\mathbf{B}}}$, index_mat $\bar{\bar{\mathbf{X}}}$ and tri_leaf_mat $\bar{\bar{\mathbf{T}}}$. $\bar{\bar{\mathbf{B}}}$ is a $6 \times n$ matrix storing the x-, y- and z-intervals of all aabbs, by their maximum and minimum values, where n is the total number of bounding volumes in the BVH. If k is the depth of the tree, then n is given by:

$$n = 2^{k+1} - 1. \tag{A.1}$$

$\bar{\bar{\mathbf{T}}}$ is a $4 \times 3 \times q \times m$ matrix, storing the vertices and centroids of all triangles in the BVH leafs. q is the maximum allowed number of triangles in a leaf, and m is the total number of leafs, where m is given by $m = 2^k$. $\bar{\bar{\mathbf{X}}}$ is a $2 \times n$ matrix storing the child indices of the parent nodes. If a node has no children, i.e. is a leaf node, $\bar{\bar{\mathbf{X}}}$ stores the index location of the triangles in $\bar{\bar{\mathbf{T}}}$. The $\bar{\bar{\mathbf{X}}}$ matrix could look like this for an arbitrary BVH

$$\bar{\bar{\mathbf{X}}} = \begin{bmatrix} 1 & 3 & 5 & \dots & -9 & -9 & \dots \\ 2 & 4 & 6 & \dots & 126 & 127 & \dots \end{bmatrix}, \tag{A.2}$$

where the $\bar{\bar{\mathbf{X}}}[:, n]$ column entries points to the children of node n. If n has no children, the first entry in column n is flagged with -9, and the second entry is the m-index in $\bar{\bar{\mathbf{T}}}$.

A.2 Terahertz Frequency Domain Spectroscopy

Continuous-wave (CW) THz frequency domain spectroscopy is a method of spectroscopy, where the measuring of spectra happens in the frequency domain. Here, the frequency is scanned over a given interval emitting THz radiation, and measuring the amplitude of the incoming THz radiation for each frequency. The THz radiation is generated from two distributed feedback lasers, operating at frequencies slightly different from each other. The beams are then guided into a 2x2 coupler creating a beat note. This beat note is guided into the emitter and detector creating sum frequency radiation and sum difference radiation. These operate using photomixers, which are semiconductor devices designed to to only operate in the THz regime. The THz emitter is set to a bias voltage, creating the THz radiation when the incoming light from the lasers are applied. Likewise, the beat note is guided into the detector, which, when it detects THz radiation, generates a photocurrent. This is amplified by the lock-in detection scheme. This process is illustrated in figure A.1. [35]

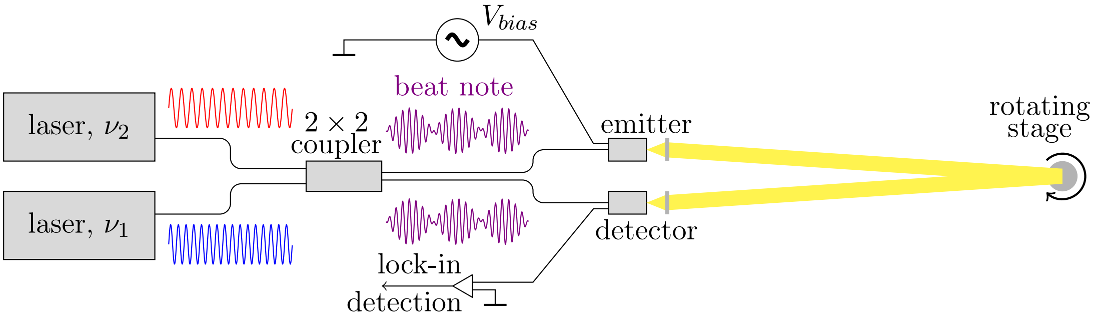
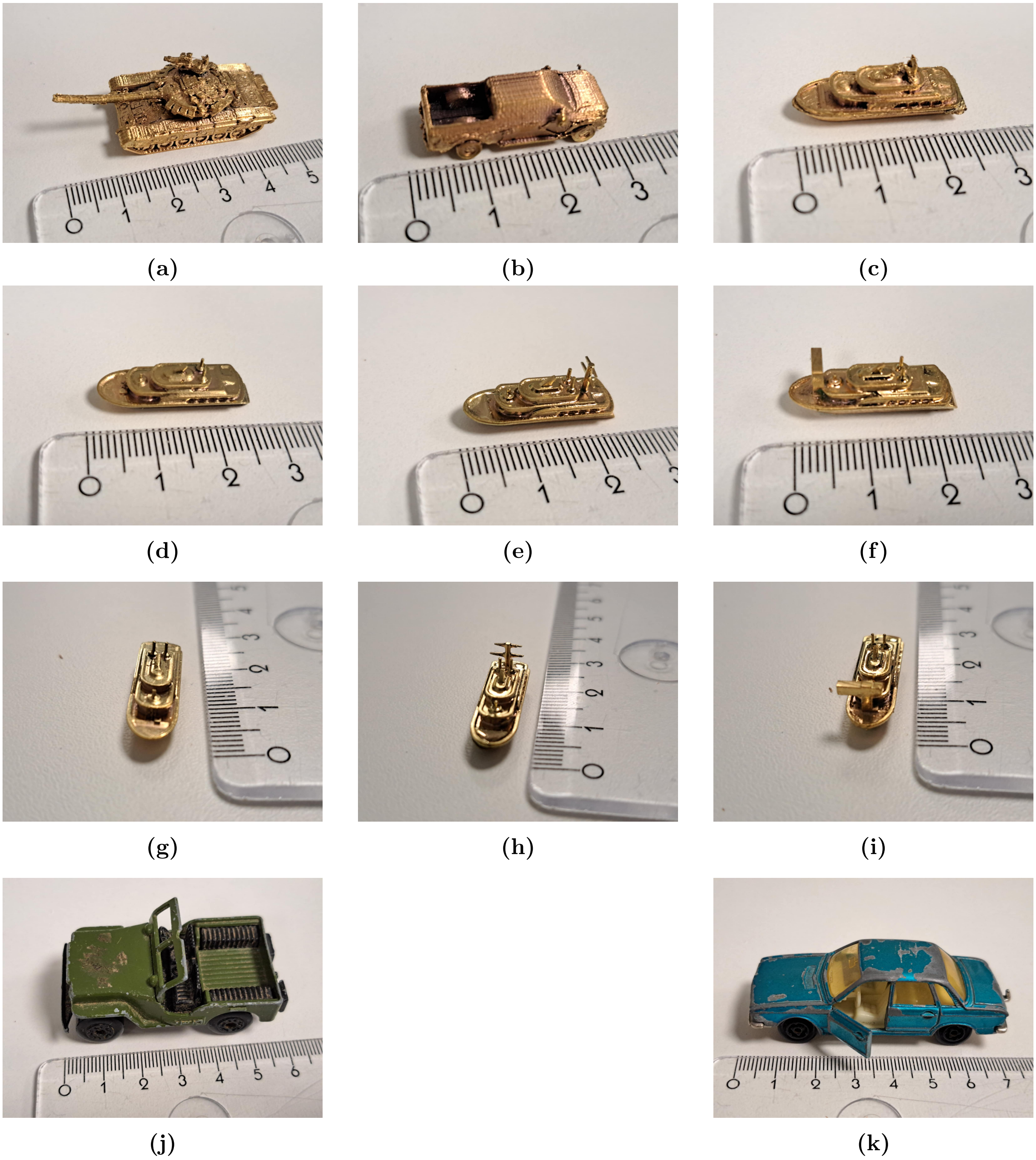


Figure A.1. *Schematic of eksperimental setup with the TOPTICA THz-FDS system.*

The principle behind photomixing governing THz generation is for the scope of the project not included, but can be found in the collective works: Coutaz, Garet and Wallace [34], Deninger et al. [35], and Vogt, Erkintalo and Leonhardt [36]

A.3 THz SAR models

Here we present images of all the models used in chapter 6, see figure A.2 below.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure A.2. ***(a)*** *T72 tank.* ***(b)*** *Ford pickup truck.* ***(c)*** *First Boat print.* ***(d,g)*** *Second boat print.* ***(e,h)*** *Boat with antenna.* ***(f,i)*** *Boat with front crane.* ***(j)*** *Toy Jeep.* ***(k)*** *Toy car.*

Appendix B

B.1 Hyperspectral SAR Imaging Theory

In SAR usually, the amplitude of the time domain signal of a scene is mapped and focused on an image. If the frequency response can also be mapped to an image in a similar manner, then a hyperspectral image can be generated. This is relevant in the THz regime since different compounds can be identified by their absorption spectra. Assuming a certain bandwidth is measured in discrete steps along frequency with a coherent detection scheme, then the received signal at a single frequency will be

$$S(f) = \sum_{n=1}^{N} A_n\left(f\right) e^{-i2\pi f t_n + \phi} + \psi. \tag{B.1}$$

Here N different reflectors in a scene with different time delays, t_n, are assumed to contribute to the signal with an amplitude and phase term. Furthermore, phase noise ϕ and additive noise ψ will also be present in a measured signal, but will be neglected in the following derivation. In a BPA manner we will divide a plane into cells, which are indexed with (i, j). An expected phase to the center of such a cell can be given by the following reference function

$$h_{i,j}(f) = e^{-i2\pi f \tilde{t}_{i,j}}, \tag{B.2}$$

where $\tilde{t}_{i,j}$ is the time required for the light to travel from the emitter to the cell center and back to the detector. If L different measurements are taken at different angles, a corresponding reference function can be applied to each, yielding

$$\begin{aligned} H_{i,j}(f) &= \sum_{l}^{L} S_l\left(f\right) h^*_{i,j}\left(f\right) \\ &= \sum_{l}^{L} A^l_{i,j}\left(f\right) e^{-i2\pi f\left(t_{i,j} - \tilde{t}_{i,j}\right))} + \sum_{l}^{L} \sum_{\substack{q,k, \\ (q,k)\neq(i,j)}}^{M} A^l_{q,k}\left(f\right) e^{-i2\pi f\left(t_{q,k} - \tilde{t}_{i,j}\right))}. \end{aligned} \tag{B.3}$$

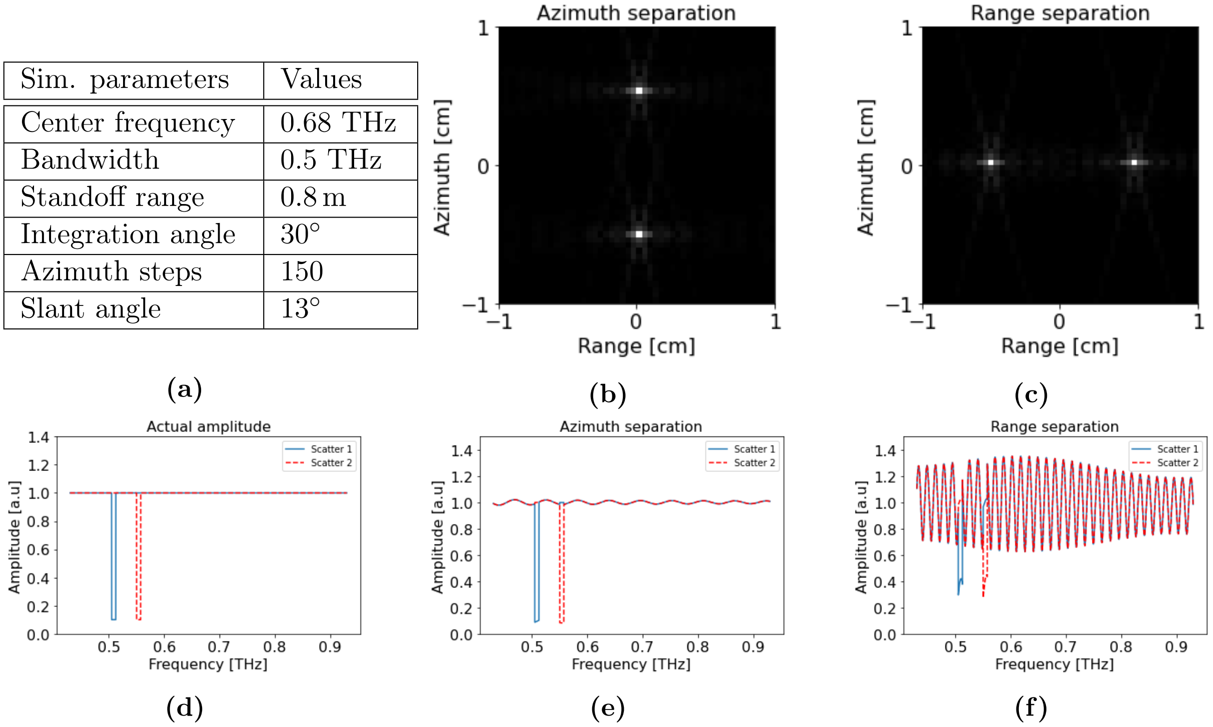
Here the $A_{i,j}$ amplitudes are the ones of interest and origination from the (i, j) cell and there are assumed to be $M \times M$ cells. Ideally, the resulting phase connected to these amplitudes should be close to zero, since $t_{i,j}$ and $\tilde{t}_{i,j}$ should be similar. Hence the values from the (i, j) cell will add up constructively. The contributions originating from all the other cells should naively have arbitrary phases and end up canceling each other out. However, this will most likely not be the case and the signal have to be filtered in the time domain to mitigate these terms, which will be discussed in the next section.

B.2 Hyperspectral SAR Imaging Simulations

To first investigate the possibility of mapping spectral information to a SAR image, simple simulations of point scatters were used. Here the received signal from N scatters is given by

$$S(f) = \sum_{n=1}^{N} A_n(f)\, e^{-i2\pi f t_n}. \tag{B.4}$$

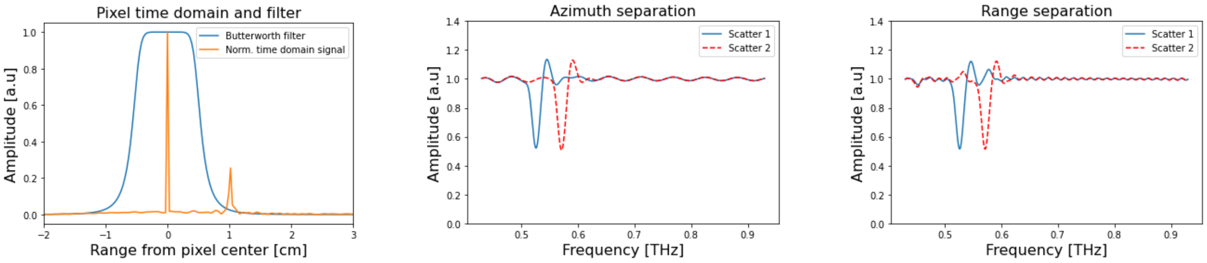
These signals are then calculated from different azimuth angles and with parameters similar to our experimental setup, which can be seen in table B.1a. SAR images of two points can be seen in figure B.1b and B.1c. The resulting spectra from the pixels on top of the points using the method descibed in section B.1 can be seen in B.1e and B.1f. The actual amplitudes which we want to get as close to as possible are presented in figure B.1d.



Sim. parameters	Values
Center frequency	0.68 THz
Bandwidth	0.5 THz
Standoff range	0.8 m
Integration angle	30°
Azimuth steps	150
Slant angle	13°

Figure B.1. *(**a**) Simulation parameters. Pointscatters separated in (**b**) azimuth direction and (**c**) range direction, where (**e**) and (**f**) are the obtained spectra from the points using the hyperspectral method. (**d**) The actual amplitudes used in the simulation.*

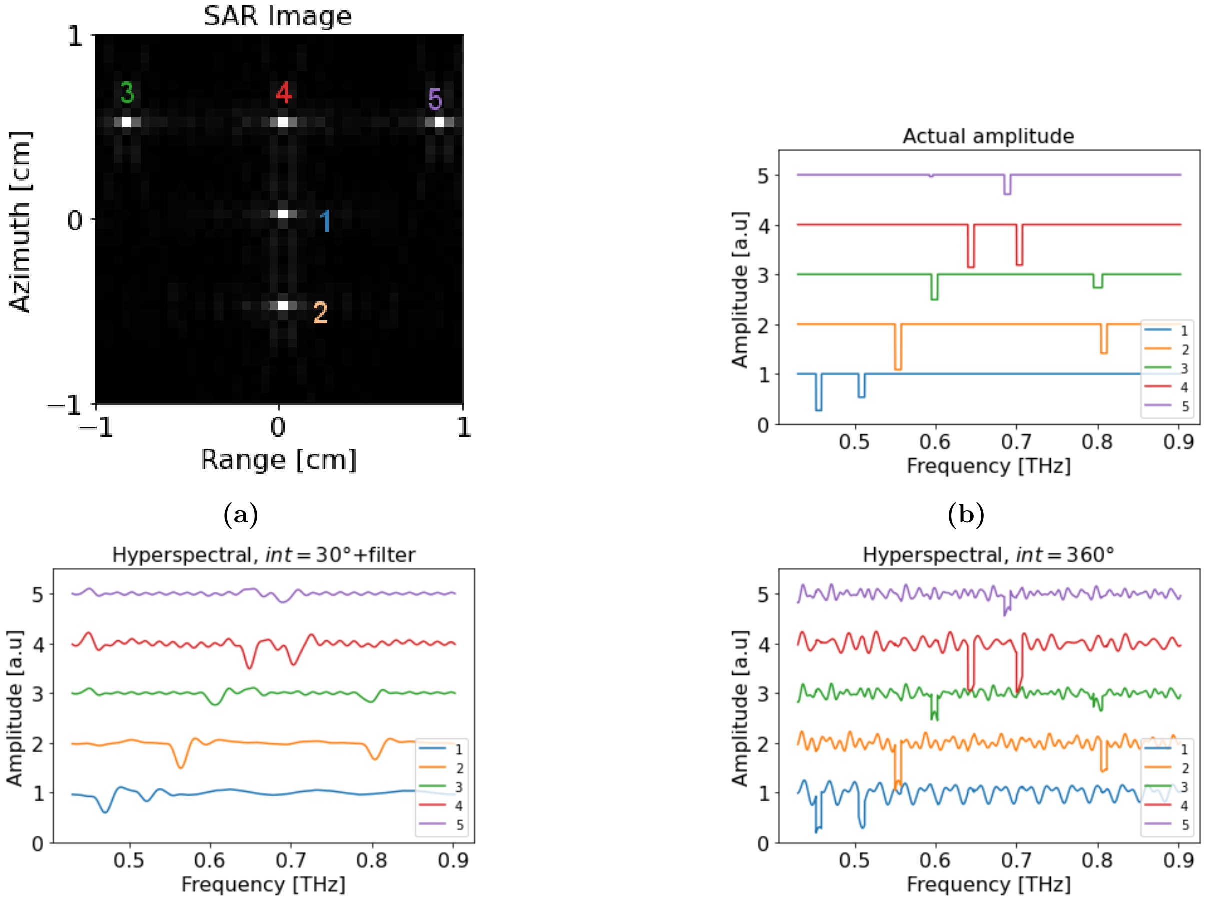
When the scatterers are separated in the azimuth direction the resulting spectra of the pixels of the two scatterers are close to the actual amplitudes. However, when the scatterers are separated in range direction, there will be significant oscillations in the spectra which comes from terms of the other scatterer. A method to remove these oscillations is to Fourier transform the complex frequency spectra for the pixels and apply a filter, which can be seen in figure B.2.



(a) (b) (c)

Figure B.2. *(**a**) Normalized Fourier transform of the spectra from the left most pointscatter in the range separated example together with a butterworth bandpass filter. (**b**) and (**c**) Spectra from azimuth and range separated examples after the filter has been applied.*

The filter heavily affects the spectral resolution which makes the absorption peaks wider and less defined however, it removes drastic oscillations when the scatterers are only separated in the range direction. For demonstration, more scatterers are put into the scene with different absorptions which can be seen in figure B.3.



(a) (b)

(c) (d)

Figure B.3. *(**a**) SAR image of 5 points with different amplitudes which can be seen in (**b**). (**c**) Amplitudes obtained by the hyperspectral method with and integration angle of* 30° *and an applied filter. (**d**) Amplitudes from the hyperspectral method with an integration angle of* 360° *and without filter.*

The obtained hyperspectral data with an integration angle of 30° and with the use of a filter, seen in figure B.3c, has less spectral resolution and more broadened features compared to the actual amplitude in figure B.3b, which is similar to what is seen previously in figure B.2. Increasing the integration angle to 360° and avoiding filtering, a good spectral resolution is maintained but there will be oscillations across the spectra, which is presented in figure B.3d. Some filtering could possibly be applied to the 360° integration to obtain a better result.

From these simulations it appears to be possible to map spectral information to SAR images and the next step would be to implement these methods in practice using the THz SAR setup. This was not pursued due to time limitations.