Arbitrary motion Synthetic Aperture Radar

Filip Alberius, Erik Rolander

Master's thesis
2023:E13

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematical Statistics

Master’s Theses in Mathematical Sciences 2023:E13
ISSN 1404-6342

LUTFMS-3470-2023

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lu.se/

Arbitrary motion Synthetic Aperture Radar

Erik Rolander and Filip Alberius

Master's Thesis

Faculty of Engineering LTH

Department of Mathematical Statistics

Supervisors: Andreas Jakobsson and Rikard Nelander

In collaboration with Acconeer AB

Lund, May 2023

Abstract

The purpose of this thesis is to develop a novel approach to producing Synthetic Aperture Radar (SAR) images, assuming the scenario of arbitrary motion in regards to the radar sensor. SAR is a well-researched method for creating 2- or 3-dimensional radar images, traditionally assuming the radar sensor's motion to be linear and highly predictable. Thus the locations where each radar data is sampled are known and follow a predefined pattern. This thesis, however, explores new methods that enable the radar to move freely.

We have developed a prototype based on the A121 Pulsed Coherent Radar sensor from Acconeer to demonstrate the performance and use-case potential of the method we came up with. The resulting system can successfully create an accurate map of its surroundings, detecting obstacles in 3D space. Using this prototype setup, we have evaluated different configurations and kinds of optimizations for the algorithm developed, as well as configurations for the radar sensor, including lens choice. We have tested the setup for many different potential targets and evaluated its performance in several different environments.

The system relies on a stream of accurate pose data in regard to the radar sensor. Two different systems were explored for this purpose. Firstly, the use of an Inertial Measurement Unit (IMU), and secondly, using the IR lighthouse tracking from virtual reality hardware. Early experiments showed less than promising results for IMU-based tracking, and this path was not deemed worth exploring further. Tracking the sensor using a VR setup, however, showed promising results and more experiments were performed to evaluate the viability and limitations of its use in creating radar images. These two approaches, though, are a subset of a vast number of potential pose-tracking systems.

The final results show that SAR is possible to implement in cases where the motion of the radar is arbitrary, given the position of the sensor can be determined to some degree of accuracy for all collected radar data points. It also demonstrates how arbitrary motion SAR could potentially enable more use cases for radar sensors, for example, in VR, robotics for the mapping of environments, and object detection.

Sammanfattning

Syftet med denna avhandling är att utveckla en ny metod för att producera bilder med syntetisk aperturradar (SAR), med utgångspunkt i scenarier med arbiträr rörelse vad gäller radarsensorn. SAR är en väletablerad metod för att skapa 2- eller 3-dimensionella radarbilder, som traditionellt sett antar att radar-sensorns rörelse är linjär och förutsägbar. Därmed är positionerna för varje radardata-punkt kända och följer ett fördefinierat mönster. Denna avhandling utforskar emellertid nya metoder som gör det möjligt för radarn att röra sig fritt.

Vi har utvecklat en prototyp baserad på Pulsed Coherent Radar-sensorn A121 från Acconeer för att demonstrera potentialen och prestandan i vår metod. Det resulterande systemet kan framgångsrikt skapa en bild av sin omgivning och upptäcka hinder i 3D-rymden. Med hjälp av denna prototypuppställning har vi utvärderat olika konfigurationer och optimeringar av algoritmen, samt konfigurationer för radarsensorn, inklusive val av lins. Vi har testat uppställningen för många olika potentiella objekt och utvärderat dess prestanda i flera olika miljöer.

Systemet förlitar sig på en ström av exakta poseringsdata för radar-sensorn. Två olika system utforskades för detta ändamål. Först en Inertial Measurement Unit (IMU), och sedan VR-hårdvara som bygger på IR-fyr-spårning. Tidiga experiment visade mindre lovande resultat för IMU-baserad spårning, och denna väg ansågs inte vara värd att utforska vidare. Att spåra sensorn med hjälp av en VR-uppsättning visade emellertid lovande resultat, och fler experiment genomfördes för att utvärdera användbarheten och begränsningarna i dess användning för att skapa radarbilder. Dessa två tillvägagångssätt är dock en delmängd av ett stort antal potentiella spårningssystem.

Våra resultat visar att SAR är möjligt att implementera i fall där radarns rörelse är arbiträr, förutsatt att sensorns position kan bestämmas med viss noggrannhet för alla insamlade radardatapunkter. Det visar också hur SAR baserad på arbiträr rörelse potentiellt kan möjliggöra fler användningsområden för radar-sensorer, till exempel inom VR och robotik för kartläggning av miljöer och detektion av objekt.

Acknowledgements

This master's thesis project was conducted at Acconeer AB in Malmö during the winter of 2022/2023 as a part of the Computer Science program at LTH. We want to thank the systems team at Acconeer for their support as well as the staff at Acconeer in general. A special thank you to Rikard Nelander for helping us along as a mentor throughout the project. We want to thank Andreas Jakobsson at the Center for Mathematical Sciences at LTH as well for being our supervisor. Finally, we want to thank Günter Alce, and Joakim Eriksson from the Department of Design Sciences at LTH for letting us use the department's equipment and premises.

Filip Alberius & Erik Rolander

Popular Science Summary

Radar is a technology that uses radio waves to detect and locate objects in its vicinity. In its most simple form, it works by emitting a radio wave signal and measuring the time it takes for the signal to bounce back after it has hit an object, much like how sonar works with sound. This information is then used to determine the distance, speed, and other characteristics of the object. More sophisticated radar technologies, though, have a wide range of use cases, from military applications to weather forecasting and air traffic control. In military applications, radar is used for surveillance, target acquisition, and weapon guidance. Weather radar is used to track and forecast storms, while air traffic control radar tracks the location and speed of aircraft in flight.

Acconeer AB, a Malmö-based company started in 2012 around research into Pulsed Coherent Radar (PCR) technology at LTH, develops a now commercially available 60GHz radar sensor with a small footprint, approachable price, and small power consumption. Its A111 and A121 radar sensors are seeing increasing adoption by a plethora of tech giants and independent inventors alike for a multitude of use cases. Recently, Volvo announced Acconeer radar-based interior presence detection will be featured in some upcoming car models. Consumer electronic use cases like this are a very novel concept, just springing up in the last decade, and there is yet a lot to explore in this field.

In recent years, there have been several modern developments in radar technology. One of these is the use of Synthetic Aperture Radar (SAR) which, for example, allows for the creation of high-resolution images of the Earth's surface by using a radar system on a moving platform, such as a satellite or airplane. SAR is a type of radar that uses the motion of the radar antenna to create images containing a lot of information that would not be possible to achieve with a single radar sweep. The antenna motion allows the radar to gather data from different positions and angles, which are combined to create a synthetic aperture. The concept has also been shrunken to the Acconeer PCR scale to produce images of objects at a more human scale. There is ongoing research into using radar for applications such as autonomous driving, where radar sensors can be used to detect and avoid obstacles on the road.

In the VR-space there is a constant push to enhance the user experience through immersion, which has pushed technology far in terms of different performance metrics like latency, resolution, and precision. There is, however, still the risk of breaking immersion by, for example, bumping into an obstacle in the environment. This has prompted research into technology to blur the lines between VR and AR by making VR users aware of their real-world surroundings. This has been tackled in different ways, for example, by using RGB-D cameras and LiDAR.

In this study, we have explored novel ways of using radar for SAR in cases where the motion of the radar sensor is unpredictable. We have implemented this successfully in the specific use case of obstacle detection in the VR application.

Notations and Symbols

Radar - Radio Detection and Ranging
SAR - Synthetic Aperture Radar
PCR - Pulsed Coherent Radar
IMU - Inertial Measurement Unit
VR - Virtual Reality
AR - Augmented Reality
HMD - Head Mounted Display
LiDAR - Light detection and ranging
SLAR - Side Looking Airborne Radar
VAA - Virtual Antenna Array
MIMO - Multiple Input Multiple Output
FFT - Fast Fourier Transform
PSD - Power Spectral Density
SLAM - Simultaneous Localization And Mapping
SNR - Signal To Noise Ratio

Contents

1 Introduction

Radar, an acronym for Radio Detection and Ranging, is an established technology that uses radio waves to detect and locate objects. Radar, in very basic terms, works by emitting a radio frequency signal from a transmitter, which then travels through the air and reflects off of objects in its path. These reflections, or echoes, are picked up by a receiver and processed to determine the location, speed, and other characteristics of the objects. Research into radar technology largely got up to speed during World War II due to its potential for military use cases. Still, lately, radar has seen increasing use in the consumer market, for instance, being used in the automotive, industrial, and consumer electronic sectors. [1]

Synthetic Aperture Radar (SAR) is the concept of utilizing multiple radars, or the controlled motion of one radar, to produce more information than one can gather from one singular, equivalent radar. Using this technique, one can create 2D images or 3D reconstructions of target objects. Traditionally, this is mostly used in the field of satellite and aircraft-mounted radar, for example, to image the topology of landscapes. [2]

Our idea for this project started around VR, namely the possibility of using radar for obstacle detection in a VR context. Utilizing a novel approach to SAR which uses the arbitrary motion of a Head Mounted Display (HMD), we wanted to detect objects in proximity to a VR user and warn them of obstacles in real time. There are solutions for this already in VR headsets today that use cameras to map the surroundings, and rumors of LiDAR (Light Detection and Ranging) sensors being utilized in next-gen VR headsets. Compared to these approaches, radar has advantages both in price and complexity.

To realize this concept, we worked with Acconeer and their radar technology. Acconeer, with roots in research at LTH, is at the forefront of developing the novel Pulsed Coherent Radar (PCR) technology aimed at the high-volume consumer market. The PCR technology has the benefit of combining the low power consumption of pulsed radar with the high accuracy of coherent radar. Acconeer's A121 PCR sensor, with a small footprint of a 5x5mm chip, operates at 60GHz. With a wavelength of 4.9mm, and with the use of phase coherency information the sensor achieves sub-millimeter precision and can sense objects up to 20m away. Acconeer and the A121 sensor are the basis for our research into SAR-based obstacle detection.

One could imagine a naive solution to the obstacle detection problem, making simple use of a radar sensor to warn when objects get within close proximity. We however imagined a more sophisticated solution where concepts lifted from SAR technology could be used to create a 3D mapping of the surroundings. This would have many benefits, like keeping previously detected obstacles in memory to be able to warn the user of them despite them not being in view of the radar. Also, virtual representations of objects could be created, with their shape being reconstructed accurately.

1.1 Objectives

The goal of this master's thesis was to explore the prospects of using arbitrary antenna motion to implement SAR with the purpose of detecting objects in space. Traditionally, highly predictable locations along linear vectors are used to create virtual antenna arrays when performing SAR imaging, but we wanted to break SAR technology free of this constraint. Our goals for this project were three-fold:

1. We set out to explore a novel approach to allow SAR to be used in platforms that move unpredictably, assuming the radar's motion can be precisely tracked, which would unlock many possibilities for SAR-based applications.
2. More practically, we wanted to produce a working prototype using some means of pose-tracking to prove this concept.
3. We wanted to examine the possibilities of using SAR-based object detection as a viable feature for more general use cases, such as in robotics.

1.2 Thesis Outline

This thesis report is structured as follows:

- **Chapter 2: Technical background**

  A description of different essential concepts that make out the basis of the work in the thesis.

- **Chapter 3: Methodology**

  Explaining how the algorithms and prototype developed were built and how it works, as well as describing the different experiments used to evaluate them.

- **Chapter 4: Results**

  The results are presented here in the form of graphs and radar images, as well as observations from the experiments.

- **Chapter 5: Discussion**

  Discussions and analysis of the results, and evaluation of the prototype's performance.

- **Chapter 6: Conclusion**

  The conclusions and takeaways from the project. Potential for future work is also presented.

2 Technical Background

2.1 Radar

Radar is a remote sensing technology that utilizes radio waves to detect and locate objects in space. It was first developed in the early 20th century, primarily for military applications such as detecting enemy aircraft during World War II. The basic principle of radar is to send out short pulses of radio waves from a transmitter, which then travel through space, bounce off the objects, and are reflected to a receiver. The time delay between the transmission and reception of the signal, known as the time of flight, determines the distance to the object. The strength of the returned signal provides information about the object's size, shape, and material composition. These properties determine the radar cross-section of the object. By measuring the Doppler shift in the frequency of the returned signal, radar can also determine the object's velocity relative to the radar system. Modern radar systems utilize a wide range of signal processing techniques, including frequency modulation, pulse compression, and synthetic aperture radar (SAR), to improve their performance and accuracy. Radar has many civilian and military applications, including air traffic control, weather forecasting, oceanography, and surveillance [1]. This project is conducted within the field of radar signal processing. Specifically, we have worked with radar sensors of the PCR (Pulsed Coherent Radar) variety.

2.1.1 Pulsed Coherent Radar

Acconeer's radar technology expands the boundaries of what is possible for radar, with the key being the high-performance, energy-efficient pulsed coherent radar design.

The principle of pulsed coherent radar design is based on the pulse. The circuit can emit a very short burst, or pulse, of electromagnetic radiation. The frequency of the radiation is 60GHz, and the pulse only lasts for a few hundred picoseconds. The radar transmitter emits the pulse, which then travels through the air and reflects off any reflective object. This reflection is detected by the radar receiver. Meanwhile, a delay circuit in the radar sensor creates an identical pulse after a set time. This delay equals the time time-of-flight to a specific distance. The delayed signal is then multiplied by the signal from the receiver. The amplitude of the resulting signal will correspond to how well the distance traveled by the radar pulse matches the delayed pulse. The resulting signal is sampled by the circuit and represented as a complex value. [3]

By rapidly making multiple consecutive pulses while incrementally increasing the delay in the delay circuit, we sample the radar signal at different distances. We refer to the resulting array of complex numbers as a sweep. There are a lot of parameters for the sweep that may be configured. Among these are the start point, step length, and the number of pulses, allowing for control over the range and resolution of the sweep. The radar will make a set number of these sweeps and put them into an array called a frame. The sweeps in the frame will resemble the ones in figure 2.1. [4]

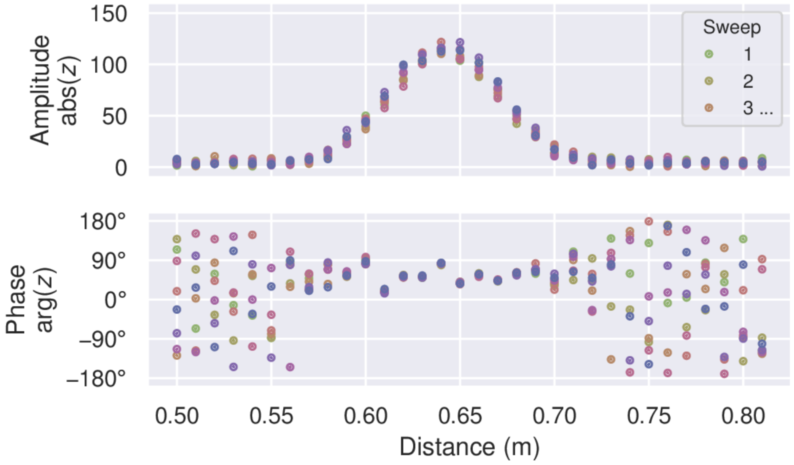


Figure 2.1: Amplitude and phase data from multiple sweeps in a frame.

The complex numbers received from each pulse contain phase information. This is where the "Coherent" in PCR comes from, and is another hugely useful detail of the PCR technology. The wavelength of the radiation emitted by the radar is $\lambda = f^{-1} \cdot c = (60 \cdot 10^9)^{-1} \cdot c \approx 4.96mm$. Within this one wavelength, the phase of the signal rotates 2π. With this information, one can get sub-millimeter precision from the sensor from the phase data in between sweeps. If for example, the signal's phase rotates by π between sweeps, we can conclude that the object has moved about 1.23mm in relation to the sensor. The distance is half the change in phase since the signal travels both to the object and back to the sensor. This assumes the object has not moved more than half a wavelength between frames. We get the same phase information if the object moves $n2\pi + \pi/2$, for any n. [5]

In figure 2.1 we see the data from a case of no relative movement between the radar and the object generating the reflection received. In figure 2.2 however, the phase varies between sweeps as the object generating the reflection moves in relation to the radar.

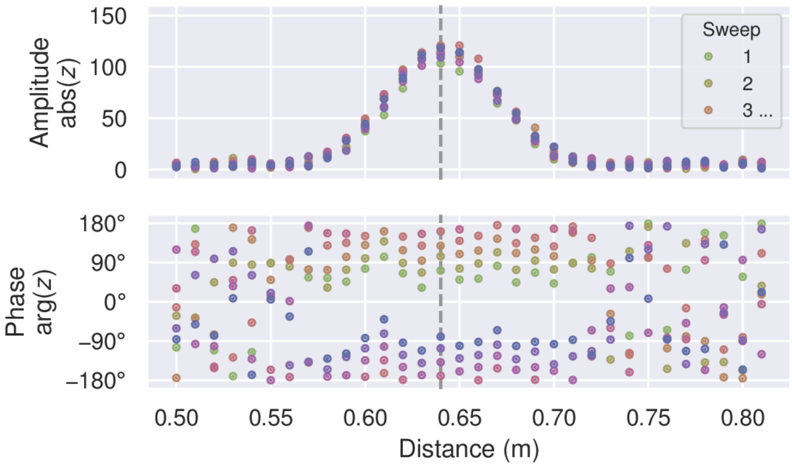


Figure 2.2: Amplitude and phase data from a moving object.

2.2 Synthetic Aperture Radar

Synthetic-Aperture Radar (SAR) is a common and well-researched radar technique for creating high-quality radar images, and the purpose of this project is to create a new variant of SAR, that yields the same high-quality images without the need for controlled movement.

The SAR technique was born out of Side Looking Airborne Radar (SLAR), where radar beams are transmitted from the side of airplanes to get an image of the reflectivity or topology of the underlying ground area. [6] [7]

The resolution of SLAR is the minimum distance needed between two objects for them to show up independently on the resulting image. It is constrained by the beam width and commonly measured in two directions, the azimuth resolution along the flight path and the slant range resolution perpendicular to the flight path. The slant range resolution is limited by the system bandwidth and the speed of light, while the azimuth resolution is given by the azimuth beam width and the slant range distance. The beam width is, in turn, given by the wavelength and length of the antenna. To get a finer azimuth resolution, one would need to increase the radar's aperture by increasing the antenna's length. This, however, quickly becomes unfeasible as the antenna must be transported on an aircraft. [6] [7]

The problem is solved by creating a synthetic antenna with a much larger aperture than what is possible otherwise. By flying along a linear path and continuously taking readings, the results from the readings can be combined as if they were a single reading from an antenna with the same length as the flight path. The resolution is then limited by the maximum length of the flight path instead of the antenna size. The flight path's length is only limited by the time during which the radar is within the range given by the beam width of the returning signal. It is, therefore, desirable to have a small physical aperture when doing SAR, as a large beam width means a larger time frame in which a point is visible to the radar and, therefore, a longer flight path. [6] [7]

2.3 Virtual Antenna Arrays

A Virtual Antenna Array (VAA) is a synthetic antenna created through different designs and means of post-processing, intending to increase the capacity or resolution of a received signal. In our case, the virtual antenna array is randomly created from arbitrary poses. SAR, briefly explained in section 2.2, is an example of a VAA in which 2- or 3-dimensional images can be created by moving the physical antenna. VAAs, however, are broader in definition and can be constructed in several ways. For example, VAAs can be created by several physical antennas in Multiple Input Multiple Output (MIMO) setups. If a time-invariant signal is used, time synchronization isn't needed, and a single physical antenna in motion could be used, passing through different locations. Each location would be part of the resulting array. [8]

2.4 Matched Filter Method

The matched filter method is a signal processing technique that is commonly used in communication systems to detect and extract a known signal from noisy data. The SAR algorithm is based on this method. The method involves correlating the received signal with a replica of the original signal that is generated using prior knowledge of the signal characteristics. The resulting output of the correlation is then thresholded to make a binary decision on the presence or absence of the signal. The matched filter method is particularly effective in scenarios where the received signal is corrupted by additive white Gaussian noise and where the characteristics of the signal are known in advance. The method is widely used in radar, sonar, and communication systems for tasks such as target detection, signal demodulation, and channel equalization. Despite its effectiveness, the matched filter method requires prior knowledge of the signal characteristics, which may not always be available in practice. As such, researchers have developed more advanced signal processing methods to address this challenge, such as machine learning-based approaches that can learn the signal characteristics directly from data. [9]

2.5 Acconeer Obstacle Detector

As a part of their software and signal processing offering, Acconeer has implemented a simple obstacle detection algorithm, which neatly demonstrates the use of SAR for the Acconeer PCR form factor. The goal of the algorithm is to find the angle α and distance d to obstacles in front of a moving robot. The algorithm can be found in Acconeer's Python-based Exploration Tool [10]. The obstacle detector can be considered to be a simpler predecessor to what we are aiming to create.

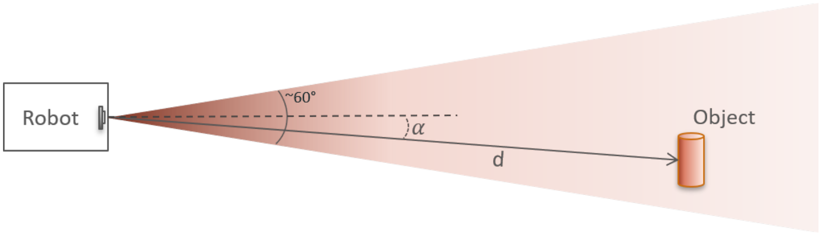


Figure 2.3: Obstacle Detector setup showing distance (d) and angle (α) to object.

The algorithm consists of several steps, including sweep collection, FFT (Fast Fourier Transform)-based Power Spectral Density (PSD), peak detection, and computing range and angle. First, a configurable number of sweeps are gathered at a set frequency while the robot moves straight ahead at a constant speed. The positions of these sweeps are stored as a VAA. Later, when a new sweep is gathered, it is added to the array, while the oldest is thrown away. The PSD step increases the signal-to-noise ratio by performing an FFT along the time-axis of the set of sweeps. The resulting FFT is plotted in figure 2.4. Each bin along the X-axis corresponds to a phase change of the obstacle, which in turn correlates to its velocity relative to the robot. [10]

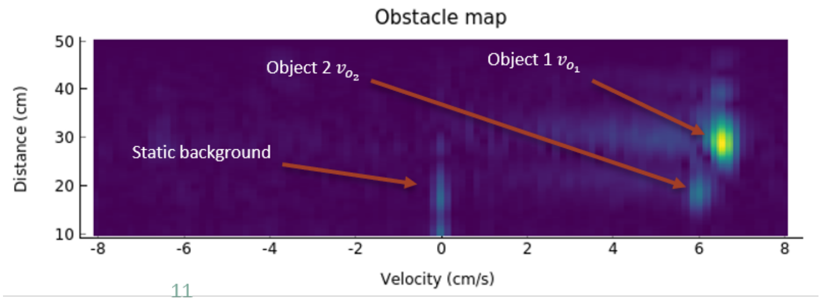


Figure 2.4: Obstacle Detector FFT showing distance and velocity of detected objects.

The peak detection step identifies potential obstacles in the filtered radar data by searching for local maxima in the PSD. Finally, the algorithm uses some simple trigonometry to calculate the angle to every detected obstacle, given its velocity relative to the velocity of the robot. The program outputs a list of detected obstacles along with their range, velocity, and angle, which can be used to guide the robot or trigger other actions. [10]

2.6 Pose Tracking

To make our system work, the radar's position and orientation will need to be tracked. A position and an orientation together make a *pose*, which is the word we will use from now on. The general concept of our idea hinges on the ability to somehow track the pose of the radar sensor with high accuracy, and at a high rate. There is a wide range of potential solutions to this requirement hardware-wise. In the initial phases of our project, though, we landed on two main alternatives, which we explain briefly in the following sections.

2.6.1 IMUs

Inertial Measurement Units (IMUs) are sensors that measure linear and angular motion using accelerometers, gyroscopes, and sometimes magnetometers. IMUs have many uses and can be found in phones, for example, or for applications in robotics, aerospace, and automotive sectors to provide information about an object's orientation and motion. The accelerometer measures a body's specific force, which correlates linearly with its acceleration. The acceleration can be used to determine the object's velocity, and even position, by integrating over time once or twice. The gyroscope measures the angular rate of the body, which similarly can be used to determine the object's orientation by integrating the time. However, integrating discrete, noisy acceleration readings is, in practice, unstable. IMUs are usually subject to drift and noise, leading to errors in the estimation of velocity and position that increase over time. To address these issues, various calibration and filtering techniques have been developed to improve the accuracy and stability of IMU measurements. Still, in practice, highly sophisticated and expensive IMUs are needed for accurate pose tracking. When used for motion estimation, IMUs are often used in combination with other sensors, such as GPS or visual odometry, to improve accuracy and robustness. Overall, IMUs are a powerful and versatile tool for measuring motion in a variety of settings, and ongoing research aims to improve their accuracy and performance further. [11]

2.6.2 VR Pose estimation

One integral part of a convincing Virtual Reality (VR) experience is fast and accurate pose tracking. Pose tracking of a VR Head Mounted Display (HMD) has to be very precise since even just a small imperfection, such as drift or latency, could make the user nauseous. This is because the human sense of balance is very sensitive, and any decoupling between what the inner ear senses and what the eyes see is noticeable and uncomfortable. There are several ways to implement pose tracking for HMDs. The Valve Index HMD, for example, uses light-house tracking with four base stations. These base stations use infrared lasers to track photonic sensors on the HMD at a rate of over one hundred measurements per second. The result is a sub-millimeter tracking accuracy. The position data of the HMD in world space is very easy to access through any compatible game engine. The tracking is specifically made to be used and accessed in game development, meaning no extra calculations are required to get coordinates from sensor data. [12]

3 Methodology

3.1 Design concept description

As described very briefly in section 1.1, the goal of the project was to achieve three-dimensional obstacle detection capabilities from a radar-based system in arbitrary motion. The overarching design concept for solving this problem contains many parts, both in software and hardware. The general idea was to integrate some type of pose-tracking with the radar system, with the objective of tracking the arbitrary movement of the radar sensor. Synchronizing the data from the pose-tracking system with measurements made from the radar system would create the basis for implementing a SAR-based obstacle detector. Within chapter 3, the design considerations for every part of the resulting system will be described.

3.2 Implementing the Matched Filter method

To maximize the signal-to-noise ratio of the radar data, we decided to use the matched filter method, described in section 2.4. The matched filter method detects the presence of a known signal in a recording of an unknown signal. The known signal is referred to as *template*, and the recorded signal consists of an unknown signal with additive stochastic noise. The cornerstone of the algorithm is the equation

$$y(\hat{s}) = \sum_i \sum_n |x(i,n) \cdot h(i,n,\hat{s})^*| \tag{3.1}$$

where, for some test location $\hat{s}$, the resulting $y(\hat{s})$ is the matched filter output of the signal x, which is a set of radar sweeps, indexed in slow-time by i. Each sweep is in turn a set of pulses, indexed in fast-time by n, at different sampling points. The template $h(\hat{s})$ is indexed by i and n like x, and also by the test location $\hat{s}$. The $*$ denotes the complex conjugate. The matched filter output is a correlation value; a high value for the correlation means high content of the template in the recording, meaning the test location $\hat{s}$ is likely the location of an object. [9]

Some corners can be cut using the matched filter method to speed up computations at a slight cost to data quality. Equation 3.1 sums both over the pulses inside a sweep, called fast-time, and each sweep belonging to the signal, called slow-time. This double summation is slow, and most of the mathematical operations only correlate noise eventually canceling out to 0 anyways, so skipping these calculations would be useful. For a template sweep generated assuming an object at distance d_0, we would only like to correlate the simulated pulse at that distance with the pulse for that distance from the recorded signal. The resulting equation only needs to sum over slow-time:

$$y(\hat{s}) = \sum_i |x_o(i) \cdot h_o(i, \hat{s})^*| \tag{3.2}$$

This equation assumes x_o and h_o are the signal and template respectively, extracted for the appropriate distance, namely the distance to the test location $\hat{s}$.

3.2.1 Simulating radar data

The template consists of simulated data. Given an object's position in relation to the radar sensor, the predicted radar signal (the template) can be generated. Given an object o at a distance d_o from the radar sensor, the time-of-flight for the signal is $\tau_o = \frac{2d_o}{c}$. For each data point in a sweep, the radar is configured to measure the signal given some delay τ, corresponding to some distance $d = \frac{c \cdot \tau}{2}$. So: given some τ, and some τ_o, the expected radar signal $z(\tau, \tau_o)$ can be generated through the following equation:

$$z(\tau, \tau_o) = e^{2\pi i \cdot f \cdot \tau_o} \cdot E(\tau - \tau_o) \cdot D(\tau_o) \cdot A(\Theta) \tag{3.3}$$

where f is the carrier frequency (60GHz), $E(\tau - \tau_o)$ is the envelope of the pulse, $D(\tau_o)$ is the distance falloff, and $A(\Theta)$ is the angular falloff given an angle Θ to the object.

The envelope $E(\tau - \tau_o)$ of the pulse is roughly triangular and has a width of 14cm. This means a perfectly infinitesimal point reflector, with no noise in the signal, will give off a triangular radar response. We found, however, the shape of radar responses in the real world to be more similar to a normal distribution, and therefore simulated the envelope as a normal distribution, pictured in figure 3.1.

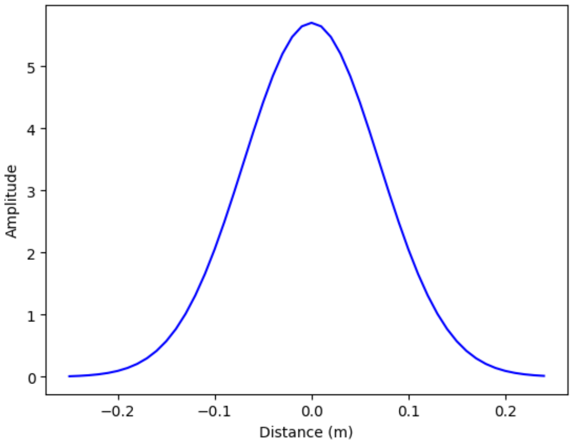


Figure 3.1: The plot of a simulated normal distribution envelope of a signal

The distance falloff $D(\tau_o)$ of the radar signal follows the inverse-square law, meaning the intensity of a signal is inversely proportional to the distance the signal travels squared [13]. And since the effect of our signal is reflected back from an object, the inverse square law is applied twice. Once on the way to the object and once on the way back to the sensor. This double distance is however in turn cancelled out by the fact that the sensor measures the square of the effect.

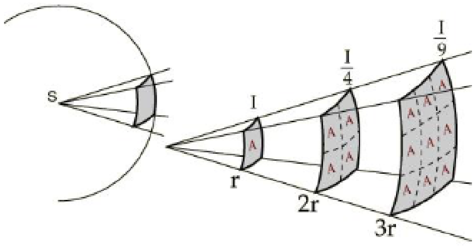


Figure 3.2: The inverse-square law [13].

This means, sampling the signal at the distance d, the simulated signal will contain the factor:

$$D(d) = \frac{1}{d^2} \tag{3.4}$$

or given a delay τ, as above:

$$D(\tau) = \frac{1}{(\frac{c \cdot \tau}{2})^2} \tag{3.5}$$

The angular falloff $A(\Theta)$ is the dampening of the radar signal intensity given some angle Θ to the object. The radar signal intensity is the strongest where the radar is pointed, and any translation of a reflector off-axis will result in a weaker signal. To model this accurately in our data simulation, we used real-world data from testing the sensor. Given a set of sample angles, we matched the angular falloff to the corresponding recorded dampening of the signal, plotted in figure 3.3. The data was collected from a setup where a Fresnel lens was used to maximize SNR at small angles. For any angles above 45°, we assumed total suppression of the signal.

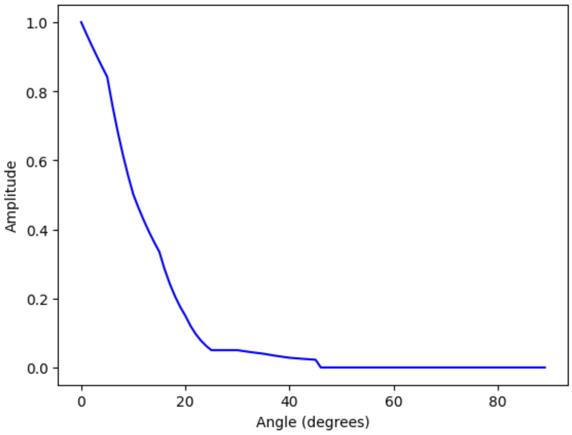


Figure 3.3: The angular falloff factor given angles between 0°and 90°.

Figure 3.4 depicts some examples of simulated radar sweeps. To produce these simulations a theoretical scene containing a reflective object at some distance and angle is imaged as a sweep of pulses by sampling points in a given range at some given interval. The individual sample points are complex-valued, but for legibility, the amplitude of the signal is plotted in the graph. The sweep in blue represents an object at a distance of 1m, head-on. The object in the orange sweep is also at 1m but has an angle of 15°, hence the lower amplitude. In the green sweep, an object is placed head-on at 1.5m, demonstrating the distance falloff.

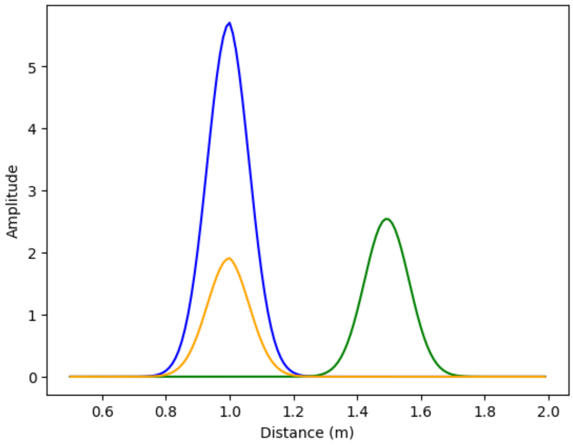


Figure 3.4: Amplitude plotted against distance for three sweeps with different angles and distances.
Blue: distance of 1m, and angle of 0°,
Orange: distance of 1m, and angle of 15°,
Green: distance of 1.5m, and angle of 0°

3.3 VR Prototype

One way we considered attempting to incorporate the algorithm described above into a working prototype was through the use of a VR system. The general idea was to mount a radar sensor to a VR headset and use the pose-tracking capabilities it offers as a basis for creating Virtual Antenna Arrays to create SAR images.

The prototype we built, presented in figure 3.5 consists, in terms of hardware, of four main parts: a computer, the Valve Index VR-system, the Acconeer XE121 radar sensor evaluation kit connected via USB-c to the computer, and a 3D-printed connection piece designed to fit the XE121 to the VR-headset. The Valve Index features a compartment at the front of the headset with a USB port that connects to the computer, designed specifically to encourage users to create their own attachments and explore the possibilities of the Index hardware. This made the design process and assembly of the prototype very easy.

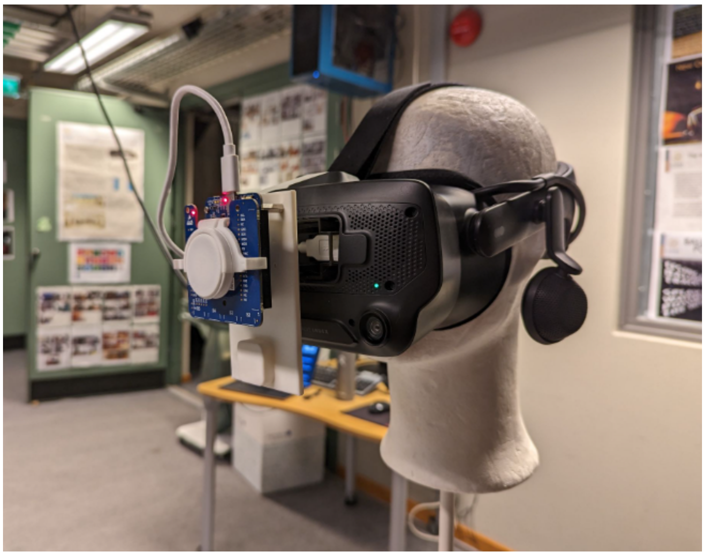


Figure 3.5: The finished VR prototype, with the XE121 PCR sensor connected through USB and 3D-printed parts to the attachment point of the VR headset

In terms of software, we used the Unity game engine as a basis for the system. To use the Valve Index hardware, we used the SteamVR plugin from the Valve Corporation which comprises pre-built VR assets and features that can be easily integrated into a Unity project. The Acconeer A121 can be controlled through Python code via Acconeer-specific software associated with the Acconeer Exploration Tool. This posed a slight hurdle, as Unity uses C# scripts to control game objects, making it incompatible with the radar sensor. To solve this, we found a way of running Python code in Unity by using an obscure plugin, the Python scripting package. This way we could control the radar sensor through Unity as if it were a game object.

Using this setup, we set up a system for recording data from the VR headset and A121 sensor simultaneously. In game development, events are often triggered each time a new frame is generated. Pose data generation in the VR headset is no different. It generates new readings for the pose at the same rate as the framerate at which the ”game” is running, which is usually about 120 Hz. Using this information, we created a script that utilizes the Python scripting plugin to connect to the A121 radar sensor. The script then, in a loop that is triggered at the time each frame is generated by the game, writes the pose data from the VR headset to a file and also tells the radar sensor to write the result of the next available radar sweep to another file.

The pose of the VR headset can be easily converted to the pose of the radar sensor. Keeping in mind that the sensor is mounted at some fixed distance in front of the VR headset, the relationship of the VR headset's pose to the radar sensor's is a simple translation in space. To obtain the pose of the radar sensor in real-world space in practice, a game object was created with a parent-child relationship to the VR headset, meaning it inherits the VR headset's coordinate system. It was then translated to the radar sensor's real-world position relative to the VR headset. The real-world pose of the sensor was then acquired by transforming its coordinates into world space. Although the poses were initially in the VR environment coordinate system, with origo at the center of some arbitrary play area, the study required the coordinate system to be defined by the starting pose of the radar sensor. To achieve this, the first recorded pose of the radar sensor was saved and later used to translate each subsequent pose into that coordinate system.

3.4 Processing data

Processing the data recorded through the use of the VR prototype (poses and radar data) was done after the fact and separately in a Jupyter Notebook. The goal was that when given a three-dimensional scene of some size, to be able to produce a three-dimensional image of that scene where obstacles are visible. To generate this image we used a three-dimensional matrix of sample locations later referred to as the search area. Each sample location was investigated for the presence of obstacles through the use of the matched filter method. The sample points were fixed in space and shared the same coordinate system as the pose data we receive from the VR-prototype. When the radar sensor moved, it moved relative to the sample point matrix. Producing an image was done by applying the matched filter method, iterating by means of a triple for-loop over each position in the sample location matrix. At each sample location, the recorded data was correlated using the matched filter method with a template that assumes a stationary reflector.

For each sampled time instance, the recorded data consisted of two parts: the pose of the radar, and the radar sweep. Using the pose of the radar, a distance and an angle towards each sample location in the grid were calculated. As outlined in the section 3.2.1, this is everything needed to simulate a radar pulse for the location in question. These calculations took place in slow-time. Given the calculated distance to the sample point, the slow-time sweep was indexed for the fast-time pulse correlating to that distance. Simultaneously, a pulse was simulated for the given distance and angle. The recorded pulse and the simulated pulse were then correlated by multiplying one signal with the complex conjugate of the other and taking the result's absolute value. The correlations yielded by each data point were then summed together, according to the matched filter method, as outlined in equation 3.2.

The matched filter process outlined above was repeated for each point in the three-dimensional sampling matrix. The resulting matrix contains a mapping of positions in real-world space to the correlation score associated with that position being occupied by a radar-reflective object. If a majority of the results being summed show a high correlation, then the resulting correlation will be high, and vice versa if a majority of the points have a low correlation, then the resulting sum will be low. This means that stationary objects will leave an accurate lasting representation in the image whereas non-stationary objects will just result in small amounts of noise. Thus, a limitation of the system we have developed is that it only works with stationary objects.

3.4.1 Thinning

While developing the algorithm above, some pre-processing for optimization was discovered that resulted in great benefits to performance. Generally, more data points were recorded than could feasibly be used, so some set of points had to be thrown out before applying the algorithm above. Starting out, we manually selected a time window of points we believed would generate good results. This quickly turned into a skill consisting of finding data points where there was a sufficient motion of the radar, as well as sufficient Signal To Noise Ratio (SNR) in the radar signal. Later on, though, we optimized and automated the process as described in this section. Data thinning is generally not used in traditional linear SAR, since every data point is collected at predetermined intervals. Meaning that the pose of two readings will never be the same, and no data is therefore superfluous. If a smaller data set would be desired in an effort to lessen computation time, then it is simply easier to gather fewer data instead of thinning. In the cases where data reduction is used in traditional SAR, then it's most often a case of reducing for the sake of transmission capabilities, and not removing for the sake of computation time or to remove superfluous data [14].

Thresholding

The radar sensor generally generates sweeps of good SNR when aimed somewhat directly at a reflective object. Upon recording data using the VR prototype, a lot of the radar sweeps would be generated while looking away from any object of interest, resulting in sweeps mainly containing noise. These data points do not contribute to the quality of the resulting output data, but they do consume computation time nonetheless. To counter this issue we introduce a filter to the data that selects radar sweeps containing amplitudes exceeding some set threshold, therefore removing all radar data compromised mainly of noise. There is a risk that we get rid of sweeps containing faint signals using this approach, but this is a trade-off we deemed worthwhile. To define the threshold, the maximum single amplitude of a pulse is found in all of the data. We denoted this value x_{max}. We can then define a coefficient T, which multiplied by x_{max} makes the threshold by which we filter all data. If the maximum amplitude of a sweep does not exceed $T \cdot x_{max}$, it is ignored.

The filter is defined by the equation below where X is the entire data set consisting of n radar sweeps containing i pulses, x_{max} is the maximum pulse amplitude in the data, and T is the threshold coefficient.

$$Y = \{x \in X \mid max(x) \geq T \cdot x_{max}\} \tag{3.6}$$

Hull points

Sufficient distance between the positions of the sensor is needed to derive information about the position of the object. Given a set of data points, and the number of points to be selected, it would be beneficial to find the positions that generate the largest possible shape in three dimensions. With this in mind, we developed a filter that selects the subset of data points that together form the largest possible hull encapsulating all data points in the set. Imagining all of the data in the positions data set as a point cloud in 3D space, we realized there is a set of points that make out the convex hull which encapsulates all other points.

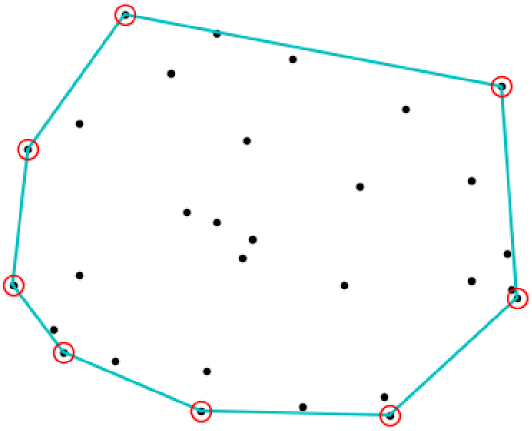


Figure 3.6: Some points in 2D, and their complex hull

After applying both the thresholding and hull point filters, the filter is defined by the equation below where X is the entire data set consisting of n radar sweeps containing i pulses, X_{hull} is the convex hull, x_{max} is the maximum pulse amplitude in the data, and T is the threshold coefficient.

$$Y = \{x \in X_{hull} \mid max(x) \geq T \cdot x_{max}\} \tag{3.7}$$

Requiring some specific number of data points, we would randomly select them from this set. This is because points sampled from this resulting data set can be assumed to generally contain larger average distances between points than a number of randomly selected points from the original set, and also have better SNR on average.

3.5 VR Prototype experiments

The following set of experiments was designed to test our approach in practice, using our VR prototype as a testbed. We wanted to prove that the design we came up with works in practice, and in general prove that the concept is viable for practical use. Thus what we set out to prove was that obstacles in the environment can be viably detected using some reasonable configuration for data collection, and data processing. The most efficient way we could test this was to attempt to prove the contrary. We designed a set of tests to prod for errors and find cases where the viability of the design falls apart.

3.5.1 Experiment setup

While experimenting with the prototype, we often used a corner reflector as a target obstacle. A corner reflector is made out of some radar-reflective material and shaped like three sides of a cube. This configuration is the basis of reflexes since it always reflects rays back toward their source. This is useful to us since we can always be sure to get decent SNR from pulses hitting the corner reflector. The image of the corner reflector is also expected to be almost perfectly shaped like a very small flat surface in 3D, showing up as a single point in the plots.

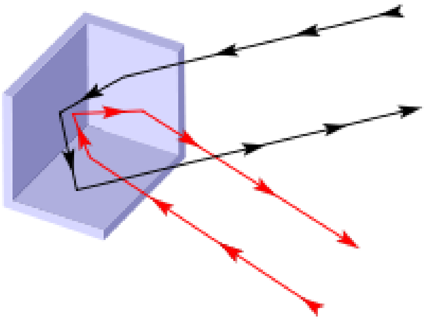
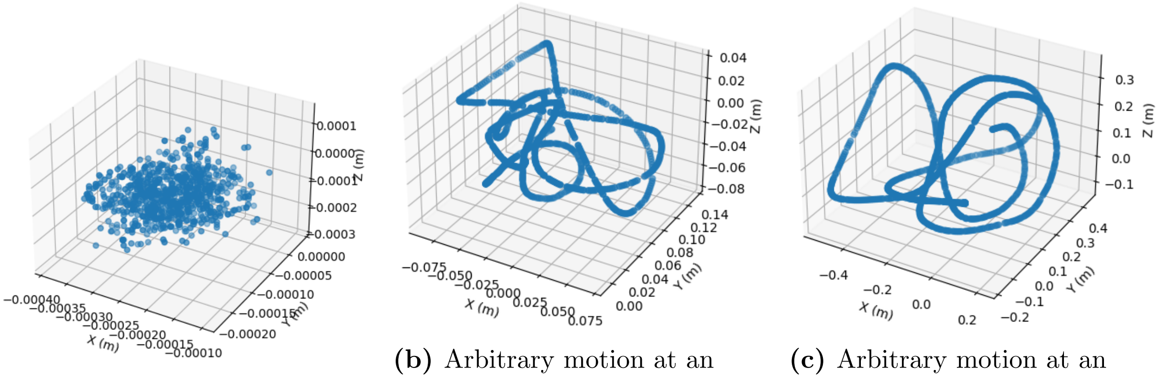


Figure 3.7: The concept of a corner reflector.

For many of the experiments defined below, we used ”arbitrary motion” when recording the data. This usually meant holding the VR headset and randomly moving it around according to some constraints. Figure 3.8 displays one example of the path taken by the headset. This set of points makes out the virtual antenna array that makes the basis for SAR-based image generation.



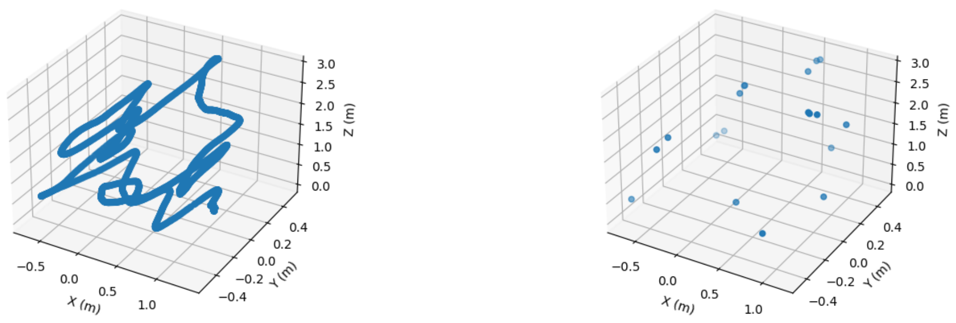
(a) No motion of radar

(b) Arbitrary motion at an amplitude of roughly 10cm

(c) Arbitrary motion at an amplitude of roughly 50cm

Figure 3.8: 3 examples of radar sensor motion plotted in 3D

Furthermore, as described in section 3.4.1, we consider only a subset of the data points in the set, discriminating on the basis of SNR and belonging to the set of hull points. Figure 3.9 shows the positions of the data set before, and after hull points thinning. As evident in the plots, the VAA can be very sparse. According to the results of experiments presented later, the algorithm even performs better with sparse VAAs.



(a) Original positions data-set.

(b) Positions data-set after thinning.

Figure 3.9: Positions data-set before, and after thinning.

Another important factor to consider when designing our experiment setup was whether or not to use a lens for focusing the radar beam. There is the option of attaching one of a few different types of lenses to the radar sensor, namely a dome-shaped lens, and a Fresnel-type lens. These lenses are designed to focus the radar beam to increase the intensity of the signal straight in front of the sensor, at the cost of decreasing its strength at increasing angles. This has the benefit of maximizing SNR for objects straight in front of the sensor, meaning the sensor can viably detect objects at greater distances, or of lesser radar reflectivity. We needed to consider this benefit and weigh it against more viably being able to detect objects at an angle in relation to the sensor. For all experiments we will explain in the following chapters, no lens was used when gathering data. We will elaborate on this later.

PCR configuration

The Acconeer A121 PCR radar sensor is possible to configure through a wide set of parameters, to where it can be fit into many different possible niches. Before any experiments evaluating the performance of the prototype, we tinkered with several different configuration parameters to find the most suitable setup for our use case.

The most significant part of the radar configuration is the sweep configuration. This defines the array of distances that are sampled by the sensor through three parameters. The *start point* defines the distance from the sensor at which the sweep starts. The *step size* sets the distance between pulses by the number of wavelengths. The *number of points* decides the length of the sweep by setting the number of pulses gathered.

What we found when experimenting with the radar configuration was that the system was quite robust to changes in settings. We found it useful, however, to define the sweep configuration such that the length of the sweep matches the sample grid in size. If the grid was to be 3 meters cubed, the radar was set to sample up to 3 meters. This appeared to maximize the likelihood of detecting objects throughout the grid.

The step length of the radar sweeps seemed quite unimportant to our system. Many different step lengths were tested for, with very little impact on the results. This was to be expected since the sampling points of the grid were usually the order of several centimeters, while the step length was in the range of a few millimeters. At the stage of processing the data, the resolution of the radar sweeps decides the mean distance to the nearest point sampled by the radar in relation to a grid point. The probability that a grid point matches up well with a radar-sampled point is larger when the resolution of the radar is finer. Using a small step length increased the performance slightly overall since there usually were radar sweeps to provide better matches at computing the matched filter correlation, but this had very little impact on the image quality.

The configuration parameters we used for the rest of the experiments are reported in table 3.1 below.

Table 3.1: PCR Configuration

Parameter	Value
Sweeps per frame	1
Start point	80
Num points	321
Step length	1
HWAAS	16
Profile	3

3.5.2 Lens choice

To determine whether or not to use a lens for the radar sensor we performed four data readings with the sensor, all contained in a 3m cubed search area pictured in figure 3.10. In the first two, a corner reflector was placed at a depth of 1m, with a slight offset above and to the left of the center. We performed one reading with a Fresnel lens and one without it to find out which will produce the better image. We then switched the corner reflector for a chair and repeated the data collection with, and without the Fresnel Lens. The reason for this was to get a sense of how the system will perform in a setup more analogous to a real use case. The results of the chair test would arguably be more important than those of the corner reflector in the VR-related use case.

Upon processing data, the same template signal was used, regardless of lens choice. In regards especially to angular falloff, the template is only an approximation of the real-world signal. The behavior we want the angular falloff factor to contribute is not a very precise art but rather objects at an angle generally producing lower amplitude signal responses.

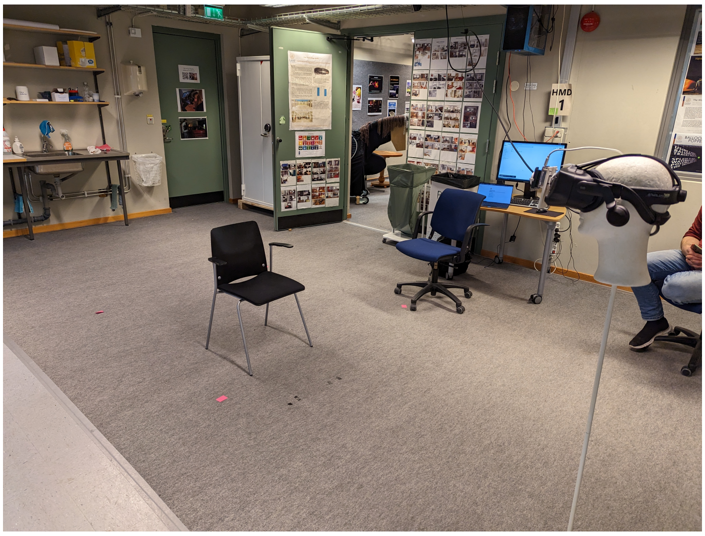


Figure 3.10: The experiment setup used for the lens experiments.

3.5.3 Data set size

The motivation for this experiment was two-fold. Firstly, we quickly noticed that a large factor in computation time while processing the data was the number of data points we run the algorithm on. It would be beneficial to minimize the number of data points while still retaining image quality, and we would therefore usually select a subset of points from the collected data set before applying the matched filter method and SAR imaging. We set out to evaluate at what point we find diminishing improvement to the image despite adding more data.

Secondly, we wanted to search for any unwanted behavior or issues by varying the size of the data set. If, for example, we found that increasing the number of data points would decrease image quality, that would indicate issues in the system.

The experiment for determining the minimum number of data points needed was performed by placing a corner reflector radar target in a 3m cubed search area, at a height of 1.5m, depth of 1m, and at 0.5m left of center. Some data was collected by moving the sensor in a random pattern while keeping it pointed at the reflector. We then extracted three different subsets of this data of varying sizes, namely 1, 5, and 50 data points. The results of the experiment were to be drawn from the accuracy and quality of the resulting radar images.

3.5.4 Testing for synchronization issues

At one point while working with the prototype we started suspecting issues in the synchronization between pose data and radar data, due to some unexplained errors in images produced in some experiments. Therefore, we created the following experiment to investigate the synchronization between position readings and radar sensor readings. To achieve this, data were collected using a corner reflector target located in the center of a 3m cubed search area. The sensor was moved straight towards the reflector and back while minimizing any significant deviation in height or lateral movement. As such, the recorded movement of the VR headset was primarily confined to the z-axis, which represents the direct path to the reflector. Any changes in the position would therefore result in an equal but opposite change in the distance to the target recorded by the sensor. Overlaying the VR headset's recorded movement along the z-axis and distance to the reflector, as recorded by the radar, through time in the same graph allowed us to find synchronization issues. With no synchronization issues, the system's motion and radar readings should coincide perfectly in time.

3.5.5 Radar sensor motion

This experiment was conducted to explore how the motion of the radar sensor impacts the results in terms of image quality. Generally, more motion should allow for the matched filter to more accurately place objects in 3D space, as there is more information to go on in the process of "triangulating" their positions. In the context of this experiment, a variety of measurements were conducted with the objective of comparing the resulting SAR images and facilitating analytical discussion pertaining to their quality and characteristics.

A 3m cubed search area was used in all data collection efforts, with a corner reflector positioned leftward, and slightly above the center of the area. The initial three measurements were intended to gather data pertaining to the significance of the boundaries of the motion applied to the sensor during data collection. The motion in this case, and the resulting VAAs, took the shape of a path entirely encapsulated by a sphere of some known size. We used the radius of this sphere to define the amplitude of the motion. Accordingly, one measurement was conducted with the sensor fixed in place. One was performed with random movement within a 10cm radius sphere. A final measurement was conducted with random movement within a 50cm radius sphere. The data from these measurements will make the basis for the discussion on the motion's importance in generating SAR images.

Two additional readings were conducted with the same basic setup, except that the constraint to move solely along the x-axis was placed upon the motion of the sensor. The measurements were then conducted with the sensor moving back and forth 10cm and 50cm, respectively. The purpose of these two readings was to assess any potential impact on the resulting images if the motion was isolated to one axis. The hypothesis was that the image would work well along the depth, and the x-axis, but not produce accurate imagery of the obstacle along the y-axis.

3.5.6 Radar sensor velocity

We had some doubts that the system would stay coherent, and produce accurate SAR images if the sensor were to move at high speeds during data collection. This doubt came firstly from the previously mentioned worry of the system containing synchronization issues and in part from basic radar principles. The phase of the received radar signal will move more than π between readings if the relative motion of the radar and the object is more than one-quarter wavelength (1.225mm) in the duration between sweeps. Systems that rely on the coherency of the radar signal for velocity estimation, therefore, are constrained by the sweep rate of the radar. In our case, running at 120Hz, the velocity at which we might encounter issues is, therefore, around 0.3m/s.

To determine the effect of the movement speed of the sensor on the produced SAR image, we performed three rounds of data collection inside the 3m cubed search area, with a corner reflector in the center. Data was gathered while moving the sensor toward the reflector target at three different speeds, namely: 1m/s, 1.5m/s, and 2.5m/s. The resulting images will be used as a basis for evaluating whether our system is constrained by movement speed.

3.5.7 Imaging common objects

Finally, to test the performance of the system in environments more akin to those a VR user might encounter in real life, we wanted to create some images of obstacles that might be found in a living room for example. From this experiment, we can evaluate how useful the technology we developed might be in the use case of an obstacle detector in a VR setting. One question we want to answer is how the system reacts to obstacles of different materials and therefore different radar reflectivities.

Three rounds of data collection were performed in a 3m cubed search area. The first reading is of a metal clothes hanger shown in figure 3.11, the second of a single chair made out of metal and wood shown in figure 3.12, and the third of two chairs, one made of metal and wood, and one made of plastic and fabric, shown in figure 3.13.

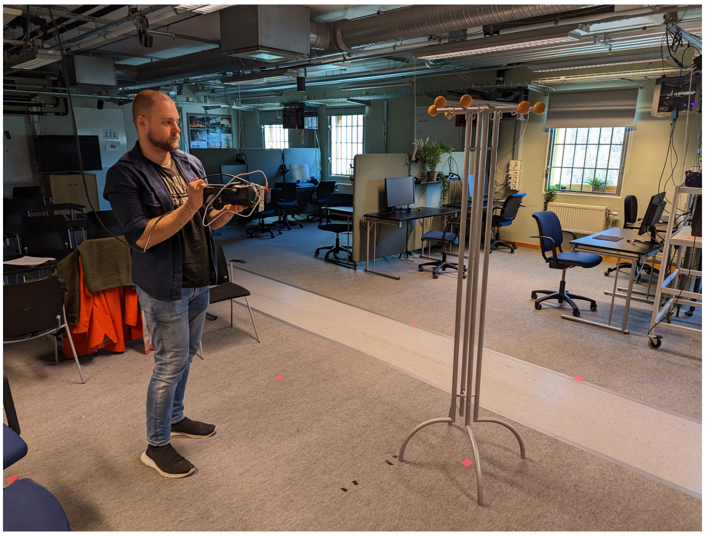


Figure 3.11: Experiment setup for the metal clothes hanger.

3.6 IMU-based prototype

The initial idea was to use an Inertial Measurement Unit (IMU) for pose tracking, with the benefit of being able to develop a prototype with applicability for a wide

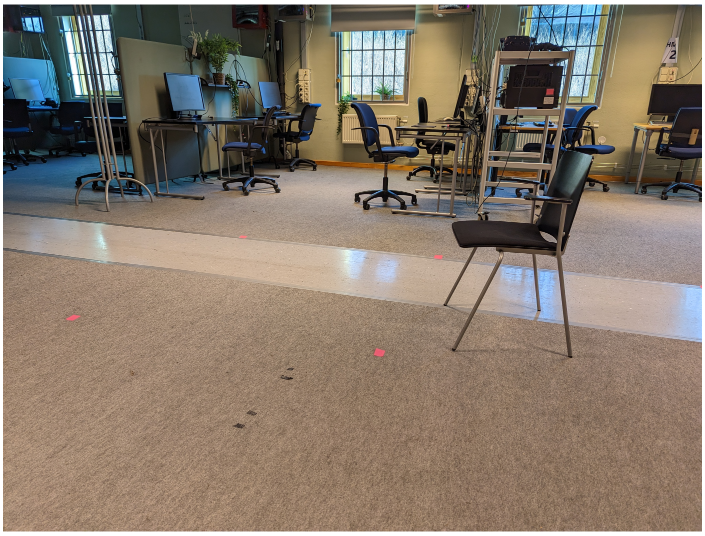


Figure 3.12: Experiment setup for one chair.

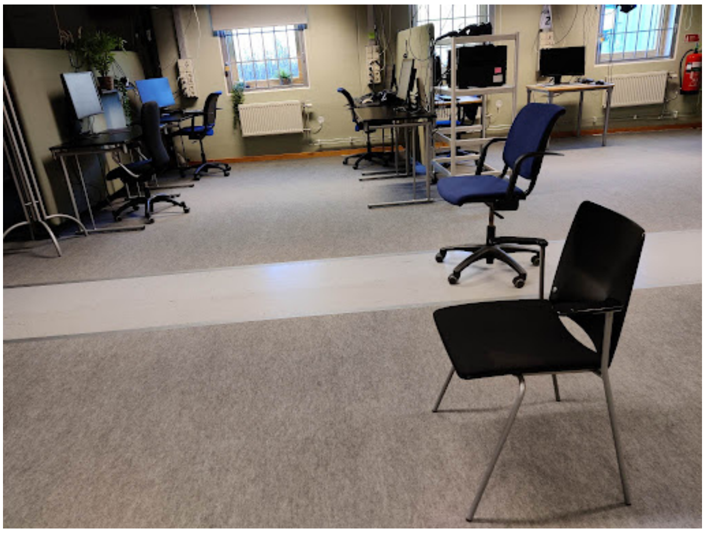
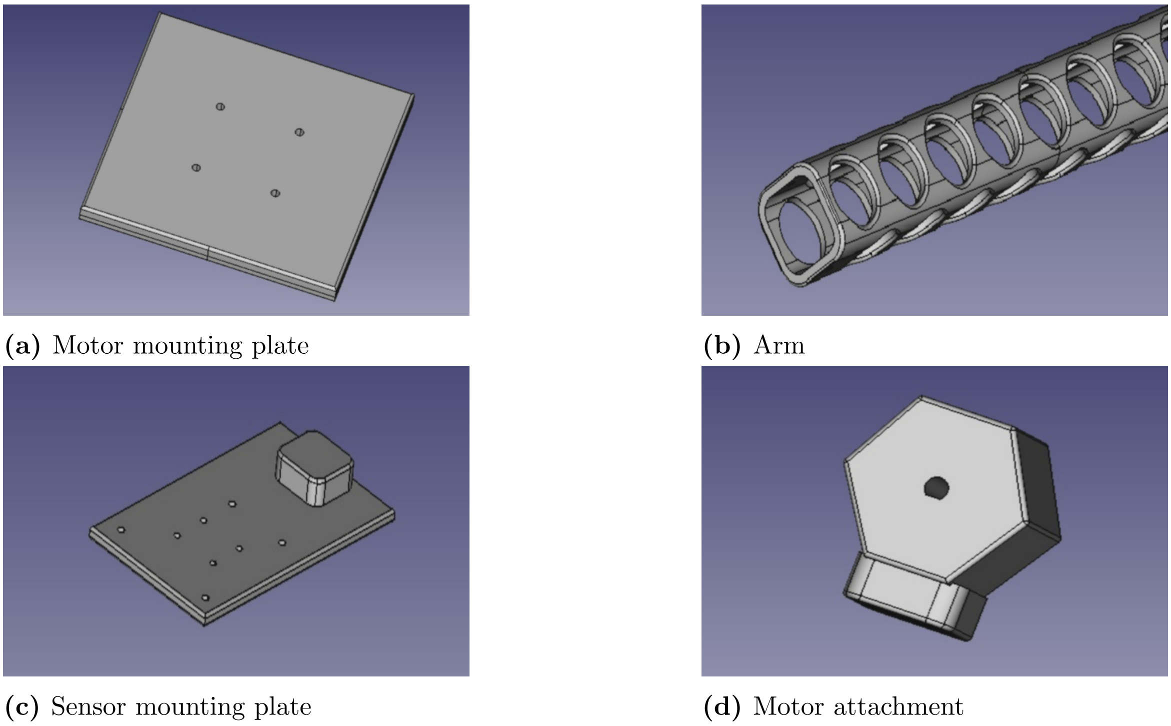


Figure 3.13: Experiment setup for two chairs.

range of use cases. This did not work out well for reasons we will elaborate on later, so this section will be kept very brief.

To develop this system we considered it necessary to create a testbed to evaluate the prototype as we went along, and aid in running controlled experiments. The design we came up with integrated a stepper motor and some 3D-printed parts to mount the radar sensor, and an IMU to a swinging arm. Using the stepper motor we could control the motion of the sensor and gyro with high precision along a circular arch. Everything would be controlled by software running on a Raspberry Pi.

The motor was powered by an external power supply and controlled by a motor control unit connected to the Raspberry Pi. The motor was mounted to a 3D-printed plate, shown in figure 3.14a, that was clamped to a table for stability. A 30cm long 3D-printed arm, shown in figure 3.14b, with a 3D-printed mounting plate for the sensors, shown in figure 3.14c, was attached to the stepper motor using the 3D-printed attachment piece shown in figure 3.14d. The final setup can be seen in figure 3.15.



(a) Motor mounting plate

(b) Arm

(c) Sensor mounting plate

(d) Motor attachment

Figure 3.14: CAD images of the 3D-printed parts used in the setup

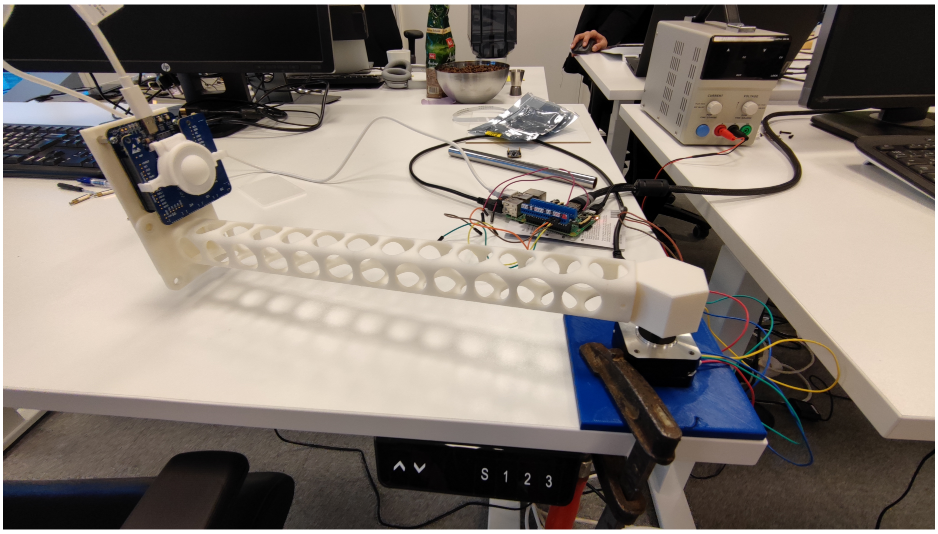


Figure 3.15: The final setup for IMU experiments, with the radar sensor and IMU mounted to the stepper motor via the 3D-printed parts

We planned to test whether the IMU-based approach worked by moving the arm around an arch at a constant angular velocity while logging readings from the IMU and radar sensor simultaneously. Since the length of the arm is known the location of the sensor can be calculated at any point during this timeframe, serving as ground truth to the IMU-based position estimations. These position estimations would have to be derived from double integrations of the acceleration readings from the IMU over time. The estimations for positions could then be used in conjunction with readings from the radar sensor in the matched filter-based SAR algorithm described in section 2.4.

4 Results

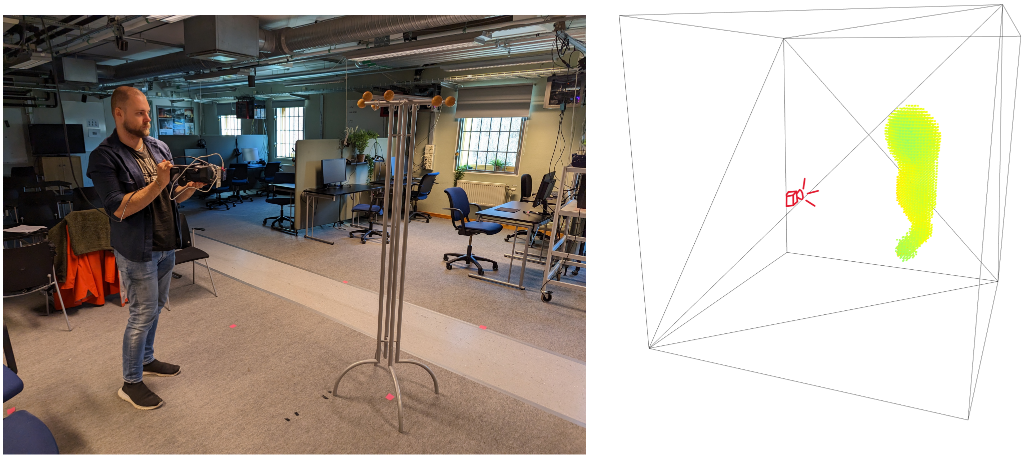
4.1 VR-based prototype experiments

Throughout the results section we present a lot of data through a type of diagram this section will explain briefly. The images produced through the use of our algorithm are three-dimensional, which poses the question of how to best represent them in 2D. The three-dimensional image consists of correlation values returned by the matched filter method for each point in space. In the location an object was placed in real life, there will be a large correlation value. This is demonstrated in figure 4.1, where we can see the results of producing an image of a clothes hanger. In figure 4.1b, the object is represented as a three-dimensional point cloud. To produce this image, a threshold was applied to the correlation values in the entire sampling grid, with the result being only the points of high correlation showing up as a point cloud, revealing the shape of the clothes hanger.

Remember that the 3D sample point grid is defined from the initial position the headset takes when initializing data collection. The sample point grid will take the shape of a cuboid stretching some distance out in front of the headset, and an equal distance above and below, as well as to the right and left. To illustrate this, the initial position of the headset is shown as a camera in figure 4.1b.

To be able to present the results in two dimensions we use two-dimensional cross-sections from the three-dimensional matrix of sampled points. Along each axis, of which there are three, there is a number of cross-sections defined by the resolution of the grid. In a grid with 30^3 points, a commonly used resolution, there are 30 cross-sections along each axis. Along each axis, all cross-sections, being two-dimensional matrices, are summed element-wise into one matrix. In figure 4.2 we see an example of how the technique we settled on works. One can intuitively think of the resulting images as looking through the 3D sample point grid if it was slightly transparent. We present the three images produced of the matrix in a diagram that, in order, is analogous to looking into the sample point matrix from the side, from the top, and from straight ahead, where the radar was looking initially. These *radar images* or *SAR images*, as they are henceforth referred to, act as heat maps representing the correlation values found in the matrix, and give a general feel for the shape and location of the objects detected.

We will reflect quite a bit on the ”image quality” throughout the rest of this report. One can define the quality of the images we produce in a plethora of ways, depending on what fits the conversation. Usually, however, high quality will mean high contrast, since high contrast will commonly correlate with high SNR in the images. This holds especially when imaging a corner reflector. In examples like the one in figure 4.1, though, defining ”quality” is a bit more subjective, and usually comes down to how well the image shows the object in question.



(a) Example of a target used for SAR image **(b)** 3D point cloud SAR image

Figure 4.1: Example of experiment setup and the resulting 3D point cloud

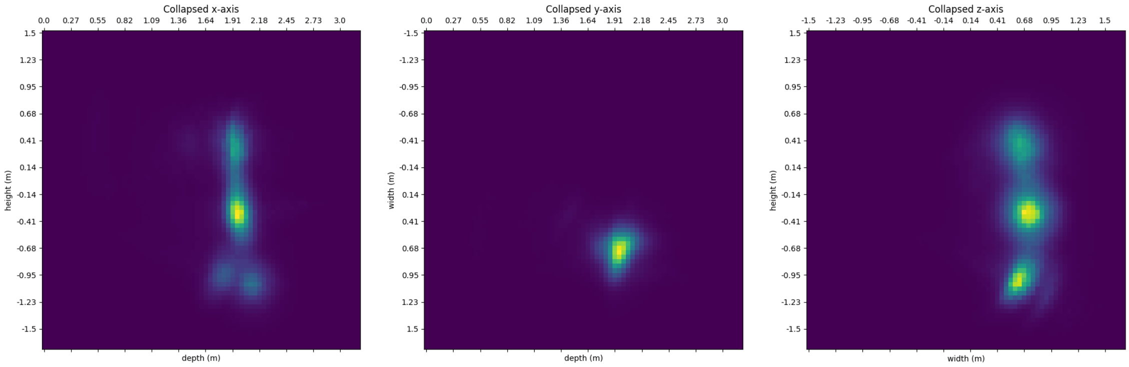


Figure 4.2: Accumulated cross-sections along each axis. First height is plotted against depth, then width against depth, and lastly height against width.

4.1.1 Lens choice

The radar images generated of a corner reflector without the use of the Fresnel lens in figure 4.3, and with the lens in figure 4.4 both display high precision in terms of the location of the reflector. The SNR of both images is fairly good, although the image taken without a lens seems to be a little more concentrated. Arguably the more important test, imaging an obstacle that might be encountered in real life, largely shows the same trend. For the radar images, we see a lot higher SNR in the image with a lens in figure 4.6, than that without the lens in figure 4.5.

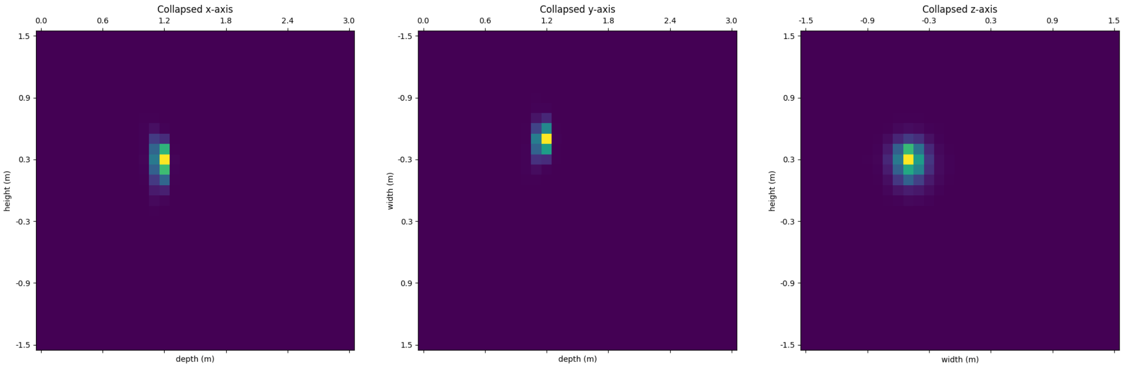


Figure 4.3: Radar image of a reflector with no lens

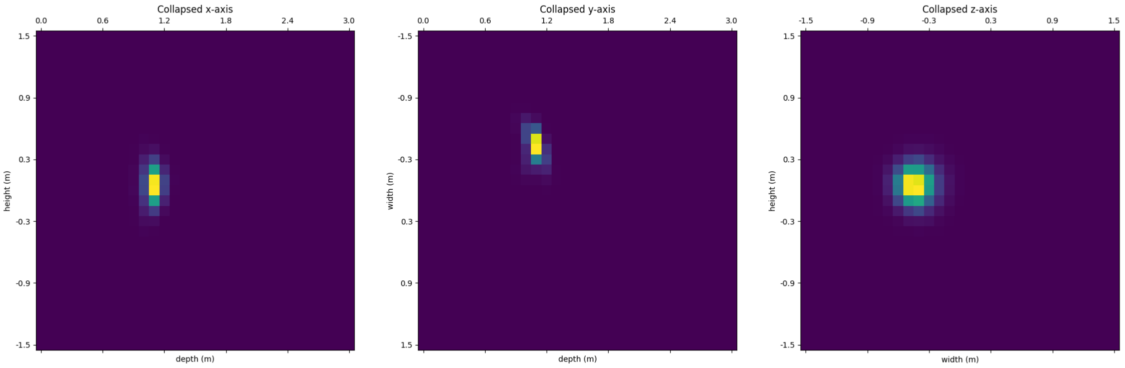


Figure 4.4: Radar image of a reflector with lens

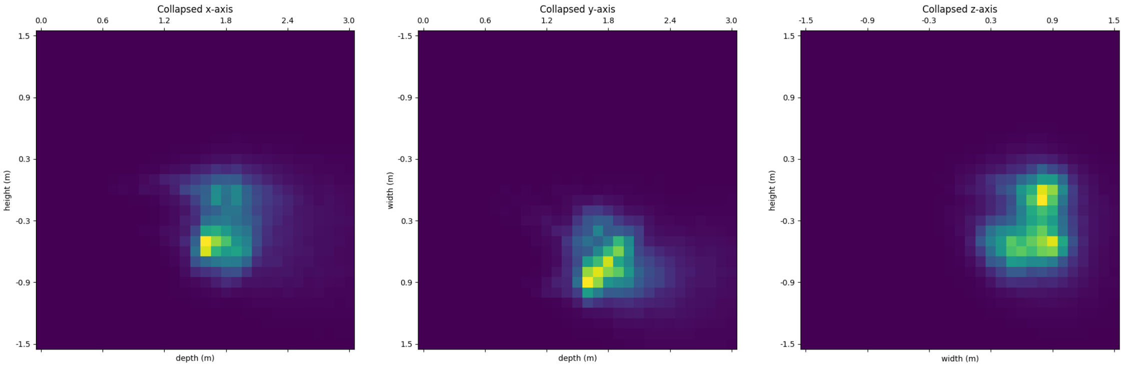


Figure 4.5: Radar image of a chair with no lens

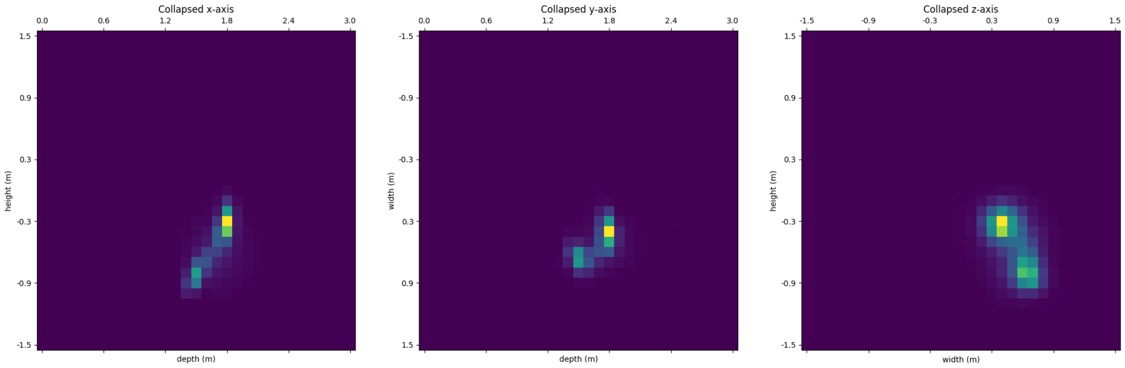


Figure 4.6: Radar image of a chair with lens

4.1.2 Data set size

The results of the data set size experiment can be found in figure 4.7 through 4.9. We produced three 3D SAR images of the 3m cubed search area. The images are represented as three 2D images according to the method described in section 4.1. Upon recording data for this experiment, the corner reflector was placed at a height of 1.5m, 1m from the sensor, and 0.3m left of the center of the space. We expect this configuration to be reflected in the images produced.

Figure 4.7 shows the image created using only a single data point, meaning the matched filter method for SAR imaging was run on one single radar sweep, taken from one single point in space. This yielded a fairly clear image of the reflector with approximately the correct depth, but the position along the x and y-axis is incorrect. The image of the reflector ended up right in front of the radar, in the center of the lowest image.

Figure 4.8 was created using five data points. As with figure 4.7, the depth is correct. In this image, however, the location of the reflector in the x- and y-axis also follow our expectations, appearing approximately 0.3m left of the center.

Figure 4.9 is very similar to the second image as the image of the reflector shows up where it is expected. With an increasing number of data points, though, the sharpness of the image increases slightly. For figure 4.9 produced with 50 data points, the footprint of the reflector is very small. This means the correlation with the largest amplitude returned from the matched filter algorithm is relatively larger than the surrounding points in this image when compared to other images.

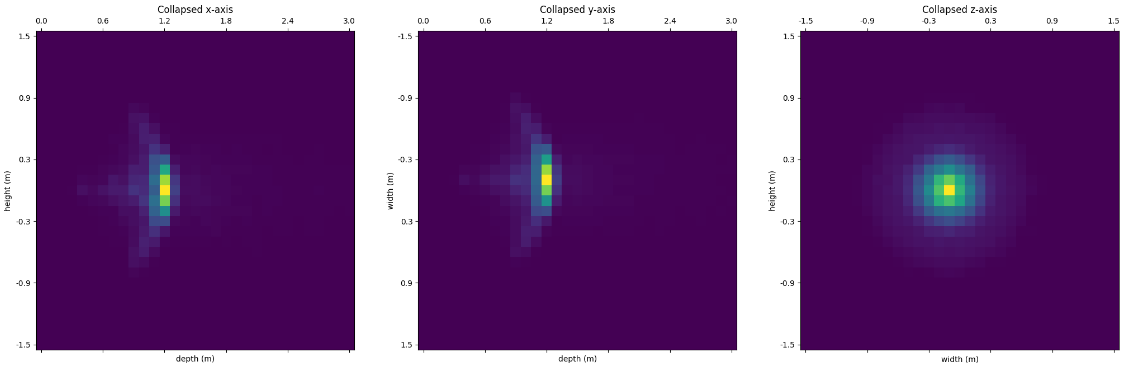


Figure 4.7: Radar image for 1 data point

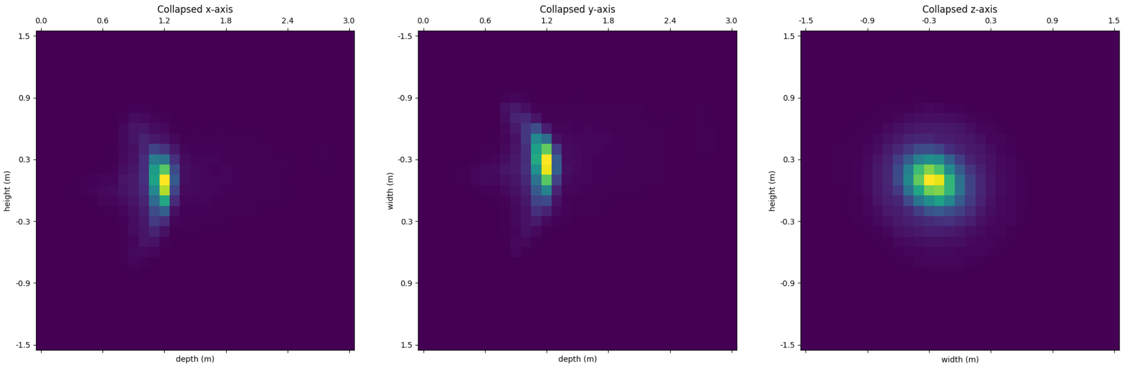


Figure 4.8: Radar image for 5 data points

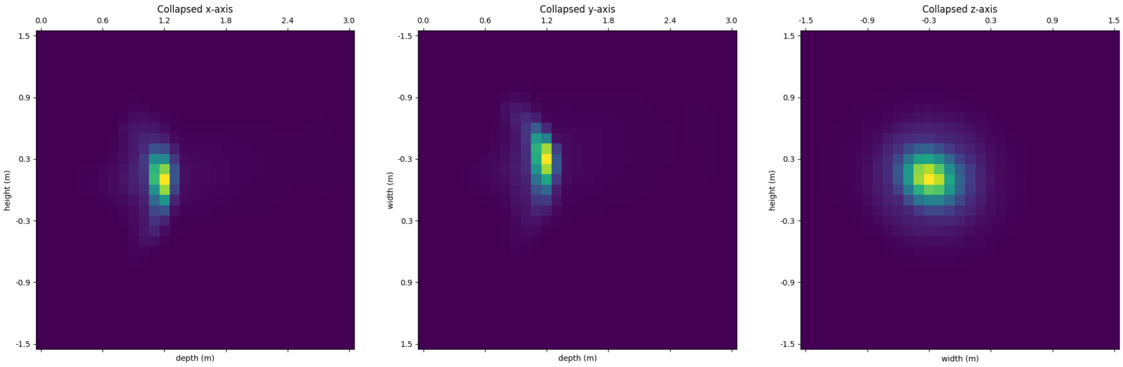


Figure 4.9: Radar image for 50 data points

4.1.3 Testing for synchronization issues

The results from the data synchronization experiment are presented in figure 4.10. As anticipated, the plot reveals that as the VR headset moves forward and the z-axis distance increases, the distance to the reflector measured by the radar sensor decreases. A blue vertical line is placed at the time instance exhibiting the maximum amplitude of the headset motion, and it aligns nearly perfectly with the time instance where the minimum distance measured by radar appears. Note however that the initial peaks do not appear to align well, while all subsequent peaks after the maximum and minimum points seem to be well-aligned. The results of the experiment show that there indeed are synchronization issues in the system, although confined to a window of time at the beginning of data recording. The synchronization error starts at about 100 frames at 120fps, amounting to a delay in the radar signal of about 0.8 seconds. After 500-600 frames, though, which equates to about 4-5 seconds, the synchronization error appears to have resolved itself.

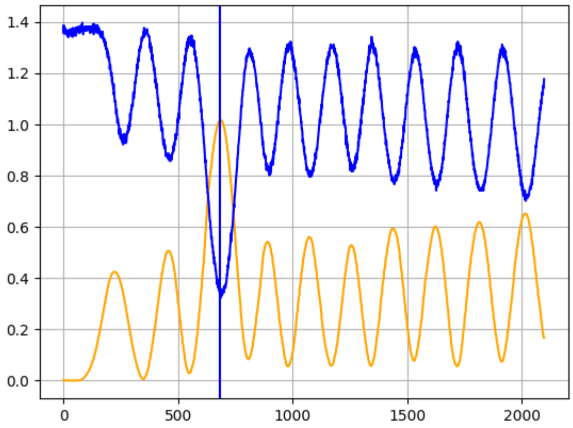


Figure 4.10: The results of the synchronization experiment. Graph of the z-axis coordinates (in orange) overlayed with the distance to the reflector measured by radar (in blue), plotted through time, and labeled by the indices of the data points.

4.1.4 Radar sensor motion

In figures 4.11 through 4.20 we see the resulting SAR images of the radar sensor motion experiments, as well as their VAAs plotted as point clouds in 3D.

Figure 4.11 shows the SAR image generated from the stationary experiment. It conveys the highest amplitude of the matched filter algorithm is the dead center of the z-axis plot, which is not the actual location of the reflector, but rather the direction the radar sensor had when collecting data.

Applying motion constrained to a 10cm radius sphere resulted in the SAR image displayed in figure 4.13. It shows the position of the reflector is in roughly the right location, but it is slightly blurry. The data collected from 50cm of motion, however, produced a SAR image 4.15 where the reflector is in the correct location, and apparently has a much better SNR.

The SAR images in figures 4.17, and 4.19 are created using 10cm of motion, and 50cm of motion respectively constrained to the x-axis of the setup. In neither image, the amplitude-maxima correlates perfectly with the reflector's position during data collection. In the 10cm case, the blob is once again placed in the dead center along the x- and y-axes. The image generated from when the sensor moved 50cm linearly, shows the correlation maxima to the left of the center, where the reflector actually was, but with the height still being in the middle.

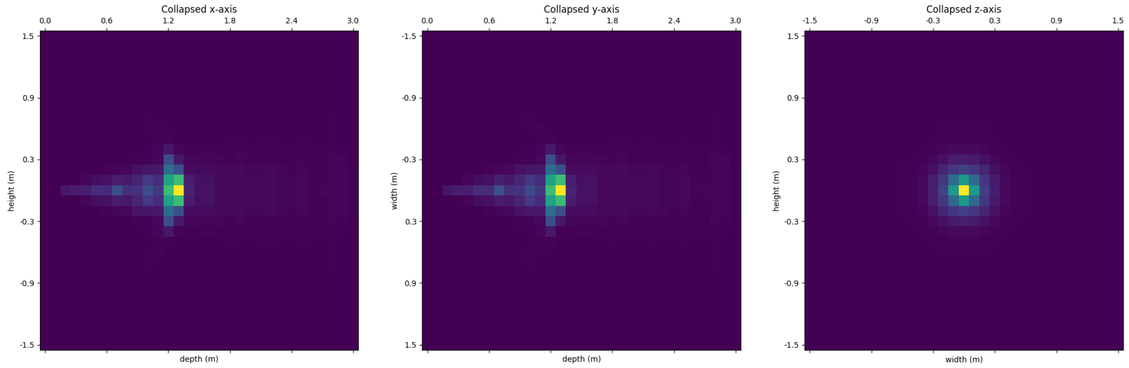
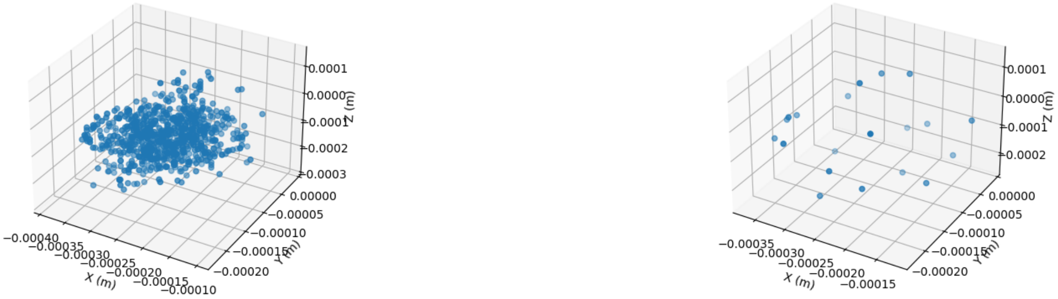


Figure 4.11: Radar image using a stationary sensor



(a) VAA candidates for stationary sensor

(b) VAA candidates for stationary sensor, after thinning

Figure 4.12: VAA for data collection with stationary sensor

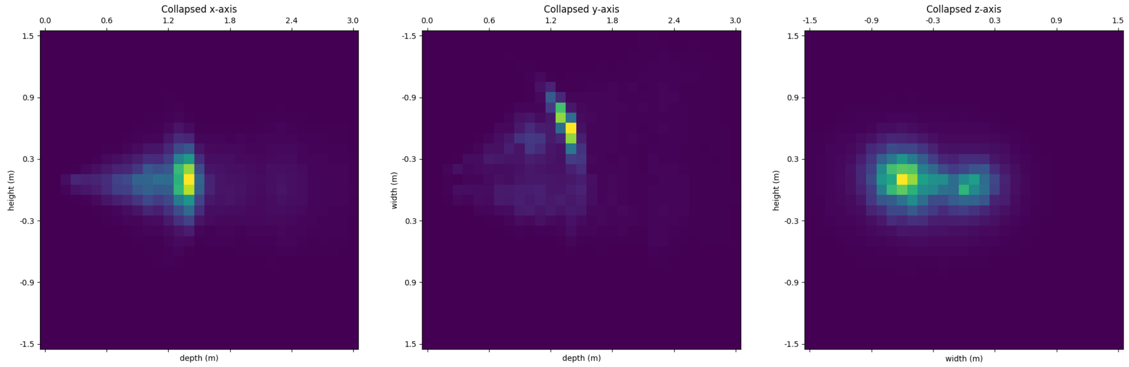
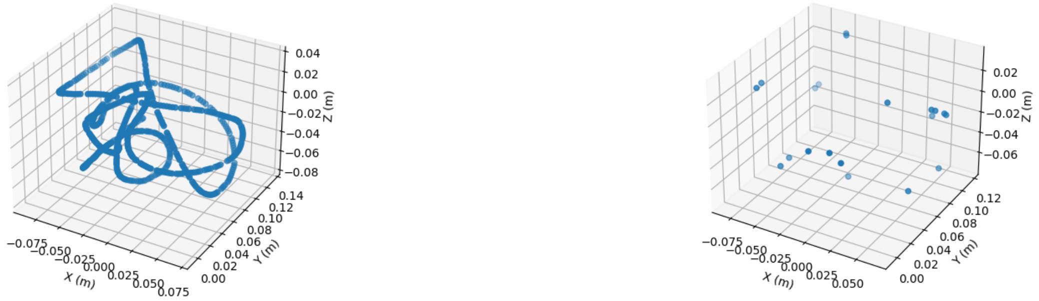


Figure 4.13: Radar image using 10cm arbitrary motion



(a) VAA candidates for 10cm sensor motion

(b) VAA candidates for 10cm sensor motion, after thinning

Figure 4.14: VAA for data collection with 10cm sensor motion

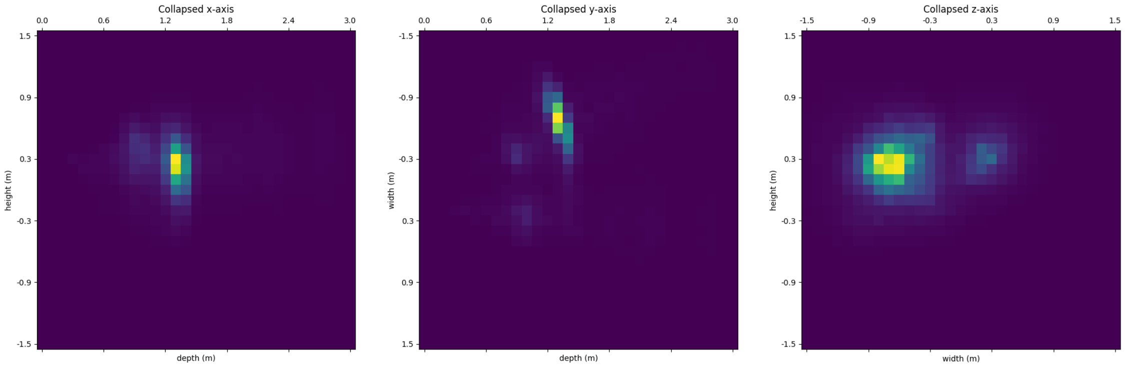
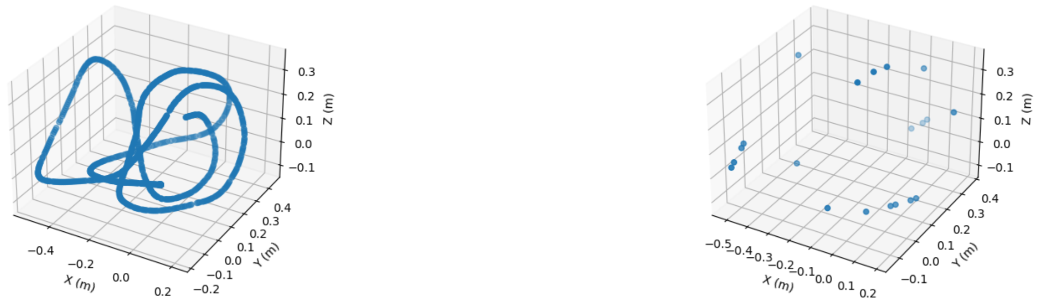


Figure 4.15: Radar image using 50cm arbitrary motion



(a) VAA candidates for 50cm sensor motion

(b) VAA candidates for 50cm sensor motion, after thinning

Figure 4.16: VAA for data collection with 50cm sensor motion

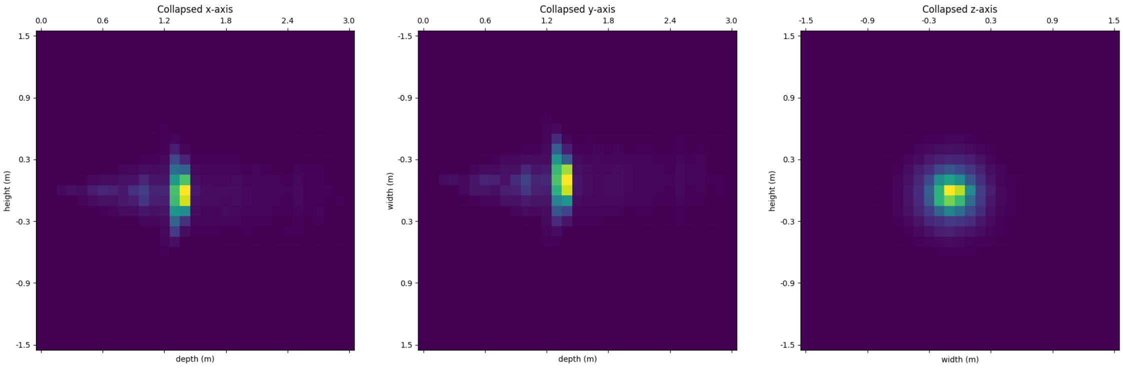
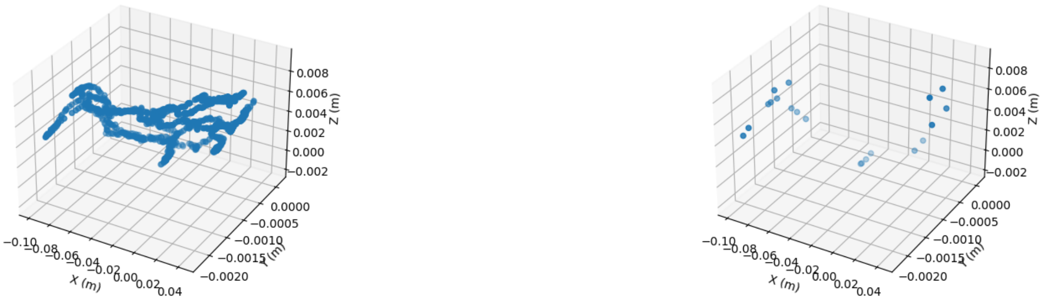


Figure 4.17: Radar image using 10cm linear motion



(a) VAA candidates for 10cm linear motion

(b) VAA candidates for 10cm linear motion, after thinning

Figure 4.18: VAA for data collection with 10cm linear motion

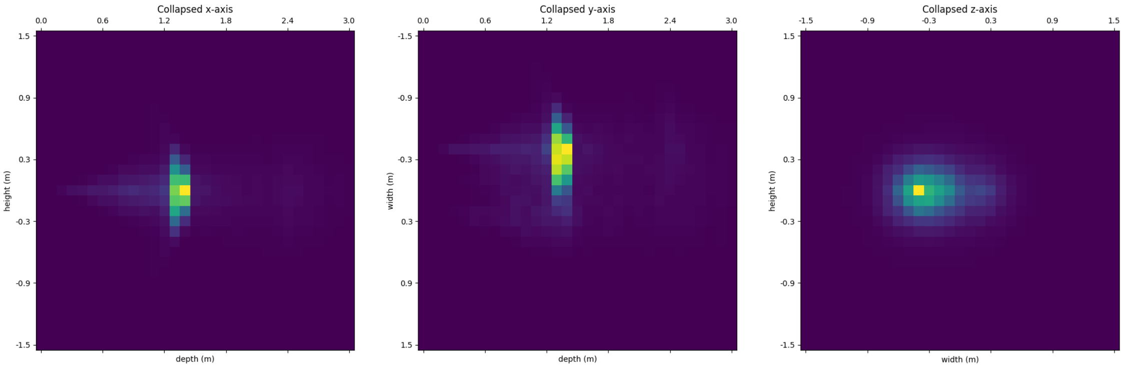
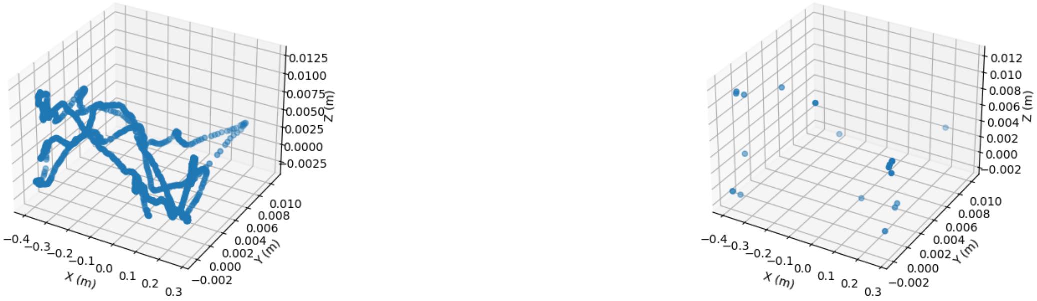


Figure 4.19: Radar image using 50cm linear motion

4.1.5 Radar sensor velocity

In figure 4.21 through figure 4.26, we see the resulting SAR images from the radar sensor velocity experiment, as well as their respective VAAs plotted as point clouds in 3D.

In all three of the SAR images, shown in figures 4.21, 4.23, and 4.25 we see an amplitude maxima confined to the location that was expected, where the corner reflector was



(a) VAA candidates for 50cm linear motion

(b) VAA candidates for 50cm linear motion, after thinning

Figure 4.20: VAA for data collection with 50cm linear motion

placed at data collection. We can however see that the dot becomes slightly more blurry as the speed is increased, hinting at a link between radar velocity and image quality.

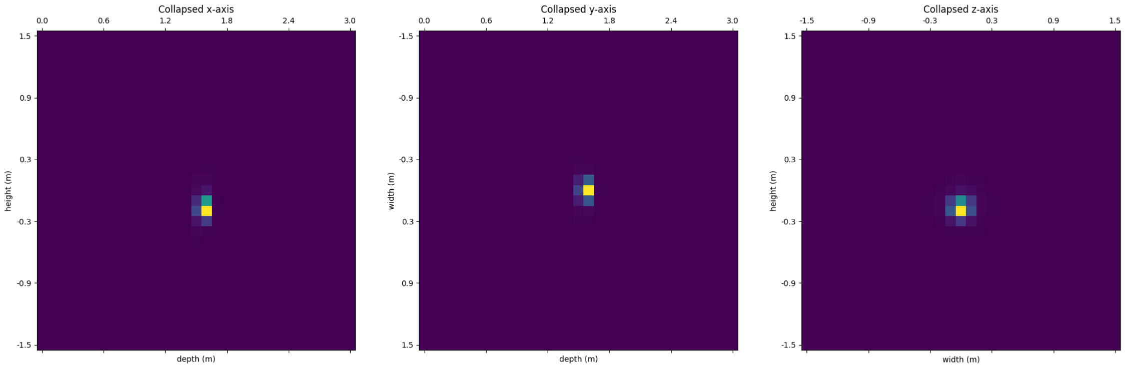
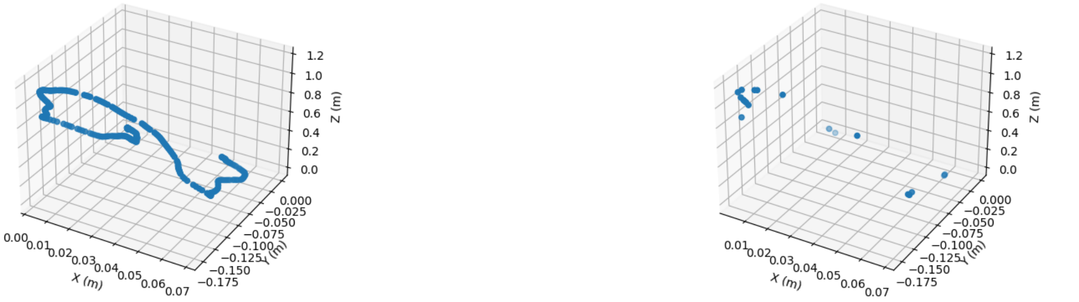


Figure 4.21: SAR image generated from data gathered when the sensor was subject to 1m/s motion.



(a) VAA candidates for the sensor at a velocity of 1m/s

(b) VAA candidates for the sensor at a velocity of 1m/s, after thinning

Figure 4.22: VAA for data collection with the sensor at a velocity of 1m/s

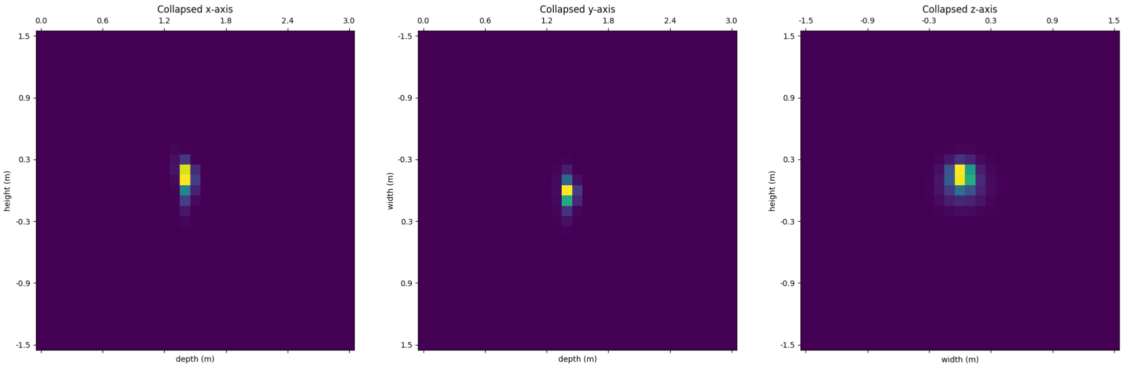
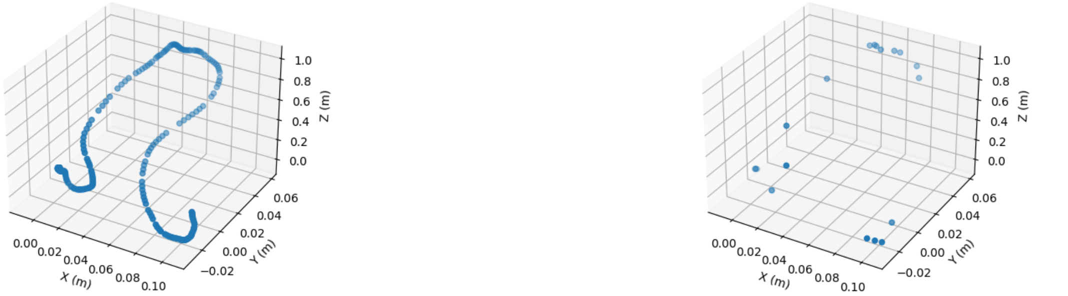


Figure 4.23: SAR image generated from data gathered when the sensor was subject to 1.5m/s motion.



(a) VAA candidates for the sensor at a velocity of 1.5m/s

(b) VAA candidates for the sensor at a velocity of 1.5m/s, after thinning

Figure 4.24: VAA for data collection with the sensor at a velocity of 1.5m/s

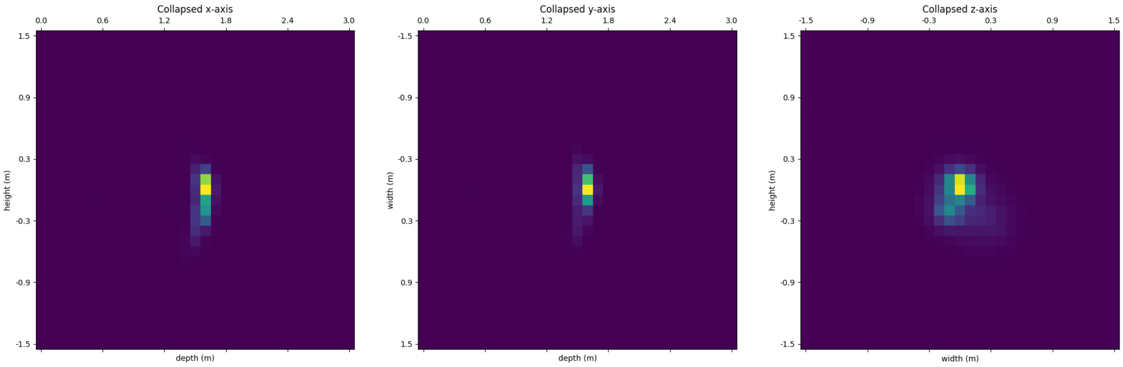
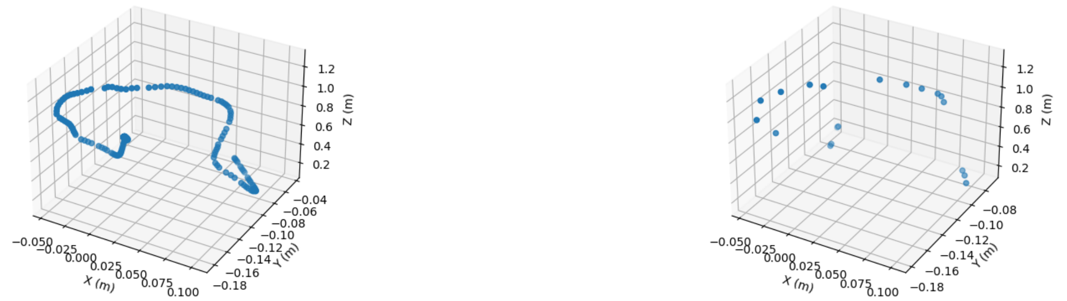


Figure 4.25: SAR image generated from data gathered when the sensor was subject to 2.5m/s motion.

4.1.6 Imaging common objects

In figure 4.27 through figure 4.32, we see the resulting SAR images from the experiment of imaging common objects, as well as their respective VAAs plotted as point clouds in 3D.



(a) VAA candidates for the sensor at a velocity of 2.5m/s

(b) VAA candidates for the sensor at a velocity of 1.5m/s, after thinning

Figure 4.26: VAA for data collection with the sensor at a velocity of 1.5m/s

In figure 4.27 we can see the SAR image of the clothes hanger shown in figure 3.11. One can clearly see a blob of approximately the right size and shape in the expected location.

The SAR image in figure 4.29 of the single chair, shown in figure 3.12, also turned out as expected. A somewhat chair-shaped blob shows up in the image, at the location we expected it to.

Creating a SAR image of two chairs, shown in 3.13, resulted in an image, found in figure 4.31, with one clearly defined blob in the location of one of the chairs, and one less clear blob also in the location of the other chair.

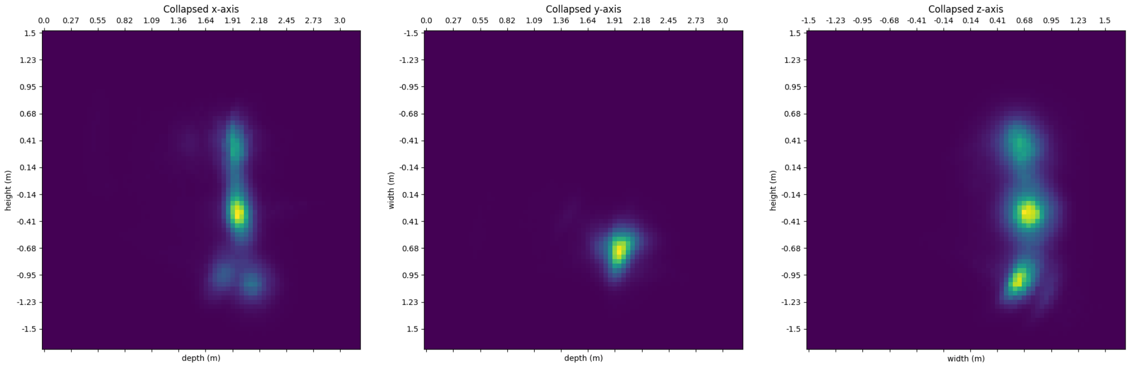
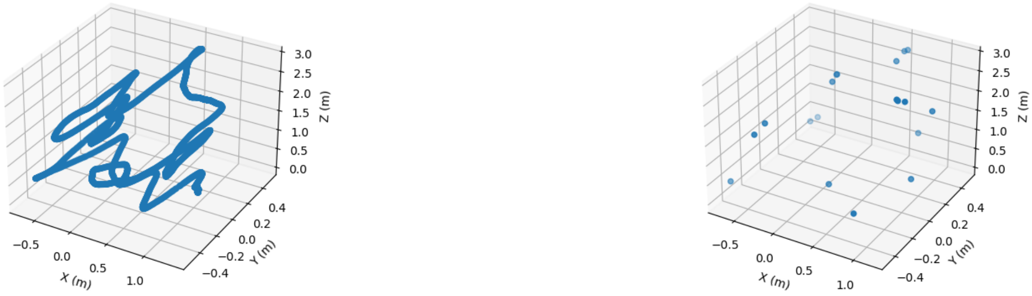


Figure 4.27: SAR image generated depicting a clothes hanger



(a) VAA candidates for the hanger SAR image

(b) VAA candidates for the hanger SAR image, after thinning

Figure 4.28: VAA for data collection to create a SAR image of a clothes hanger

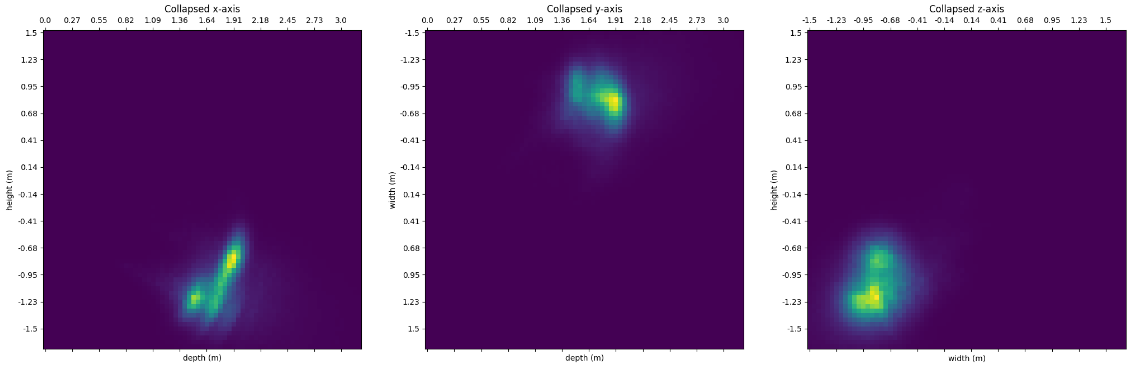
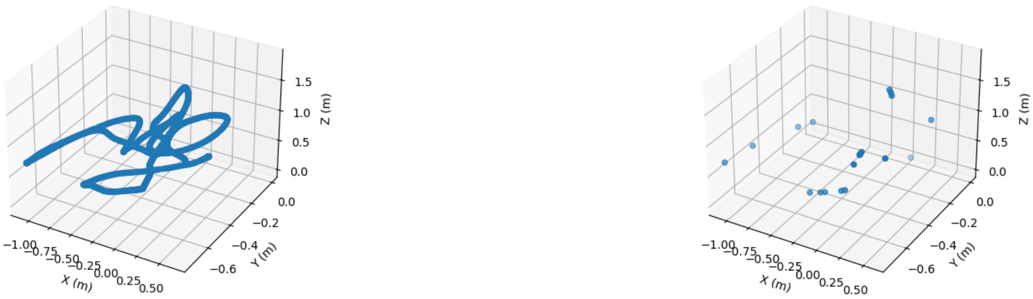


Figure 4.29: SAR image generated depicting a chair



(a) VAA candidates for the chair SAR image

(b) VAA candidates for the hanger chair image, after thinning

Figure 4.30: VAA for data collection to create a SAR image of a chair

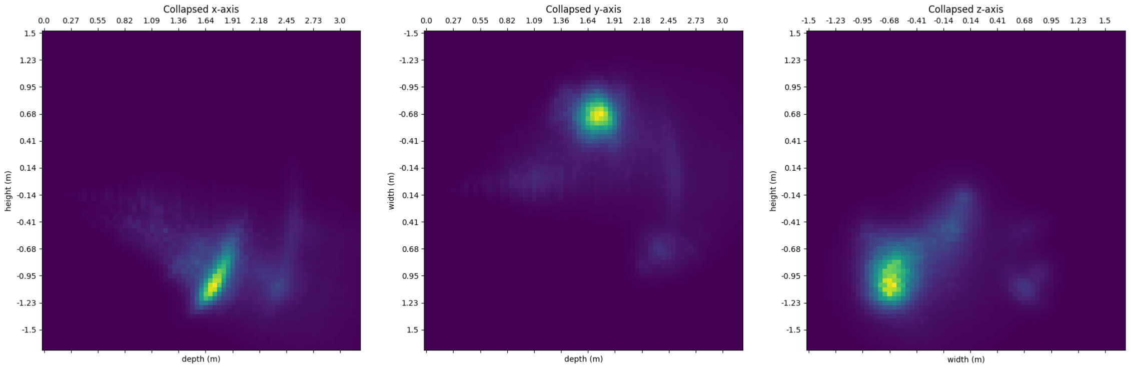
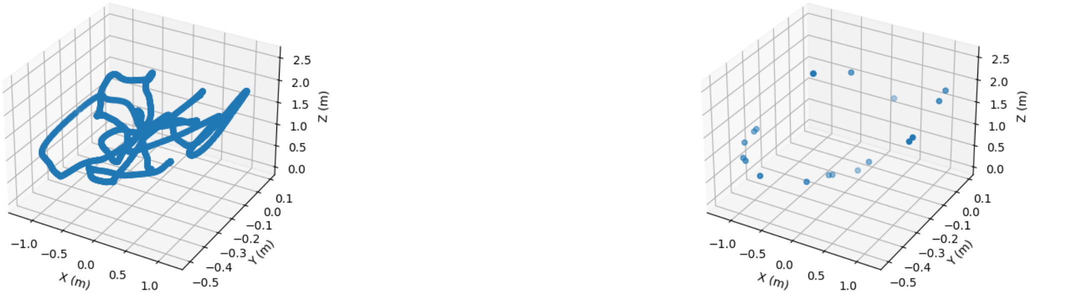


Figure 4.31: SAR image generated depicting two chairs



(a) VAA candidates for the SAR image of two chairs

(b) VAA candidates for the SAR image of two chairs, after thinning

Figure 4.32: VAA for data collection to create a SAR image of two chairs

4.2 IMU prototype testing

Quickly upon starting to evaluate the performance of our IMU-based prototype, we started running into problems stemming from the IMU. Pose estimation, or the task of determining the position and orientation of an object in space, is a challenging problem that is essential for many applications in robotics, virtual and augmented reality, and human-computer interaction. While there are various techniques for achieving pose estimation, using an IMU is often considered an attractive option due to its compact size, low cost, and real-time response. However, despite its advantages, IMU-based pose estimation is difficult to achieve because IMUs suffer from significant drift and noise, which can accumulate over time and cause the estimated position and orientation to diverge from the true values. This is due to the fact that IMUs rely on integrating acceleration and angular velocity measurements over time, which are prone to errors caused by sensor noise, biases, and external disturbances. Therefore, achieving accurate and robust pose estimation using an IMU requires sophisticated algorithms for filtering and correcting sensor data, as well as calibration and initialization procedures to mitigate sensor errors and drift.

The setup we used had the IMU connected to the Raspberry Pi recording data. We developed a script that continuously calculated the position of the IMU, using a double integral over the acceleration data reported, and printed it to the terminal. Within a few seconds, the estimated position of the IMU had drifted by several centimeters, despite the IMU being completely stationary, and the drift was accelerating. Since the tracking capabilities of the VR prototype appeared more promising. We decided at this point to not proceed with further experiments using the IMU prototype, and instead prioritize the VR prototype.

5 Discussion

In this chapter, we will use the results gathered from our experiments to reason on the viability and performance of the Arbitrary Motion SAR system.

5.1 Data set size

This experiment was conducted to examine how the amount of data provided to the matched filter method would impact the resulting images.

Based on the observed images, found in section 4.1.2, it can be inferred that a solitary data point is insufficient to accurately determine the location of an object. This stands, despite the system being given ideal conditions, having used a corner reflector as the object to be detected. The system clearly detects the presence of an object in space, but cannot place it correctly in the three-dimensional space. The highest point of correlation is the point straight in front of the radar sensor at the distance of the object. This outcome was anticipated, given that a single data point cannot theoretically provide enough information to create any meaningful results. The test represents the case of having a VAA consisting of only one point, which means the matched filter method will not be able to place the object in 2 degrees of freedom. We can observe this in our results, where the results are incorrect in 2 degrees of freedom, namely the x- and y-plane. If the system was given a VAA of two points, it could determine one more degree of freedom. If the VAA spanned some distance in the x-axis for example, the position of the object would be able to be determined in the x-plane, and vice versa. With three points, the system would be able to determine the correct location of the object in space, given the VAA has points that span some distance in both the x- and y-axes. No data was gathered to prove this concept in practice, but the theory holds. Making these measurements is left for future research.

Contrasting the radar images for 5, and 50 data points, marginal differences are discernible, although the image with 5 data points appears somewhat blurred. The gains in accuracy obtained from increasing the number of data points appear to diminish rapidly, with 5 data points producing satisfactory images. This reasoning, though, holds only for the corner reflector we used for testing. Depending on the type of object to be imaged, more data points can be required. Given some complex objects, detecting the shape of each of its features requires a small set of data points. For a chair, for example, a few data points are required to detect a leg, the backrest, the seat, etc. It should hold that the minimum required number of data points scales with the geometrical complexity of the object. Also, using more data points one can expect higher-quality images. It is also worth noting that an equal increase in data points will not always lead to an equal increase in accuracy. Since the last step of the hull point thinning, described in section 3.4.1, involves a random selection of hull points, then it can't be guaranteed that the extra points picked in the processing of one image will contribute as much as the extra points in the processing of another.

5.2 Lens choice

The use of a lens in radar technology is intended to concentrate the beam of radar waves and provide higher SNR directly in front of the sensor. Focusing the beam like this, however, negatively impacts the quality of the signal received from objects located in the periphery of the radar beam.

As is made clear throughout the images shown in section 4.1.1, using a Fresnel lens generally yields SAR images of better SNR. This is definitely a positive in regard to the performance of the system. In the images of the corner reflector, the footprint of the reflection is a lot smaller while using a lens. However, ideal conditions like looking straight at a corner reflector will not occur often in a realistic use case of the system. It is a lot more likely that an obstacle will not be perfectly in front of the radar sensor, and for it not to have such high reflectivity.

We would like to maximize the potential time an object spends reflecting the radar signal back to the sensor for detection, and one step that can be taken for this is removing the lens from the system. The results of this approach can be found in the images of the chair using no lens in section 4.1.1. In the context of our SAR imaging prototype, the inclusion of a lens appears to impede the acquisition of high-quality radar images. For the rest of the experiments, no lens was used for the radar sensor.

5.3 Testing for synchronization issues

The results reported in section 4.1.3 reveal issues in regard to synchronization between the radar signal, and radar sensor pose. With synchronization issues like this, the system cannot be expected to produce any viable results, since the VAA will be decoupled from the radar sweeps the radar sensor is reporting. Until we discovered this issue, we observed previously unexplainable noise and a general lack of image quality in our results.

We attempted to locate the source of the issue but found none. The leading theory, however, is that there is some latency in the software implementation of the VR prototype, specifically in regard to the Python scripting plugin used to control the radar sensor. However, as can be seen in figure 4.10, the issue is confined to the very beginning of the data set, namely the first 4-5 seconds. The latency of the radar signal to the pose data shrinks with time and is for all intents and purposes mitigated entirely after that period.

Due to our project nearing its deadline, a hacky workaround was created to get around the issue. The script logging pose data and radar data was made to wait 10 seconds after starting data gathering before starting to log data to the output files. This workaround is a quick fix, and not very practical for the user of the prototype but it does solve the synchronization issues. The results of applying this approach are shown in figure 5.1. Here, the first data point is delayed 10 seconds after starting data collection and therefore has no synchronization issues. The workaround worked well, and the quality of images improved significantly after applying it. There was, though, some small amount of latency left consisting of about 5-10 frames, which we then adjusted manually by truncating the datasets (the beginning of the pose data set and the end of the radar data set) to align perfectly.

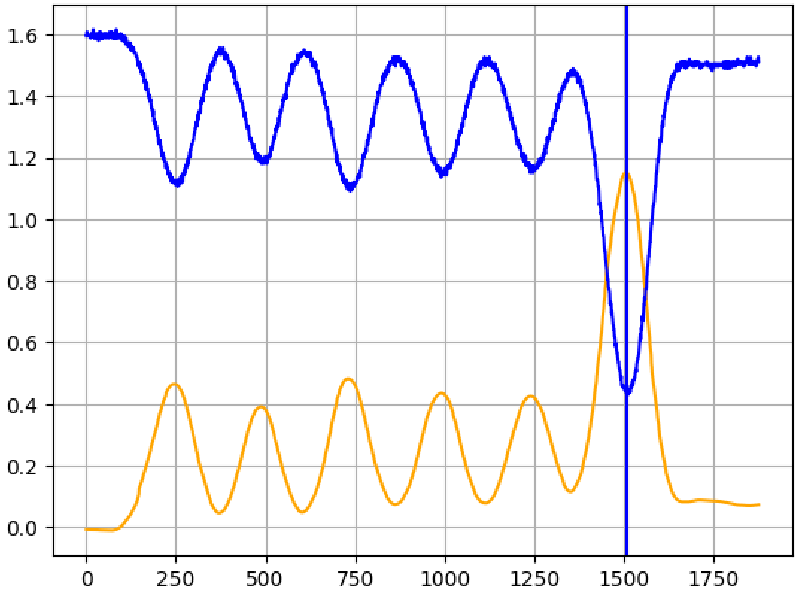


Figure 5.1: The results of applying the fix to synchronization issues. Graph of the z-axis coordinates (in orange) overlayed with the distance to the reflector measured by radar (in blue), plotted through time, and labeled by the indices of the data points.

5.4 Radar sensor motion

This experiment focuses on analyzing the effects of sensor motion on the quality of SAR images. Data was collected using the system subject to several configurations of motion, and the resulting images were evaluated to investigate the influence of sensor motion on the quality of the images. The results of the experiment can be found throughout section 4.1.4.

Firstly, we found that the stationary radar image in figure 4.11 was comparable to the single data point SAR image in figure 4.7 from the "Data set size"-experiment. The stationary image resulted in a dot in the center of the search area, straight in front of the radar sensor, despite this not being the actual position of the corner reflector in real life. The single data point SAR image displayed similar behavior. This experiment further demonstrates the importance of a VAA that spans a shape in a three-dimensional space large enough to determine the location of the object with three degrees of freedom. Otherwise, the resulting best correlation appears straight in front of the sensor as if only one data point was available for data processing.

The radar images resulting from 10cm and 50cm random motion were also analyzed. These images displayed accurate information and the correct position of the object. However, the image resulting from the 10cm movement was found to have slightly poorer SNR, resulting in a blurrier image, indicating the importance of obtaining radar sweeps from diverse locations to achieve high-quality radar images. To accurately place objects in space though, not a lot of motion is required, which is important to the viability of using this system in a realistic use case.

The study also explored the effect of limited movement on the quality of radar images, confining the motion of the sensor to only one axis. The images found in figure 4.17, and 4.19 show the radar images resulting from 10cm and 50cm linear movement. The 10 cm linear image exhibited a correlation maximum drawn out along the x-axis in the center of the image, suggesting some angle information along the traversed axis was gathered, but not enough to place the object entirely correctly. In contrast, the 50cm linear image showed the image of the corner reflector to the left of the center, where it was in real life. This provides further evidence to support the theory that the movement along one axis may influence the data quality along another axis.

Consequently, the study concludes that to obtain an accurate radar image, it is necessary to move the sensor along all axes.

5.5 Radar sensor velocity

The results of the radar sensor velocity experiment can be found in section 4.1.5.

Upon comparing the radar images at different movement speeds, it was observed that higher movement speeds appear to introduce slight blurriness to the resulting image. This phenomenon may be attributed to minor synchronization discrepancies between the recorded pose measurements and radar sweeps, which would be exacerbated at higher sensor velocities. Alternatively, it could be due to limitations of the VR hardware or radar sensors in acquiring accurate data during high-speed motion. However, the latter possibility seems unlikely as both hardware systems are designed to provide high-accuracy readings even at high velocities.

The aforementioned effect, though, is unlikely to pose any significant challenge in any practical applications of the radar system. Notably, the maximum speed tested in this study, 2.5m/s, exceeds any reasonable scenarios in at least the VR use case. Even at this speed, the decrease in image quality was insignificant. For future scenarios, though, where higher speeds may be required, a more streamlined approach that integrates the data gathering and processing closer to the hardware could be employed. This approach would enable the collection of fully synchronized data, thereby eliminating speed as a confounding factor in image quality.

5.6 Imaging common objects

This experiment was conducted to get an idea of how the system might perform creating images of objects of the type likely to appear in the VR use case.

In the first image (figure 5.2) depicting a clothes hanger, the object's overall shape is evident despite the lack of fine details. The top portion of the hanger shows higher correlations, which can be attributed to the increased number of angles and corners, resulting in more radar reflections. This observation is reasonable, and for practical purposes such as robot collision avoidance, or warning a VR user of obstacles in near proximity, this level of detail suffices, as the general shape, size, and position of the object are accurately represented.

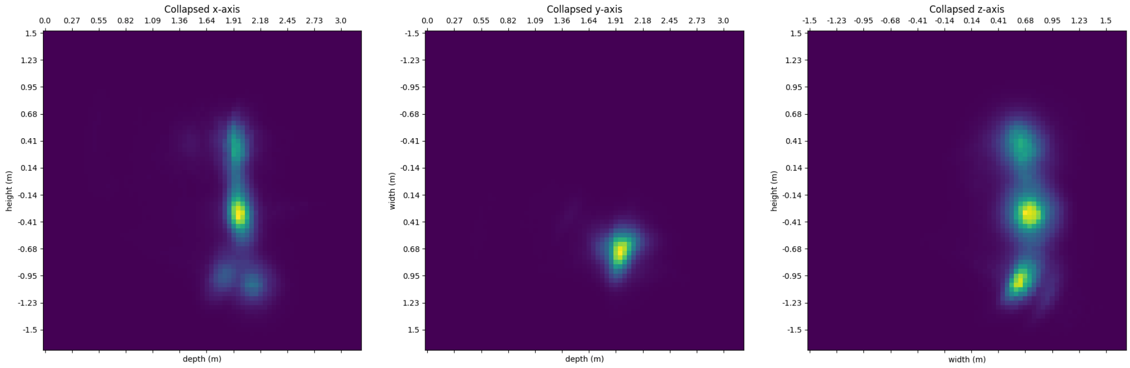


Figure 5.2: SAR image generated depicting a clothes hanger

The second image featuring the chair, displayed in figure 5.3, displays a clear outline of the object, albeit appearing slightly below the expected position. The radar image suggests that the metal legs are reflecting more than the wooden seat and backrest. This can be attributed to several factors.

Firstly, the wooden parts of the chair are less reflective of the radar beam, and will therefore appear less clearly in the image. This fact poses issues for the system in general. Obstacles composed of different materials will be hard to image well. Additionally, when creating the radar images, the correlation data is squared to increase the signal-to-noise ratio, potentially exaggerating the difference in signal strength between the metal legs and wooden parts.

Secondly, the flat seat and backrest might reflect the radar signal away from the sensor. This too poses issues. Any smooth surfaces angled away from the radar will reflect the radar signal off into the distance, and will not be detectable.

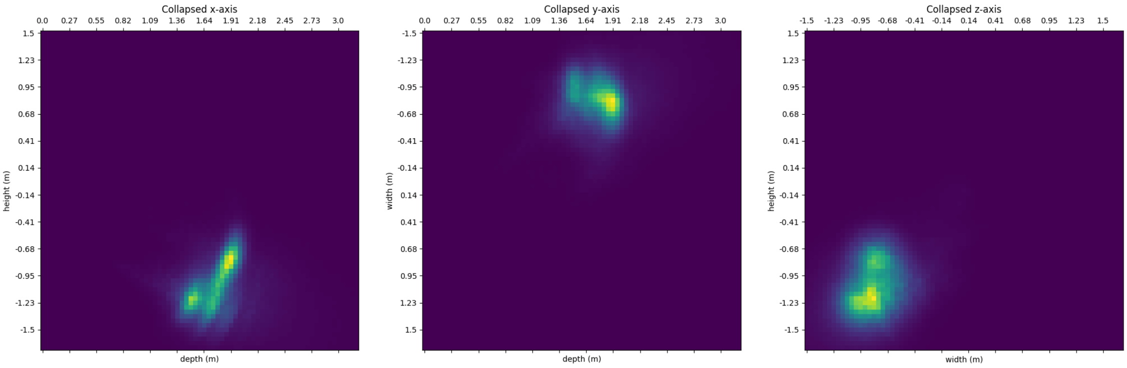


Figure 5.3: SAR image depicting a chair

In the third image depicting the two chairs, found in figure 5.4 one chair displays a visible shape in the radar image, while the other chair's trace is barely discernible. This observation supports the idea that highly reflective surfaces may block less reflective surfaces, raising the threshold for detecting the latter. However, the presence of both chairs in the radar image is a positive indication that the prototype can detect multiple objects in a search area, which is a common scenario in real-world environments.

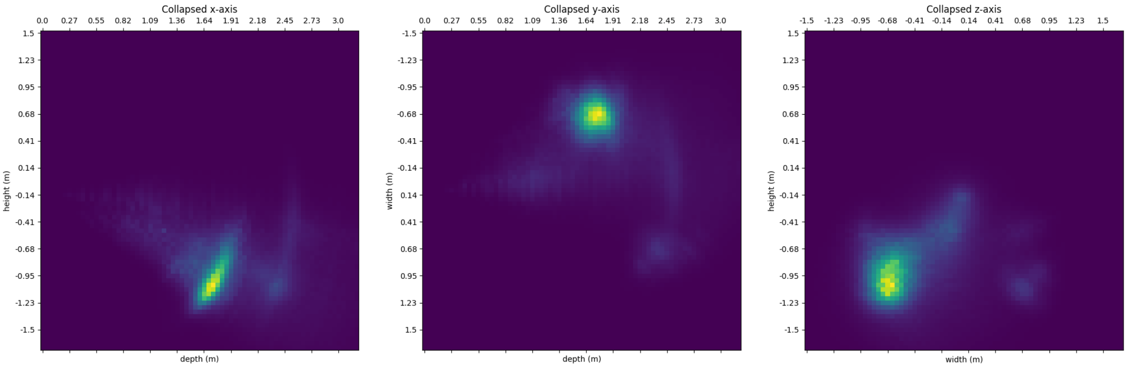


Figure 5.4: SAR image generated depicting two chairs

In all three images we can see a clear shape of the right approximate size, shape, and position. We can however not see any finer detail of the objects. This was as expected since we used a cubic search area with 3m sides, parted into 30 sampling points along each axis, which means that the distance between sampling points is 1dm at the shortest. It then follows that the resulting resolution of the images will be 1dm which isn't sufficient to capture any finer detail of most room-sized objects. The limit on resolution is set by the computation time and resolution of the radar sensor. Since the PCR has a resolution of only a few millimeters, then theoretically we should also be able to create radar images with a resolution of a few millimeters. However, the required computation time for a 3m search area would then be unfeasible for any practical use. Due to lack of time, we have not performed any experiments testing the limits of the search area size or the system resolution, but these would be interesting for future research.

5.7 IMU prototype testing

The IMU-based tracking could still be feasible. But the VR option was prioritized due to the IMU having a more significant drift leading to imprecise readings. Nevertheless, the VR prototype experimentation yielded noteworthy results, revealing that a lesser quantity of data points than anticipated was necessary to generate a satisfactory radar image. This may suggest that more accurate position measurements could be obtained for an extended duration by utilizing an improved IMU, thus enabling the recording of all required radar sweeps. However, the viability of this approach for real-world applications remains doubtful, as the significance of movement and the acquisition of radar sweeps from diverse angles and positions necessitate a more extended recording time, which is improbable to occur naturally. A more realistic approach to the use of an IMU in our system would be to use it in combination with other positioning technologies, either the lighthouse tracking used by the VR hardware or others. The combination of IMU with other sensors for pose tracking is already being studied [15].

5.8 Potential Improvements

5.8.1 Matched Filter

The matched filter method is a commonly used technique in synthetic aperture radar (SAR) for detecting and estimating the parameters of a signal of interest. However, there are other alternatives to the matched filter method in SAR. Constant False Alarm Rate (CFAR) detection, for example, is a statistical method that can be used to detect targets in noisy environments. It uses a sliding window to estimate the local noise level and then compares the signal strength within the window to the estimated noise level to determine if a target is present [16]. Adaptive filtering techniques can be used to improve the quality of SAR images by removing noise and clutter. These techniques involve adjusting filter parameters based on the data being processed, allowing the filter to adapt to changing conditions [17]. Polarimetric SAR (PolSAR) uses multiple polarizations of the transmitted and received signals to extract additional information about the scattering properties of the imaged objects. PolSAR can be used to improve target detection and classification, as well as to estimate properties such as the roughness and orientation of surfaces.[18] Interferometric SAR (InSAR) uses pairs of SAR images taken from slightly different positions to measure changes in the phase of the signal caused by changes in the distance between the radar and the imaged objects. InSAR can be used for a variety of applications, including measuring surface deformation and mapping topography. [19] These are just a few examples of the alternatives to the matched filter method in SAR. With a basis in the developed arbitrary motion SAR, any one of these choices makes a good candidate for an alternative to the matched filter method we used to produce SAR images. The choice of method will depend on the specific application and the characteristics of the data being processed.

5.8.2 ML classification

Our system creates point cloud images of objects in three dimensions. It would potentially be possible to develop machine learning algorithms to implement object classification using this data. This could open up for a completely different avenue of use cases in object recognition for our prototype. The effectiveness of the output of our system as input to an ML classification system will most likely depend on the resolution of the images, and the limits on resolution are yet to be fully explored.

5.8.3 Computational otimizations

The resolution of the 3D images is mostly limited by the computation time which increases exponentially with the number of sampling points. An increase in computational efficiency could therefore be traded for an increase in image quality. There is a plethora of ways one can improve our implementation. The data processing for our prototype is all implemented in Python, and since we wanted to quickly develop a proof of concept, the efficiency of the code itself was not prioritized. Much more efficient data processing could be achieved by working with a faster language such as C, and with speed and efficiency in mind. One could also consider parallelizing the data processing using the matched filter method and running it on a GPU. This would hugely benefit the computation time of the system.

As the algorithm is designed now, the entire matrix of sample points is traversed iteratively, but a majority of the points do not contain any useful information, and we spend computation time calculating correlation scores for empty space. If we could traverse the matrix with a more sophisticated scheme examining only areas of high correlation scores at high resolution, computation times would benefit hugely.

5.9 Alternative technologies

There are other technologies that could be used for the same purpose as our system, such as LiDAR, RGB-D cameras, and stereoscopic cameras. They all have their own challenges and advantages compared to one another and to our system, but they all have in common that they are a lot more expensive than our system has the potential to be. While the Intel RealSense^{TM} RGB-D camera, for example, costs in the range of 500$, the Acconeer A121 PCR sensor costs in the range of 10$. The greatest benefit of our system as opposed to other systems for object detection and imaging is the cost efficiency. Another benefit, however, is complexity. Our system uses no moving parts as opposed to Lidar, and cameras are complicated and the optics can break easily. The Radar sensor can even be mounted internally behind for example a plastic covering, protecting it and hiding it from view.

6 Conclusion

When we commenced work on this project, we set out to answer three questions, namely:

- We set out to explore a novel approach to allow SAR to be used in platforms that move unpredictably, assuming the radar's motion can be precisely tracked, which would unlock many possibilities for SAR-based applications.
- More practically, we wanted to produce a working prototype using some means of pose-tracking to prove this concept.
- We wanted to examine the possibilities of using SAR-based object detection as a viable feature for more general use cases, such as in robotics.

Our findings show that if the radar's motion can be accurately tracked, then the SAR algorithm still works at an adequate and comparable level to the traditional methods.

We succeeded in creating a functional VR prototype implementing our free-form SAR algorithm, capable of producing accurate images of its surroundings.

By evaluating our VR prototype and analyzing its resulting images, we can see that the 3D images created by the system are produced at a speed and with an accuracy that makes three-dimensional object detection possible.

6.1 VR Prototype viability

Based on our analysis, we find that the results obtained from our experiments exhibit significant potential for the VR prototype to be used in a wide range of applications, such as in an obstacle detection system for VR users. The knowledge gained from our previous experiments enabled us to optimize our data-gathering process, resulting in radar images that accurately depict the shape, size, and location of obstacles in the environment. Moreover, our results also demonstrate the feasibility of utilizing a completely arbitrary VAA for SAR, as long as the path can be measured with precision.

6.2 IMU Prototype viability

We were not able to conduct any meaningful experiments with our IMU-based prototype. This came down to issues in precisely determining the position of the radar sensor using the IMU. Pose determination with the use of the IMU was subject to a lot of noise and drift, which made it impossible for us to use.

6.3 Future Work

There is still much exciting research to be conducted in this field.

Firstly, a lot more work can be done attempting to make the system work with the use of an IMU. As discussed in section 3.6, there is a wide range of ways to enhance the quality of pose estimation with the use of IMUs. We did not have the time to incorporate the methods available for our project, but it would be exciting to get it working in the future, enabling a wide range of use cases for the technology. Imagine, for instance, the use of the IMU found in a phone for this purpose. One could then use radar for producing models of any item at any time.

Secondly, there is great potential for research into implementing a real-time demo of the VR prototype. Instead of printing data to a file from the VR game, it could be used to run the algorithm locally, in real-time. The SAR images could then be used to render geometry in the VR game. An obstacle the user might encounter in real life can thus be shown in the game, with the location matching that of the obstacle in real life. This has the potential to truly blur the line between VR and AR.

There is also work to be done in creative ways of post-processing the data generated, and optimizing the algorithm. The quality of images can be enhanced in a lot of different ways we did not have time to delve into. If the algorithms used can be optimized, one could use a lot more data to produce an image or increase its resolution, making for a much more compelling image. One optimization that could be applied is that of improving the traversal of the sample point matrix when applying the matched filter method. In the environment, obstacles are likely sparse. Thus, examining every single grid point is highly inefficient. Focusing on examining the areas of large correlation values in high resolution, while leaving large, empty, areas unexamined would further enhance computation time and provide room to increase the resolution of the image, for example.

The three-dimensional point clouds created by our system could be used as data for machine learning algorithms, making it viable for many potential machine learning projects in the field of object recognition.

There is also potential to use our system as the basis for implementing SLAM (Simultaneous Localization and Mapping). SLAM aims to simultaneously estimate the trajectory of a mobile robot, and create a map of its surrounding environment. It typically makes use of a combination of sensor measurements, such as visual, laser, or sonar data, to estimate the robot's pose and the locations of features in the environment [20]. We think some future work can be done to make SLAM a use case for our system since the data created from our algorithm could potentially be used as sensor data for SLAM algorithms.

One can also ditch the premise of arbitrary VAAs and mount the system to an agent whose motion is controllable with fine precision. For example, a robot arm. Commonly in robotics expensive, stereoscopic RGB-D cameras are used for object localization and determination. Using an Acconeer radar sensor in conjunction with our algorithm in its place could save greatly on cost and development time.

Bibliography

[1] Ingemar Carlsson, Sten-G. Wemmert and Östen Erikmats. “Radar”. In: *Nationalencyklopedin* (2023).

[2] “SAR”. In: *Nationalencyklopedin* (2023).

[3] Acconeer. *Pulse Coherent Radar.* Accessed: 2023-03-13. URL: https://docs.acconeer.com/en/latest/getting_started/pcr.html.

[4] Acconeer. *Frames, sweeps and subsweeps.* Accessed: 2023-03-13. URL: https://docs.acconeer.com/en/latest/handbook/a121/sweeps_and_frames.html.

[5] Acconeer. *Interpreting radar data.* Accessed: 2023-03-13. URL: https://docs.acconeer.com/en/latest/handbook/a121/interpreting_radar_data.html.

[6] M. Younis G. Krieger I. Hajnsek A. Moreira P. Prats-Iraola and K. P. Papathanassiou. “A tutorial on synthetic aperture radar”. In: *IEEE Geoscience and Remote Sensing Magazine* 1.1 (2013), pp. 6–43.

[7] Y. K. Chan and V. C. Koo. “An introduction to Synthetic Aperture Radar (SAR)”. In: *Progress In Electromagnetics Research B* 2 (2008), pp. 27–60.

[8] Stefan Pratschner, Sebastian Caban, Daniel Schützenhöfer, Martin Lerch, Erich Zöchmann and Markus Rupp. “A Fair Comparison of Virtual to Full Antenna Array Measurements”. In: *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC).* 2018, pp. 1–5. DOI: 10.1109/SPAWC.2018.8445923.

[9] Armin Doerry, Edward Bishop and John Miller. *Basics of Backprojection Algorithm for Processing Synthetic Aperture Radar Images.* Feb. 2016.

[10] Acconeer. *Obstacle Detection.* Accessed: 2023-03-15. URL: https://docs.acconeer.com/en/latest/exploration_tool/algo/a111/obstacle.html.

[11] Norhafizan Ahmad, Raja Ariffin Raja Ghazilla, Nazirah M Khairi and Vijayabaskar Kasi. “Reviews on various inertial measurement unit (IMU) sensor applications”. In: *International Journal of Signal Processing Systems* 1.2 (2013), pp. 256–262.

[12] Satoru Emura and Susumu Tachi. “Multisensor Integrated Prediction for Virtual Reality”. In: *Presence: Teleoperators and Virtual Environments* 7.4 (Aug. 1998), pp. 410–422. DOI: 10.1162/105474698565811. URL: https://doi.org/10.1162/105474698565811.

[13] Nikolaos Voudoukis and Sarantos Oikonomidis. “Inverse Square Law for Light and Radiation: A Unifying Educational Approach”. In: *European Journal of Engineering and Technology Research* 2.11 (2017), 23–27. URL: https://ej-eng.org/index.php/ejeng/article/view/517.

[14] Hyonik Lee, Se-Young Kim, Sol Kim and Jong-Won Yu. “A Digital Filter Bank Based Method for Data Reduction in Spaceborne SAR Electronics”. In: *IEEE Access* 10 (2022), pp. 100911–100921. DOI: 10.1109/ACCESS.2022.3200385.

[15] Mary B. Alatise and Gerhard P. Hancke. “Pose Estimation of a Mobile Robot Based on Fusion of IMU Data and Vision Data Using an Extended Kalman Filter”. In: *Sensors* 17.10 (2017). ISSN: 1424-8220. DOI: 10.3390/s17102164. URL: https://www.mdpi.com/1424-8220/17/10/2164.

[16] Hermann Rohling. “Ordered statistic CFAR technique - an overview”. In: *2011 12th International Radar Symposium (IRS)*. 2011, pp. 631–638.

[17] Pratibha Tiwari, Dr. Agya Mishra. “Adaptive Filtering Applications in Radar Signal Processing: A Litrature Review”. In: *International Journal of Engineering and Technical Research* 8.12 (2018), pp. 37–42.

[18] NASA Jet Propulsion Laboratory. *Polarimetry*. https://nisar.jpl.nasa.gov/mission/get-to-know-sar/polarimetry/. Accessed on April 28, 2023.

[19] NASA Jet Propulsion Laboratory. *Interferometry*. https://nisar.jpl.nasa.gov/mission/get-to-know-sar/interferometry/. Accessed on April 28, 2023.

[20] H. Durrant-Whyte and T. Bailey. “Simultaneous localization and mapping: part I”. In: *IEEE Robotics Automation Magazine* 13.2 (2006), pp. 99–110. DOI: 10.1109/MRA.2006.1638022.