Jakob Pirker, BSc

Inverse Sensor Modeling for Automotive Radar Sensors

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Information and Computer Engineering

submitted to

Graz University of Technology

Supervisor

Assoc.Prof. Dipl.-Ing. Dr.techn. Gerald Steinbauer

Institute for Softwaretechnology
Head: Univ.-Prof. Dipl.-Ing. Dr.techn. Franz Wotawa

Graz, April 2021

This document is set in Palatino, compiled with pdfLaTeX2e and Biber.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

Abstract

Occupancy grids are a common method for creating maps of the environment for uncertain sensor data. Measurements are integrated into this map by means of an inverse sensor model. This thesis shows alternative approaches for inverse sensor models for automotive radar sensors.

Commonly used approaches usually rely on the standard output for radar sensors, the object list. This has several shortcomings especially the loss of geometric information. Therefore a geometric representation derived from the raw radar data is used as a basis for the models. It represents the sensor data in a form that can be interpreted more intuitively for integrating the measurements into a gridmap. Two different ways of transforming the sensor's output into a probability of occupancy will be presented. This two transformations are used in three different sensor models.

The models are tested on measurements that have been conducted with a mobile robot platform. This platform was equipped with a radar sensor and odometry for creating gridmaps from the radar data. Additionally a camera and a laser-scanner to create reference occupancy grids were attached to it. The reference gridmaps and the camera images have been used to qualitatively assess the radar-generated gridmaps. 34 different measurement scenarios have been recorded to cover a broad variety of applications and provide data to examine certain sensor and model characteristics.

To be able to assess the final gridmaps criteria are derived that formalize the quality of the generated occupancy grids. These criteria are applied to the generated maps and the results evaluated. Characteristics of the sensor or the modeling approach that led to a reduced quality of the gridmaps are discussed based on these criteria.

Contents

List of Figures

1. Introduction

1.1. Motivation

Applications for automotive radar sensors range from advanced driver assistant systems (ADAS's) [12] to complete autonomous driving [11]. Due to it's ability to also work in bad visibility conditions, it's comparably low price and it's ability to measure the velocity of objects it will also play a key-role in applications where conventional sensors perform poorly. Autonomous driving in bad weather conditions would be such an application. Another one would be navigation in rooms filled with smoke e.g. for firefighting.

An important task in this context is generating a map of the static environment. For this task occupancy grids are a very popular solution. In this thesis solutions for effectively creating occupancy grids from radar measurements should be investigated. However the literature on this field is still quite sparse. There are some projects where radar is used for creating occupancy grids, but in almost all of them the object list is used as a basis for creating the grids. The object list is a standard output of radar sensors, which lists the location and the velocity of the objects the sensor has detected in it's field of view (FOV). This attempt has several shortcomings in the mapping context especially the mapping of free space and the loss of geometric information.

1.1.1. Goals

The main goal of this thesis is to develop an alternative way for creating occupancy grids from radar measurements. Instead of the object list the raw radar data should be utilized for this purpose. The final outcome of this efforts should be an inverse sensor model for a radar sensor that incorporates the sensor's properties as good as possible into the occupancy framework. The found model should also be able to provide a modeling of free space that utilizes the data provided by the sensor more excessively. It should be evaluated by real-world measurements. As a side effect it is desired to gain a deeper understanding of the limitations and possibilities of the radar technology for mapping. Finally shortcomings and future improvements for the developed approach should be presented.

1.1.2. Challenges

The first challenge was to find a mathematical model. This was particularly difficult as the whole radar framework has a lot of tuneable parameters that influence the results. Also there is no derived model in the literature for this purpose. As a complete mathematical derivation proved too difficult in the scope of this thesis a total of six heuristic attempts for an inverse sensor model have been proposed. For all of this models suitable parameters had to be found. The main challenge was to find a suitable mapping from radar-data to probabilities of occupancy that incorporates the sensor's properties.

The second challenge has been conducting the measurements. They needed to cover a variety of different scenarios to get insights into the sensor's properties. At the same time additional data had to be recorded for processing and validation purposes (e.g. reliable positional data and laser scans of the scene). For the final validation it was important to find and apply criteria that accurately describe the quality of an occupancy grid. In this context it was an important point to trace corrupting effects back to the properties of the sensor and/or the model.

1.1.3. Contributions

A novel approach has been developed to extract geometric data from radar measurements. Unlike the object list which relies on the range doppler map, the used models rely on a representation that will later be called range angle map. The six different candidates for the inverse sensor models have been validated on the conducted measurements. Some of the approaches provided decent results. Also an outlook has been given which models could be further investigated to provide better results and utilize the data provided by the radar sensor as good as possible.

1.2. Outline

First there will be a short overview on the related work (Section 2) . Then Section 3 covers important prerequisites of the radar technology and occupancy grids. After that Section 4 explains how a series of measurements is processed into an occupancy grid. Next Section 5 first describes the conducted measurements, in particular the used platform, measurement environments and parameters. After that the results will be evaluated and discussed. Finally in Section 6 the thesis will be concluded and an outlook for further improvements is given.

2. Related Work

2.1. Occupancy Grid Maps

Occupancy grids originate from the field of robotics and were first proposed in [1]. Since then they have been widely used and adapted. The basic idea of this concept is to represent a map by equally big grid-cells with a probability of occupancy. Each sensor measurement is transformed into such a probability by the inverse sensor model and used to update the corresponding grid-cell. [26] gives a good overview how they are used especially in the mapping context of robotics. The mathematical basics from this book that were used for this thesis are described in section 3.2.

A widely used framework in the field of robotics is the Robot Operating System (ROS)[1]. It will be used for the measurements. [14] introduces a Simultaneous Localization and Mapping (SLAM) framework for lidars that uses scan matching for the localization. A package called Hector-Mapping is available for ROS. It will also be used for the measurements. An alternative ROS package for conducting SLAM is Gmapping[2]. It is based on [8]. It's algorithm uses Rao-Blackwellized particle filters.

Applications where occupancy grids have already been used in combination with radar sensors will be described in section 2.3.

[1] ros.org

[2] openslam-org.github.io/gmapping.html

2.2. Radar Signal Processing

Signal processing covers a very big part of this thesis. Basic concepts like the Signal to Noise Ratio (SNR) and the Fast Fourier Transform (FFT) are excessively discussed in [28] and [7]. Background important for radar signal processing like the Radar Cross Section (RCS), Constant False Alarm Rate detectors (CFAR) and the Direction Of Arrival (DOA) can be found in [17] and [15]. Section 3.1 shortly summarizes and explains the relations of this books that are later used to pre-process the raw radar data.

2.3. Occupancy Grid Maps Using Radar Sensors

There have already been several projects where radar is used to create occupancy grids. This section will provide a brief overview of the work already contributed. Further the basic ideas of commonly used approaches will be explained.

2.3.1. Object List Based

The most usual approach is to directly use the common output of radar sensors, the object list. This list contains only positions of the sensor's Field Of View (FOV) on which an object has been detected. Very often it is already provided by the sensor manufacturer itself and therefore most of the signal processing and parameterization is already done. A big shortcoming of this approach is that detected objects are reduced to single points. Geometric information of the object is lost. However to compensate this loss of information usually a probability distribution is laid around the detected points. This way the geometric properties of a detection are modeled. CFAR detectors in combination with DOA estimators (e.g. Multiple Signal Classification) have been used in most cases.

[18] focuses on sensor-fusion of radar and lidar through occupancy grids. For the radar-sensor model a two dimensional Gaussian distribution (in bearing and range) is used to model a radar-beam for each object in the

object list. [16] creates a 3D grid from radar-data. The target detection is based on CFAR and MUSIC. Again a two dimensional Gaussian distribution modeling a radar-beam for each object in the object list is used as inverse sensor model.

[6] investigates two different approaches for radar-based grid mapping: Amplitude-based and an occupancy grid. Distance, bearing and SNR of the observation are taken into account for the map creation. Uncertainties of range and bearing are taken into account by Gaussian distributions.

[5] creates a 2D occupancy grid similar to the approaches stated above, with the small difference that the authors create the object list themselves by using a dedicated FPGA. The SNR values are transformed into probabilities by normalizing them depending on the maximal SNR value of a measurement. The distance to the target, the antenna-gain and the SNR are taken into account for interpreting the radar measurements as probabilities of occupancy. Additionally some post-processing in terms of different clustering algorithms is used for further improving the grid quality.

[10] creates a 3D occupancy grid using a 3D radar sensor. The detection's are weighted by the SNR and interpolated to the nearest grid cell. There is explicitly no measurement uncertainty modeling done.

All those approaches rely on the object list as a basis for the inverse sensor model. Therefore a lot of data is discarded. The inverse sensor model provided in this thesis should not discard this data, but use it to achieve higher quality occupancy grids. However SNR and antenna-gain are important factors that need to be considered for transforming the radar measurements to their corresponding probabilities of occupancy. This factors will also be considered by the models proposed in this thesis.

2.3.2. Raw Data Based

By utilizing the raw radar data the whole sensor's FOV can be used. In [3] the author uses a static, mechanical rotated, narrow beam radar to generate a three dimensional occupancy grid. For each alignment the corresponding beam is processed. Target classification is done by a CFAR algorithm on the range-doppler-map generated from the raw measurement data. The classification is used to separate the measurements in free, occupied (weighted by SNR) and non-informative (e.g. behind objects) space. More theoretical detail can be found in [2]. As the Author concentrates on narrow beam applications, spatial information of detected objects is also modeled using certain probability distributions. The idea of this sensor model has been used, and further refined for the proposed inverse sensor model.

[4] investigates 2D occupancy grid creation using radar in static scenarios especially in bad visibility situations. The used radar scans the area with a narrow beam. No CFAR but a constant, pre-calculated threshold for each measurement-bin is used for obstacle classification. [13] conducts Simultaneous Localization And Mapping (SLAM) with a mechanically scanned radar and uses different approaches for target detection than traditional CFAR techniques. By assuming different noise- and power distributions hypotheses are used for calculating the probabilities of occupancy from the radar measurements. This two papers follow completely different approaches for the inverse sensor modeling. However as CFAR detectors are commonly used, effective and robust, this approach has not been further investigated.

2.3.3. Other Approaches

[11] and [19] perform radar-based occupancy grid creation for parked vehicles, but don't give any implementation details. [21] uses an entirely different approach by trying to create occupancy grids from raw radar data using neural networks. [20] proposes enhancements for radar based occupancy grids by more explicitly mapping free space and by adapting measurement uncertainties to the vehicle speed. [22] focuses on free space detection of occupancy grids on the basis of radar and a stereo camera, but doesn't give specific implementation details. Some of this topics especially the adaptation of the measurement uncertainties could also be investigated for the proposed models in future research.

3. Prerequisites

3.1. Radar

Radar technologies work through transmitting and receiving electro-magnetic waves. The received signals are processed in a way that objects in the sensor's filed of view can be extracted. The basic radar-equation represents the ratio of received power P_r to transmitted power P_t depending on the relevant system parameters according to [12]:

$$\frac{P_r}{P_t} = \frac{G_t G_r \lambda^2}{(4\pi)^3 R^4 L} \sigma \tag{3.1}$$

G_t is the transmitting antenna's gain, G_r receiving antenna's gain, λ the transmitting wavelength, σ the RCS and R the distance to the reflecting object. $L < 1$ denotes losses within the radar sensor.

3.1.1. Radar Cross Section (RCS)

The RCS is a factor describing the ability of an object to reflect energy back towards an antenna. It depends on three major factors ([2]):

- **Size:** the bigger an object, i.e. the bigger it's reflective surface the more energy is directed back to the receiving antenna.
- **Material:** materials with good reflective properties (e.g. metal) reflect more energy than absorbing or pervious materials (e.g. plastic or wood).
- **Directivity:** the reflected energy of a surface is at it's maximum if it is arranged 90° to the wave propagation direction.

The ability of a radar sensor to perceive an object highly depends on the reflected signal's strength. This comes into play especially in real-world scenarios which include a big variety of object types and arrangements.

3.1.2. Discrete Time Fourier Transform

This section will provide a brief overview of the basic signal-processing tools connected with the Fourier Transform (FT) necessary to understand the topics that follow. A more detailed introduction to this subjects can be found in [7] and [28].

In contrast to the Fourier transform which is for continuous-time signals the Discrete Time Fourier Transform (DTFT) is used for discrete-time signals. The time domain signal $x[n]$ is converted into a frequency domain signal $X(e^{j\omega})$. It represents the magnitudes and their corresponding frequencies ω of the sinusoidal decomposition of $x[n]$. This decomposition can be visualized by computing the Fourier Series. The frequency domain representation is also called the frequency spectrum or just spectrum of $x[n]$. $x[n]$ is supposed to be equally spaced in time with $T = 1/f_s$ where f_s is the sampling frequency. The DTFT is defined as:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \tag{3.2}$$

In relation to Equation 3.2 it is important to notice that the frequency spectrum $X(e^{j\omega})$ is continuous. This is because $x[n]$ is observed infinitely long i.e. from $n = -\infty$ to ∞. As this is not possible in real systems the DTFT has to be adapted for finite length signals. This adaption is the Discrete Fourier Transform (DFT). It is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{\dfrac{-j2\pi}{N}kn} \tag{3.3}$$

The DFT assumes that $x[n]$ is periodically outside the base interval from $n = 0 \ldots N-1$. As a result the frequency spectrum is also periodic around $\omega = 0 \ldots 2\pi$. Comparing Equation 3.2 and 3.3 it can be observed that $X[k]$

represents a sampled version of $X(e^{j\omega})$ with N samples. The sample points of the DFT are:

$$\omega_k = \frac{2\pi}{N}k \quad \text{with} \quad \Delta\omega = \frac{2\pi}{N} \tag{3.4}$$

Equation 3.4 shows that by increasing N the difference between the spectral lines for the discrete spectrum becomes smaller. Thus, for increasing N $X[k]$ resembles $X(e^{j\omega})$ more exact. To get a better representation without changing the signal it is therefore possible to simply add a certain number of zeros. This is called zero-padding. Note that this process cannot achieve better frequency resolutions, the DFT simply becomes a denser sampled version of the DTFT. The Fast Fourier Transform (FFT) is a fast implementation of the DFT where N gets zero-padded to a power of two to accomplish faster computation.

According to the Nyquist-Shannon theorem $x[n]$ can be perfectly reconstructed from $X[k]$ by computing the inverse DFT if the highest occurring frequency in $x[n]$ is smaller then half of the sampling frequency f_s. If this theorem is not respected the frequency components higher than $f_s/2$ get projected back into the base spectrum and therefore corrupt it. This is called aliasing.

Taking N samples of $x[n]$ is called windowing. Mathematically the window $w[n]$ is applied by a multiplication in the time domain.

$$v[n] = x[n] \cdot w[n] \tag{3.5}$$

If the time domain samples are used directly for computing the DFT it is more precisely called a rectangular window.

$$w_R[n] = \begin{cases} 1, & \text{if } 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

The problem of this window-shape is that if the window length doesn't perfectly align with a multiple of the period of $x[n]$ there are discontinuities at the window verges due to the periodic assumption of the signal. The DFT interprets this discontinuities as edges in the signal. This edges have an infinitely high number of containing frequency components which cause aliasing. Therefore if unknown frequencies (i.e. no synchronous sampling) should be evaluated, using a different window is necessary. To avoid the mentioned discontinuities the windowing function usually fades to zero at the window's verges. Hanning- and Blackman-window are two common windowing functions for such cases.

The time-domain multiplication of window and signal causes a convolution of their corresponding spectra in the frequency domain.

$$V[k] = X[k] * W[k] \tag{3.7}$$

That means that every frequency component of $x[n]$ present in the spectrum $X[k]$ gets superimposed by the DFT of the windowing function. This shows another problem of the rectangular window: the Fourier Transform of a rectangle is a sinc-function. This function has a very narrow main-lobe at the corresponding frequency bin, but also high side-lobes next to the main-lobe. On the one hand the narrow main-lobe is desired because it points out the location of a single frequency-peak more clearly. On the other hand neighboring frequency peaks are corrupted and might even get invisible in case of small magnitudes by the high side lobes. In contrary the Hanning window has a broader main-lobe but also lower side-lobes. Thus windowing functions always have a trade-off between the side-lobes height and the main-lobes width i.e. the frequency resolution.

3.1.3. LFMCW-Radar

The most commonly used type of radar sensors in automotive applications are LFMCW-radars (Linear Frequency Modulated Continuous Wave). A continuous sinusoidal electromagnetic wave is transmitted. As illustrated in figure 3.1 the frequency of this wave is linear chirped over a certain bandwidth. Using this method the propagation delay τ of the sent signal can be evaluated by estimating the frequency difference Δf. Knowing τ allows to estimate the relative velocity and distance to an object.

Figure 3.2 shows the basic schematic of a FMCW-radar. The oscillator generates the frequency chirp. It is transmitted via the antenna. For systems where transmitting and receiving is done via the same antenna a separation block is necessary. Finally the echo is down-mixed by the sending signals

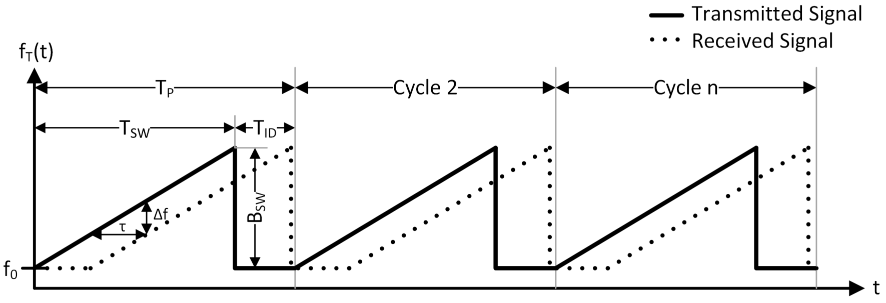


Figure 3.1.: Frequency chirp of a LFMCW radar. B_{SW} is the chrip bandwidth, f_0 the base frequency. T_{SW} is the sweep duration, T_{ID} the idle duration, necessary for settling in between two cycles. $T_P = T_{SW} + T_{ID}$ is the overall duration for one cycle. The application illustrated uses just an up-chirp (saw-tooth function), but using a down-chirp for reducing the frequency back to f_0 (triangular function) is also possible.

frequency. An important component is the mixer. It converts the received signal down by the senders frequency. Therefore the mixer already forms the differential signal needed for estimating the intermediate frequency $\Delta f = f_{IF}$. As a result the signal processing can be done in a much lower frequency-range. This relaxes the necessary hardware's requirements remarkably especially in terms of sampling frequency. The usually used Band is 76GHz - 81GHz. The simplified IF-signal model for one reflection (i.e. the signal that is received at the output of the mixer for one object in the

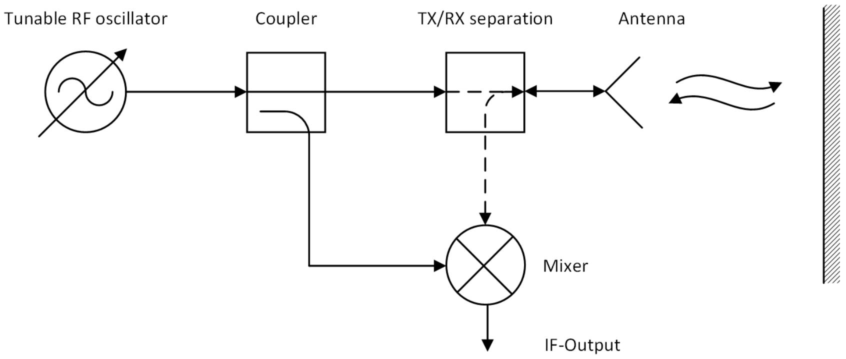
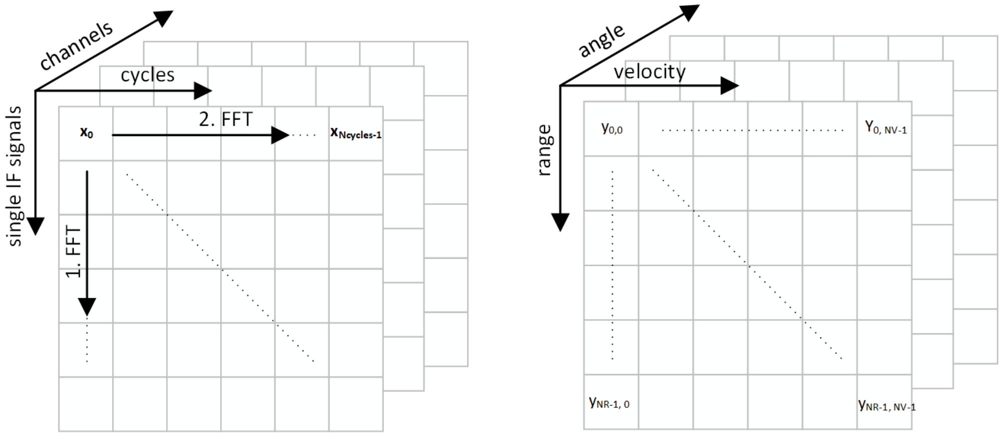


Figure 3.2.: Basic FMCW-radar schematic



(a) Matrix for the measured data.

(b) Range-doppler matrix arrangement.

Figure 3.3.: Matrix representations of measured and processed data

sensor's FOV) is given with (derivation can be found at [25]):

$$s_{IF}(t) = A_{IF} cos \left(2\pi \left(\underbrace{\frac{B_{SW}}{T_{SW}} \tau}_{f_{IF}} + f_0 \frac{2v}{c} \right) t + \underbrace{2\pi f_0 \tau}_{\Phi_0} \right) \tag{3.8}$$

3.1.4. Range Doppler Processing

Figure 3.1 illustrates that several consecutive cycles are sent and received for one measurement. Each cycle itself acquires N samples as a data vector $\boldsymbol{x}_i$ with $i = 1...N_{cycle}$. The collected data is arranged in a matrix. It's structure is depicted in Figure 3.3a. Every column of this matrix is one data vector for one chirp sequence. Therefore the map has N rows and N_{cycle} columns. First an appropriate windowing function is applied to the measured data-vector $\boldsymbol{x}_i$. Then the complex valued spectrum $\boldsymbol{y}_i$ for each data vector is calculated using a 1. FFT. The vector y_i might be longer than x_i as a result of zero-padding done in the FFT. The vectors y_i are again arranged as columns of a matrix, called the range-compressed map. Thus each column of this new matrix now contains the spectrum of one cycle. Note that by using

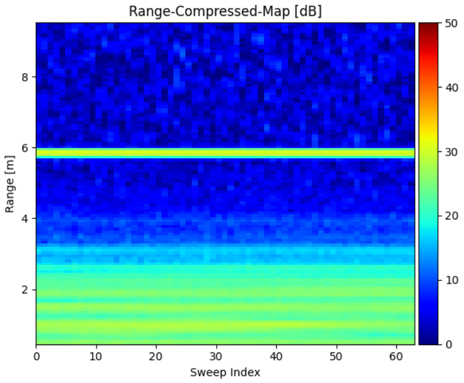


Figure 3.4.: Illustration of a range-compressed map with a target at approximately 6m.

the FFT the spectrum is symmetric on the x-axis i.e. it is the same for positive and negative ranges. As a consequence just the positive part of the range is shown. However for the target detection it is important to use the whole spectrum. Figure 3.4 shows a visualization of a range-compressed matrix. f_{IF} can be extracted by evaluating the peak of the spectrum. If there are multiple targets at different ranges in the measurement range of the radar, there are peaks at multiple frequencies visible in the spectrum. From equation 3.8 the relation

$$\tau = f_{IF} \frac{T_{SW}}{B_{SW}} \tag{3.9}$$

can be used to determine the propagation delay from f_{IF}. Knowing τ the distance to the object can be evaluated by using c the propagation speed of light. The factor 2 is necessary because the wave has to travel to the object and back.

$$r = f_{IF} \frac{c}{2} \frac{T_{SW}}{B_{SW}} \tag{3.10}$$

Due to the sampling the maximal resolvable frequency is $f_s/2$. Therefore the maximum resolvable distance is

$$r_{MAX} = f_s \frac{c}{4} \frac{T_{SW}}{B_{SW}} \tag{3.11}$$

The range resolution is calculated according to [24]

$$\Delta r = \frac{c}{2B_{SW}} \tag{3.12}$$

If the object is moving, τ changes at each cycle. To point this out mathematically, every row ξ of the range-compressed map can be interpreted as

$$Y[\xi, m] = \left[A_{\xi,1}e^{j\Phi_{0,\xi,1}}, A_{\xi,2}e^{j\Phi_{0,\xi,2}}, ..., A_{\xi,N_{cycle}}e^{j\Phi_{0,\xi,N_{cycle}}}\right] \tag{3.13}$$

This collects the phases and amplitudes of the same bin of the frequency spectra into a single vector. The phase changes proportional to $\Delta\tau$ according to

$$\Delta\Phi_0 = 2\pi f_0 \Delta\tau = 2\pi f_0 \frac{2v}{c} T_P \tag{3.14}$$

This phase change can be re-interpreted as a frequency. Assuming the amplitudes for each bin do not change, it can be interpreted as a complex exponential vector of the form

$$z[m] = A_\xi e^{j\Delta\Phi_0 m} = A_\xi e^{j2\pi f_0 \frac{2v}{c} T_P m} \tag{3.15}$$

m denotes the sweep index. Equation 3.14 shows that the phase change $\Delta\Phi_0$ is proportional to the velocity v. Therefore according to Equation 3.15 $\Delta\Phi_0$ can be extracted as a frequency which is related to the object's velocity.

Evaluating the velocity is similar to the range evaluation. By applying a 2. FFT on the rows of the range-compressed map the spectrum of the velocity dimension is calculated. Again an appropriate windowing function has to be used. The computed spectra are used as the rows of a new matrix - the final range doppler map. The column- (or also called slow-time-) dimension of this matrix represents the range. The row- (also called fast-time-) dimension represents the velocity. Figure 3.3b illustrates the structure of the matrix. Figure 3.5 shows a visualization of this matrix. The final range doppler map has a dimension of $N_{FFT1} \times N_{FFT2}$. N_{FFT1} is the length of the spectral vector of the range-compression. N_{FFT2} is the length of the spectral vector of the velocity-compression. The peaks in this the RD-map correspond to objects. By extracting the peaks of the spectrum in the fast-time dimension, the velocity of the objects can be identified. In equation 3.14 the factor $\frac{\Delta\Phi_0}{2\pi}$ can be transformed into the expression $\frac{f_{IF}}{f_s}$. Thus knowing the peak frequency

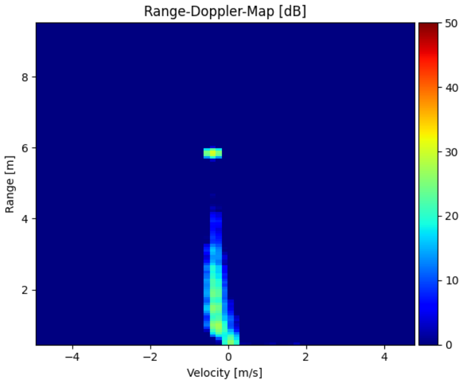


Figure 3.5.: Illustration of a range doppler map with a target at a distance of approximately 6m and a velocity of approximately 0.5 m/s.

f_v in the range doppler map the velocity can be calculated by

$$v = \frac{f_v}{f_s} \frac{c}{2 f_0 T_P} \tag{3.16}$$

The velocities range and resolution can be calculated using [24]

$$v_{MAX} = \frac{c}{4 f_0 T_P} \tag{3.17}$$

$$\Delta v = 2 \frac{v_{MAX}}{N_{cycle}} \tag{3.18}$$

The illustration in Figure 3.5 shows a range doppler map. Common radar sensors used in automotive applications have more than one receiving antennas (i.e. channels). Using the steps provided, the signal received on every one of them can be processed to a range doppler map. The map shown in Figure 3.5 has been created using non-coherent sampling. That means it is obtained by averaging the corresponding bins of the range doppler maps for every channel.

3.1.5. Target Detection

Separating objects from noise in a spectrum is called target detection. One of the biggest problems of radar sensors is that they suffer from high noise influence. The two sources of noise in a radar system are internal noise (e.g. phase noise) and external noise (e.g. clutter). Modern radar systems use Constant False Alarm Rate (CFAR) detectors to establish a reliable detection in noisy conditions. The false alarm rate P_{FA} describes the probability on how probable noise is mistakenly considered as detection. The two important components for calculating P_{FA} are the noise model and the RCS fluctuation model. The fluctuation model represents the change of the RCS over time. Usually Swerling models are used for this purpose. [17, p. 496] shows that for white Gaussian noise and a nonfluctuating target P_{FA} is

$$P_{FA} = e^{-T/\sigma_w^2} \quad \text{with} \quad T = \alpha\sigma_w^2 \tag{3.19}$$

σ_w^2 denotes the noise power. T is the detection threshold, α the scaling between threshold and noise power. This already shows that for setting a detection threshold the noise power needs to be known. As σ_w^2 is highly dependent on the environment (e.g. temperature, geometric measurement scenario...) setting a constant threshold is not feasible for general applications. Therefore it is necessary to estimate the noise power in real-time to provide a possibly constant false alarm rate. The estimate is based on the current measurement, more particular it's range doppler map. Estimating σ_w^2 of a single range doppler cell is based on two major assumptions:

- Neighboring cells that are used for estimating σ_w^2 only contain noise, no targets.
- Neighboring cells have the same noise statistics as the Cell Under Test (CUT).

The Neyman-Pearson detector optimally describes the stated threshold-detection problem. It is approximated by a square-law detector for simplicity. With independent identically distributed white Gaussian noise the detection Probability Density Function (PDF) for a single cell with the measured signal x_i is according to [17, p. 500]

$$p(x_i) = \frac{1}{\sigma_w^2} e^{-x_i/\sigma_w^2} \tag{3.20}$$

If N cells adjacent to the CUT are used for the estimate it is shown that σ_w^2 can be calculated using

$$\widehat{\sigma_w^2} = \frac{1}{N}\sum_{N} x_i \tag{3.21}$$

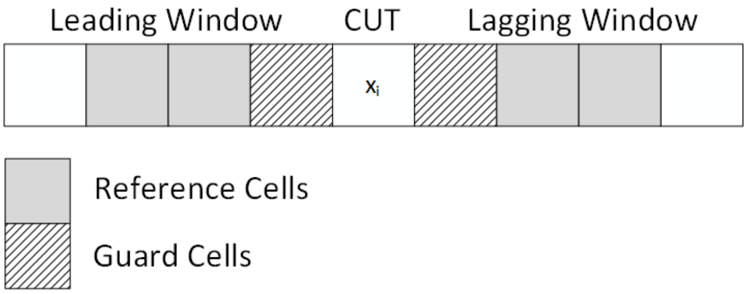


Figure 3.6.: Cell averaging CFAR

This is also referred as the cell-averaging CFAR. Figure 3.6 shows the structure for this estimation. The actual averaging is done with the values in the leading- and the lagging-window. In most cases a target will occupy several consecutive cells. To avoid using this occupied cells next to the CUT for the estimate, guard cells are placed between the windows and the CUT which are not used for the averaging. Figure 3.7 shows a cell-averaging CFAR applied to a range-doppler map. For a given average false alarm rate $\overline{p}_{FA}$ the necessary threshold multiplier α can be chosen according to [17, p.504] as

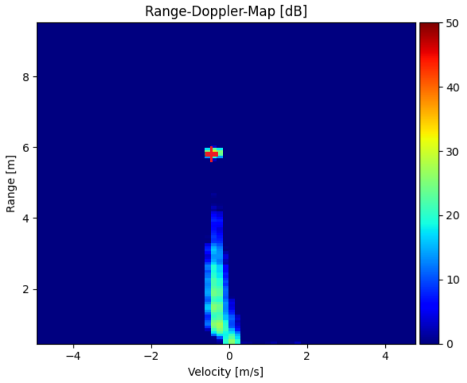


Figure 3.7.: CFAR Detection on the range doppler map. Crosses indicate a detection at a certain cell.

$$\alpha = N(\overline{p}_{FA}^{-1/N} - 1) \tag{3.22}$$

The detection probability p_D is evaluated according to [17, p.504] as

$$p_D = \left(1 + \frac{\alpha}{N(1+\xi)}\right)^{-N} \tag{3.23}$$

where ξ denotes the signal to noise ratio.

3.1.6. Direction of Arrival (DOA)

Modern radar sensors are additionally equipped with antenna array's for sending and receiving. By fusing the spatial- and the temporal information provided, it is possible to determine the Direction of Arrival (DOA) of a signal. This makes it possible to estimate the relative angle to a target. The output signal of a single receiving antenna $\boldsymbol{x}(t)$ of the array is modeled according to [15], [9] as

$$\boldsymbol{x}(t) = \boldsymbol{a}(\theta)s(t) \tag{3.24}$$

$s(t)$ describes the source- or sender- signal. $\boldsymbol{a}$ denotes the steering vector. It represents the electro-magnetic wave's amplitude in the direction given by the angle θ. Extending this model for a whole antenna array is done by collecting all steering vectors in a matrix $\boldsymbol{A}(\theta)$ and all sources in the vector $\boldsymbol{s}(t)$. Incorporating additive noise yields the commonly used model for array processing

$$\boldsymbol{x}(t) = \boldsymbol{A}(\theta)\boldsymbol{s}(t) + \boldsymbol{n}(t) \tag{3.25}$$

For further investigation it is necessary to create the spatial covariance matrix $\boldsymbol{R}$

$$\boldsymbol{R} = E\{\boldsymbol{x}(t)\boldsymbol{x}(t)^H\} \tag{3.26}$$

For real-world applications $\boldsymbol{R}$ has to be estimated from a limited number of N samples. The sample covariance matrix is such an estimate

$$\hat{\boldsymbol{R}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}(i)\boldsymbol{x}(i)^H \tag{3.27}$$

This models form the basis for further DOA-estimators, or also called beamformers. In the following paragraphs two types will be discussed more intensively: the standard or Bartlett- and the Multiple Signal Classification (MUSIC)- beamformer.

Bartlett Beamformer

This is the most basic beamformer. It poses an extension of Fourier-based spectral analysis for sensor arrays [23]. The array output of the signal coming from direction θ corrupted by additive noise is modeled according to 3.24. The idea of this estimator is to maximize the output power with respect to θ. [9] derives the resulting spatial power spectrum

$$P_{BF}(\theta) = \frac{\boldsymbol{a}^H(\theta)\hat{\boldsymbol{R}}\boldsymbol{a}(\theta)}{\boldsymbol{a}^H(\theta)\boldsymbol{a}(\theta)} \tag{3.28}$$

The denominator can be expressed as $|\boldsymbol{a}_{ULA}(\theta)|^2 = M$, with M the number of source signals in $s(t)$. The denominator is just a constant scaling for the spectrum. For a uniformly spaced linear array the steering vector is

$$\boldsymbol{a}_{ULA}(\theta) = \begin{bmatrix} 1 & e^{j\phi} & \dots & e^{j(L-1)\phi} \end{bmatrix}^T \tag{3.29}$$

ϕ is called the electrical angle, and can be translated in a geometrical angle according to [9]

$$\phi = -\frac{\omega}{c} d cos(\theta) \tag{3.30}$$

L denotes the number of antennas of the array. d is the distance between two antenna elements. Putting together Equation 3.28 and 3.27 shows that the numerator of 3.28 contains the term $\boldsymbol{a}^H(\theta)\boldsymbol{x}(t)\boldsymbol{x}(t)^H(t)\boldsymbol{a}(\theta)$. Using Equation 3.29 and comparing it with the definition of the DFT in equation 3.3 shows that this result is the spatial equivalent to the periodogram $P_{BF}(\theta) \sim |X[k]|^2$. This spectrum is formed by conducting a FFT on the samples of the angular dimension of the range doppler matrix (Figure 3.3b). Detecting targets is again done the same way as before: by extracting peaks in the spectrum. The resolution of this beamformer is limited by the resolution of the antenna array's number of antennas and their geometry. This approach also doesn't incorporate the signal quality for the estimate.

Multiple Signal Classification

A more sophisticated method is an approach called Multiple Signal Classification (MUSIC). It is a subspace-based method. The spatial covariance matrix is divided into a source- and a noise-covariance matrix

$$\boldsymbol{R} = E\{\boldsymbol{x}(t)\boldsymbol{x}(t)^H\} = \boldsymbol{A}E\{\boldsymbol{s}(t)\boldsymbol{s}(t)^H\}\boldsymbol{A}^H + E\{\boldsymbol{n}(t)\boldsymbol{n}(t)^H\} = \boldsymbol{APA}^H + \sigma^2\boldsymbol{I} \tag{3.31}$$

$\boldsymbol{P}$ denotes the source- and $\sigma^2\boldsymbol{I}$ the noise- covariance matrix with the noise power σ^2. Using a unitary transform a representation featuring the eigenvalues can be given

$$\boldsymbol{R} = \boldsymbol{U}_s\boldsymbol{\Lambda}_s\boldsymbol{U}_s^H + \boldsymbol{U}_n\boldsymbol{\Lambda}_n\boldsymbol{U}_n^H \tag{3.32}$$

$\boldsymbol{\Lambda}_s$ and $\boldsymbol{\Lambda}_n$ are diagonal matrices containing the real eigenvalues of source and noise respectively. By estimating the number of targets, MUSIC can achieve an angular resolution that goes beyond the physical resolution of the antenna array.

3.2. Occupancy - Grids

The two basic types of map representations are feature-based and location-based maps. As the name implies the elements of feature-based maps are distinct features (e.g. obstacles) of the environment along with their Cartesian coordinates. In location-based maps on the other hand each map element corresponds to a specific location. This kind of map not only contains information about the location of objects but also about free space. This property is very important for not only for autonomous driving but many things e.g. navigation, path planning and obstacle avoidance.

Occupancy grids are a special type of location based maps used for mapping static environment. They were first introduced by Elfes [1]. In a lot of frameworks they are also used as a basis to further solve the so called SLAM-problem (Simultaneous Localization And Mapping). The occupancy grid framework aims to provide means for generating a map of the environment from noisy sensor data while simultaneously dealing with unknown or uncertain position data. The general posterior probability over a map is denoted with

$$p(m|z_{1:t}, x_{1:t}). \tag{3.33}$$

This expression describes the probability for a map m given all so far conducted measurements $z_{1:t}$ and vehicle positions $x_{1:t}$. In this case the objective would be to find a map which maximizes the posterior. The map is divided in equally sized grid-cells.

$$m = \bigcup_i \boldsymbol{m}_i \tag{3.34}$$

Each of those cells is a continuous random variable that carries information about it's occupancy status. $p(m_i = 1)$ indicates that a cell is certainly occupied and $p(m_i = 0)$ certainly free respectively. The first big simplification of this model is, that the occupancy of each cell is assumed to be independent from it's neighboring cells occupancy. The posterior is approximated as the product of it's marginals according to [26, p. 225] as

$$p(m|z_{1:t}, x_{1:t}) = \prod_i p(\boldsymbol{m}_i|z_{1:t}, x_{1:t}) \tag{3.35}$$

This drastically decreases the necessary computational effort for computing the posterior. The second big simplification is to apply the Markov Assumption to the occupancy of each cell. This means, that the stat of each cell (e.g. the occupancy) is sufficient to describe all past states and measurements. The advantage of this is, that in order to update a cell only the current state and measurement are relevant. The expression

$$p(\boldsymbol{m}_i|z_t, x_t) \tag{3.36}$$

is called inverse sensor model. Inverse describes the property, that the model concludes from the impact (measurement) to the cause (occupancy = environment). It formulates how a measurement z_t and a position x_t influence the probability of occupancy for a cell. The contrary approach - creating maps using a forward model - is discussed in [27], but will not be discussed further in this thesis. To avoid certain numerical difficulties the probability is usually implemented as log-odds-ratio ([26]).

$$l_{t,i} = log\left(\frac{p(\boldsymbol{m}_i|z_{1:t}, x_{1:t})}{1 - p(\boldsymbol{m}_i|z_{1:t}, x_{1:t})}\right) \tag{3.37}$$

The probability can be recovered from the log-odds-ratio using

$$p(\boldsymbol{m}_i|z_{1:t}, x_{1:t}) = 1 - \frac{1}{1 + exp(l_{t,i})} \tag{3.38}$$

The update rule according to the Bayes filter for the log-odds ratio of a cell $\boldsymbol{m_i}$ (i.e. the measurement applies to the cell):

$$l_{t,i} = l_{t-1,i} + log(p(\boldsymbol{m}_i|z_t, x_t)) - l_0 \tag{3.39}$$

l_0 denotes the prior of occupancy in it's log-odds form.

4. Processing Pipeline

This chapter shows which data is necessary and how it is processed to generate occupancy grids from radar data as well as reference occupancy grids. The data-flow of the processing pipeline is divided into three major steps as illustrated in Figure 4.1. The first step is the data acquisition. It organizes how the measurement data is acquired and stored. Second is the radar signal processing. In this step the previously measured raw radar data is processed in a way that the interesting features for creating a gridmap are extracted. In the third step the previously modified data is incorporated into an occupancy grid.

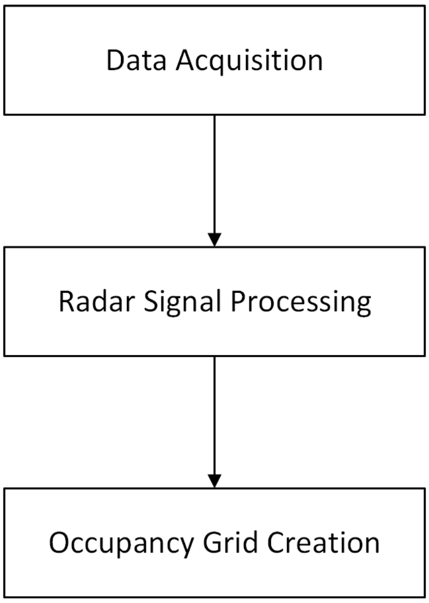


Figure 4.1.: Processing pipeline overview.

4.1. Data Acquisition

This block in the pipeline structures how the data captured by the sensors on the robot platform is acquired and stored. Figure 4.2 gives a more detailed overview. It shows the two big components for the acquisition process: a robot platform equipped with various sensors and a data converter. They will be explained in more detail in the sections to follow. The final result of this block is a file that contains the necessary data for all further steps. The file is stored in an universal data-format that can be read platform independent.

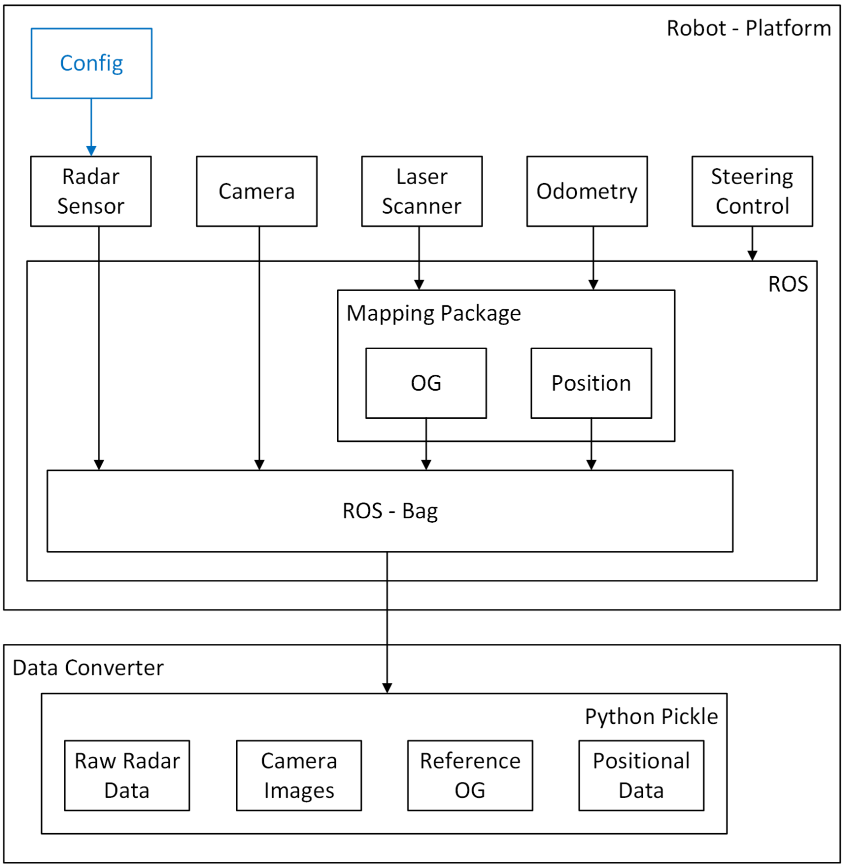


Figure 4.2.: Data acquisition overview.

4.1.1. Robot - Platform

For the evaluation it is necessary to record data from measurement scenarios with dynamic positions. Therefore the robot platform needs to be mobile but not autonomous. Steering will be done by an external operator.

For the creation of the occupancy grids the position for each radar measurement is necessary. Furthermore scans from a lidar and their corresponding positions are needed for creating the reference gridmaps. Lidar is used because it is capable of creating high quality grids. Additionally there are a lot of ROS software packages available that already provide means for creating occupancy grids from those sensors. Finally a camera is required that films from the robot's ego perspective. It documents the path and provides additional means for a qualitative evaluation.

The sensors are attached to the robot and linked to the operating system. It is mainly used for collecting and recording measurement data. The creation of the radar generated occupancy grids will be done offline. In the following sections the components that need special attention will be described in more detail.

Radar Sensor Configuration

Section 3.1.4 shows that there are several parameters influencing the performance of the sensor. Some of these parameters can be defined by the user depending on the use-case. Using the data for creating occupancy grids special attention will be given to the parameters r_{MAX} and Δr as they directly influence the grid quality. For tuning them Equations 3.11 and 3.12 show that the choice of T_{SW} and B_{SW} play an important role. The graphical interpretation in Figure 3.1 is advised to be revisited at this point. Those parameters have a special significance since they can be directly influenced by the sensor configuration.

A big chirp- bandwidth B_{SW} is beneficial for the distance resolution Δr. The bigger B_{SW} the lower the range resolution. This parameter is limited by the sensors capabilities and by the frequency band radar sensors are allowed to operate in. However the bigger B_{SW} the lower the maximum range r_{MAX}. An intuitive explanation for this is that more samples are necessary to capture the whole frequency range. This can be compensated for by using a higher sampling frequency f_s which is limited by the sensors internal hardware.

Increasing T_{SW} results in a higher maximum resolvable range but also increases the duration T_p of a single sweep-cycle. The frame rate of the sensor defines the maximum time a whole measurement is supposed to take. Therefore the frame rate defines the duration of $N_{cycles} \cdot T_p$. An increase of T_p results in a decrease of N_{cycles}. Eventually an increased T_{SW} decreases v_{MAX} (Equation 3.17) and also Δv (Equation 3.18). Note that this parameters have to be configured before the measurements

Robot Operating System (ROS)

The robot platform is running ROS. It already comes with a lot of useful features as well as packages for interpreting sensor and control data. The operating system is used for three major tasks. The first task is simply translating the steering of the operator into commands for the motor-drivers of the platform.

The second task is the creation of the reference occupancy grid. This task won't be done manually since there already exist powerful packages to create grids using laser-scanners. Using a package which comes with a fully featured SLAM algorithm is also important to provide corrected positional data as well (e.g. odometry drift).

As the processing in this thesis will be done offline, the third task is the recording of the necessary sensor data. ROS already provides a proprietary recording format for that purpose called ROS-bags. For each different measurement scenario a bag file is created.

4.1.2. Data Converter

Unfortunately ROS-bags can't easily be read by an operating system that has no ROS distribution installed. However one sub-goal of the thesis was to provide a set of measurement-scenarios that can be easily reused for different evaluations on different operating systems. To be able to also use the data on Windows-machines the data has to be converted into a different format. Therefore the ROS-bags are translated into python-pickles.

ROS simply dumps the message-objects published on the ROS-topics that are marked for recording into the bag-file. In the first step the converter creates a configuration file, that shows the data-structure for each topic. The structure is derived by reading the first message-object of a topic. For every variable present in this object an entry is created in the configuration file. Each of those entries has additional parameters that can be used for influencing the output (e.g. the length of array-like variables, the data-type etc.). By simply deleting an entry from the configuration the corresponding variable is also ignored for the later conversion. In the second step the information from each topic-object is extracted into a two dimensional data format based on the configuration file.

4.2. Radar Signal Processing

Figure 4.3 shows the steps for the radar signal processing. The input for this block is the raw radar data from the converted output file. The raw data is basically the IF-signal (Equation 3.8) recorded after the mixer in Figure 3.2. This signal is sampled for every antenna and represented as an one dimensional array at the output. This array will be transformed into a 3D-matrix such that it has the format shown in Figure 3.3a for further processing.

4.2.1. Range-Doppler Processing

The next step is the range-doppler processing. As described in Section 3.1.4 a windowed FFT is applied to the range and velocity dimensions. The windowing function used is a Hanning window. For each antenna the matrix now looks as depicted in Figure 3.3b. An exemplary measurement can be seen in Figure 3.5.

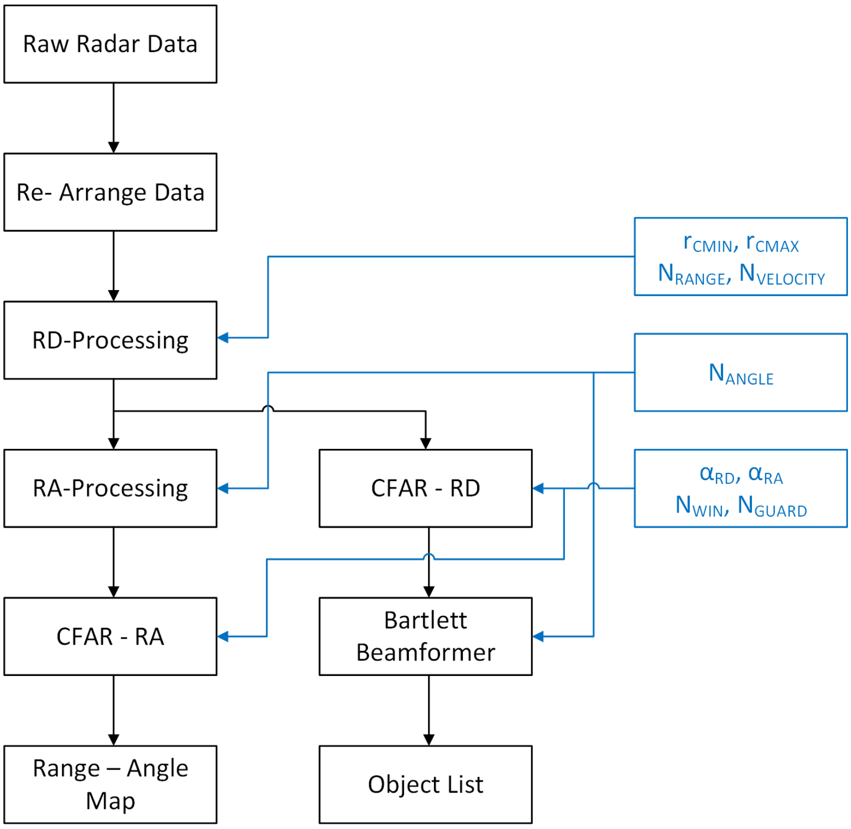


Figure 4.3.: Radar signal processing chain overview.

Important in this step are the parameters r_{CMIN}, r_{CMAX}, N_{RANGE} and $N_{VELOCITY}$. Subsequently they represent also the length of the windowing functions. r_{CMIN} and r_{CMAX} crop the measured matrix such that the bins outside the defined range are removed. Using Equation 3.11 the indices corresponding to the distances can be calculated according to

$$i_{MIN} = \left\lceil \frac{r_{CMIN}}{r_{MAX}} N_{RANGE} \right\rceil \tag{4.1}$$

i_{MAX} is achieved following the same equation. N_{RANGE} and $N_{VELOCITY}$ are the number of samples for the range and velocity FFT's. Going back to Equation 3.3 N_{RANGE} and $N_{VELOCITY}$ can be chosen arbitrary high due to zero-padding as described in Section 3.1.2. In a later processing step, the Signal to Noise Ratio (SNR) of each range bin will be transformed into a probability for occupancy. As the range r_{MAX} and the velocity v_{MAX} are divided into N_{RANGE} or $N_{VELOCITY}$ bins respectively, those parameters control how many SNR values the range doppler map has. Choosing values for N_{RANGE} and $N_{VELOCITY}$ that are too high would mean to incorporate a number of SNR values that is not supported by the physical capabilities of the sensor. Redundant information would be incorporated into the gridmap. Therefore the number of bins has to reflect sensor's capabilities. N_{RANGE} is chosen according to

$$N_{RANGE} = \frac{r_{MAX}}{\Delta r} \tag{4.2}$$

$N_{VELOCITY}$ is calculated the same way with respect to v_{MAX} and Δv (see Equations 3.17 and 3.18).

4.2.2. Range Angle Processing

Conventional occupancy grid creation with radar is usually done by detecting targets in the range doppler domain. This detections are points where the spatial information is later added using three dimensional Gaussian probability models as shown for example in [16, p. 47]. This approach has several difficulties:

- **Loss of geometric information:** In the range doppler domain objects are reduced to points. On each of this points a probabilistic distribution is applied that is supposed to model a radar beam. The real shape of an object could just be reconstructed by the superposition of the probability distributions of multiple point-targets belonging to the same object. This however requires an ideal measurement and ideal parametrization of the distributions.

- **Free space modeling:** Free Space is just modeled in front of a detection. That means space without a detection is considered unknown. Potentially important information is neglected. An example where this would prove difficult is a vehicle on an empty street with a radar sensor aligned in driving direction. Supposing there are no obstacles on the street, it would always be marked as non-informative.
- **Model parametrization:** Finding suitable models and parameters for the point- probability distributions is a very difficult task, especially the superposition case as described above has to be considered.

An alternative to analyzing the range doppler domain is analyzing the range angle domain. By conducting a third FFT (which is a Bartlett beamformer according to Equation 3.1.6) in the angular dimension (on the channels) of the data matrix (see Figure 3.3) a spatial representation is achieved. Further it will be referenced as range angle map. Figure 4.4 shows how a map created with this method can look like. The asymmetry of the angle in this representation is a result of the even number of samples used for the FFT. The negative part of the axis has a sample more since the positive part includes the sample belonging to 0°. This could be changed by using an odd number of samples for the Fourier Transform. However then FFT algorithms could not be used as they require the number of samples to be a power of two. This would lead to a significantly increased computation time since other Fourier Transform algorithms needed to be used. Note that for the final representation the maps of all channels are non-coherently combined (as for the range doppler map, see Section 3.1.4).

In this representation each bin of the matrix represents the SNR of the received power for a certain distance and angle. This can already be interpreted very intuitively: a high SNR means that there is probably an obstacle, a low SNR means there is probably just noise present in the corresponding bin. To incorporate the information into an occupancy grid a more formal definition is necessary.

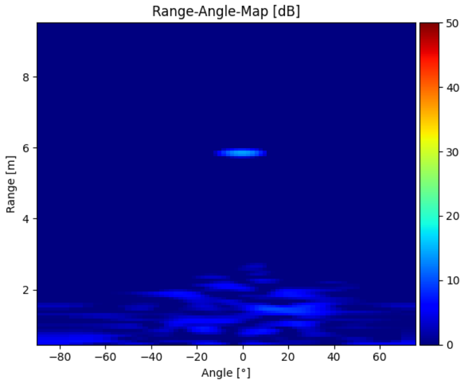


Figure 4.4.: Map of the range angle domain. A target can be seen at around 6m in front of the sensor.

4.2.3. Target Detection

The theoretical background for this block has already been described in Section 3.1.5. A Cell Averaging CFAR-algorithm is used to distinguish noise from signal information. As pointed out in Figure 4.3 it is done in two different ways: based on the range doppler map and the range angle map. In the first path the CFAR is applied on the range doppler map. The algorithm follows the classical detection scheme for radar sensors. The range doppler map is non-coherently combined over all channels, then a CA-CFAR is applied along the range dimension. The bearing of the extracted target points is determined by applying a Bartlett beamformer on the channels. As illustrated in Figure 4.5 this is done for the points on which a target has been detected. The detections with their corresponding ranges, velocities and bearings are assembled into an object list.

In the second path the CFAR algorithm is used on the non-coherently (now over all velocities) combined range angle map. The CFAR window is applied along the range dimension. This is particularly important since the peak width in this direction is approximately constant where as in the angle dimension it is not. A peak in the range angle map represents a reflective surface. In the angular dimension such a surface can be arbitrary wide. Figure 4.6 shows an exemplary measurement. For the calculation of the

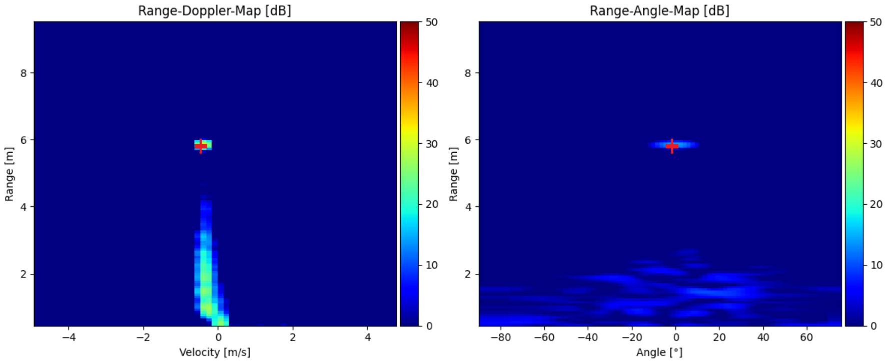


Figure 4.5.: CFAR detection on the range doppler map. Crosses indicate detections.

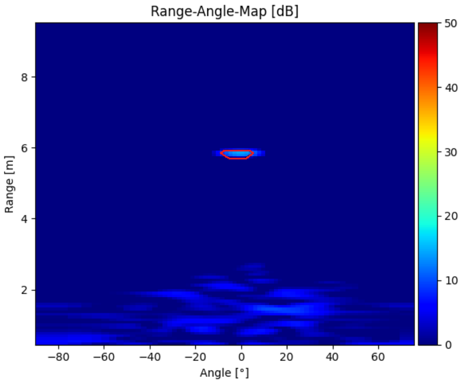


Figure 4.6.: Range angle Representation with applied CFAR - detection (inside margin).

SNR the definition

$$SNR = \frac{P_M}{P_N} = \frac{P_S + P_N}{P_N} \tag{4.3}$$

is used. P_M denotes the measured power and P_N the noise power. For P_M the measured power given in the range angle map is used, because the actual signal power can't be directly observed. It is the sum of noise- (P_N) and signal power (P_S).

There are three important parameter types for this processing step:

1. The threshold value α.
2. The averaging window length N_{WIN}.

3. The guard window length N_{GUARD}.

Equations 3.22 and 3.23 show the influence and connection between these parameters. The most straight-forward determinable parameter is N_{GUARD}. It should be set such that the total number of guard cells is equal to the width of a detection-peak. Equation 3.19 shows that a higher N_{WIN} diminishes the false alarm rate since it improves the noise estimate. On the other hand it can't be chosen too big since possible detections at the same angle may corrupt the noise estimate.

The most difficult parameter to set is the threshold. It is set differently for the range doppler- and the range angle approach; α_{RD} and α_{RA} respectively. As already described in Section 3.1.5 a signal is classified as target if $SNR > NOISE \cdot \alpha$. For radar the reflected energy highly depends on the RCS. Two of it's factors are highly scenario-dependent (e.g. material and object geometry). Two extreme complementary examples would be a car and a tree. The car on the one hand provides big flat metal surfaces which are easy to detect. The tree on the other hand does not have large reflective surfaces and additionally wood is a very poor reflector of the radar waves. A parking lot for example provides a completely different detection scenario than park. As a there are no rules or formulas the parameter α will be set heuristically.

4.3. Occupancy Grid Creation

Figure 4.7 shows the basic steps for the occupancy grid creation. The inputs for this block are the range angle map created from the radar processing and the positional data received from the robot platform. First the sensor coordinates are translated into grid coordinates. After that the range angle map obtained in the signal processing chain is further processed using a sensor model. The last step is to incorporate the local gridmap with the translated positions into the global gridmap.

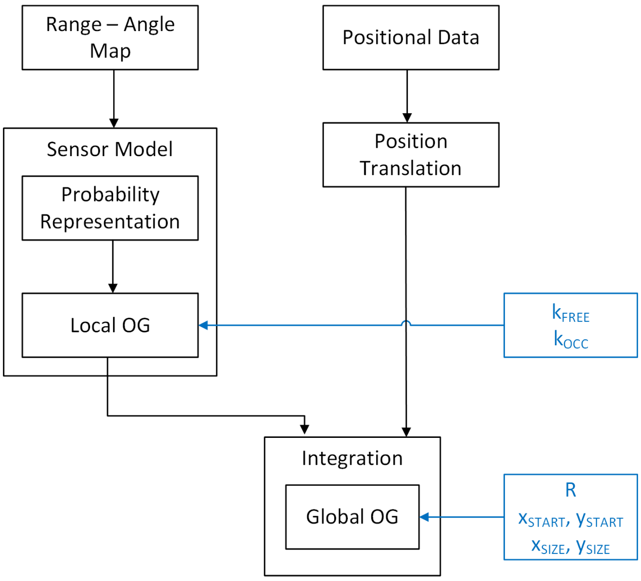


Figure 4.7.: Occupancy Grid Creation Overview

4.3.1. Sensor Model

The range angle map already is a geometric representation of the radar sensor's field of view and therefore also some kind of local map. To incorporate the information into an occupancy grid, the SNR values of each bin have to be converted into a probability of occupancy. Also additional knowledge for free space modeling based on the sensor's properties can be used. There exists no mathematically derived sensor model for the intended use case. During the thesis's course it turned out that a derivation of such a relation goes beyond it's scope. Therefore a heuristic approach will be used. Three different models will be investigated, where each model will be an extension of the previous one. The models aim to exploit the data provided by the radar sensor as good as possible. That also includes properties that are unique for radar e.g. it's capability of looking through certain objects. Each one will have the two tuneable parameters k_{OCC} and k_{FREE}. In general k_{OCC} is a weighting factor for detections and k_{FREE} weights free space. However the particular purpose depends on the model and will be described in the next sections.

Model 0: Direct SNR Transformation

The most basic model is to simply convert the SNR value of every bin of the range angle map into a probability of occupancy and directly integrate this values into the global gridmap following the assumption that noise is treated as free space. To convert a SNR value into a probability of occupancy two basic questions have to be addressed:

- Which bins can be classified as "free" and which ones as occupied?
- How should the SNR values be converted into probabilities of occupancy?

Looking at Equation 4.3 gives a clearer picture for answering that question: The SNR consists of two components the noise power P_N and signal power P_S. The higher the SNR, the bigger P_S and the more probable a cell is occupied. However no general decision boundaries for the SNR can be derived at this point since the RCS is an completely unknown parameter in the mapping context. The second component influencing the SNR is the actual present noise level. It can have a lot of different sources. Some of them may be external e.g. scattering objects or environments, and others can be sensor-internal e.g. thermal- or phase noise. Noise occurs with a certain power on the whole bandwidth of the received signal. It's presence might indicate that a bin indeed doesn't contain a reflecting object, but the signal might also have a completely different source. This issue will be given more attention later. From the literature two relations are good candidates for a mapping between the SNR and the probability of occupancy:

SNR Only Based ($p_{SNR}(\boldsymbol{b}_i)$): A similar form of this relation is already used in [3]. Combining Equations 3.37 and 3.38 yields

$$p_{SNR}(\boldsymbol{b}_i) = 1 - \frac{1}{1 + z_i} \tag{4.4}$$

$\boldsymbol{b}_i$ denotes the bin i in the range angle map, and z_i the measured SNR value for $\boldsymbol{b}_i$. The physical and probabilistic justification of this relation can be derived from the SNR definition in Equation 4.3. SNR values range from 0 to ∞. Equation 4.4 scales them into the interval from 0 to 1. Also, if $P_S = 0$ (which means the measured SNR equals the estimated noise floor) would mean $p(\boldsymbol{b}_i) = 0.5$. This result makes sense since strictly speaking noise is considered non-informative. Note that the used CFAR-threshold is not considered in the probability calculation. Figure 4.8 shows an example of the SNR based representation with it's corresponding range angle map. Note that in the practical use-case values of $p_{SNR}(\boldsymbol{b}_i) < 0.5$ can occur. This is because the noise power P_N is estimated by the CFAR algorithm. If SNR values in the range angle map are below this estimated power, $p_{SNR}(\boldsymbol{b}_i)$ is smaller than 0.5. Therefore SNR values below the noise-floor are considered as free space.

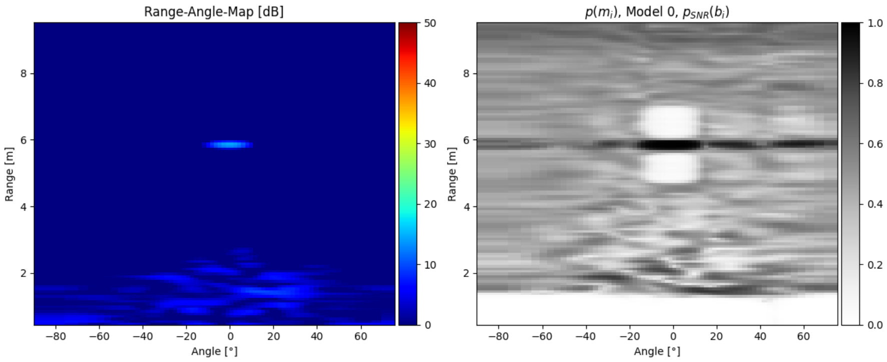


Figure 4.8.: $p_{SNR}(\boldsymbol{b}_i)$ based probability representation with the corresponding range angle map.

Detection Probability ($p_{PD}(\boldsymbol{b}_i)$): The second possible relation is using the already introduced detection probability. The measured SNR values can be transformed in a probability of occupancy by means of equation 3.23.

$$p_{PD}(\boldsymbol{b}_i) = p_D(z_i) \tag{4.5}$$

Again z_i is the measured SNR value. This model already is a mathematical derivation for the used CFAR detector. It describes how probable a measured SNR value is a detection, given the threshold α. For a better understanding figure 4.9 in principal illustrates the purpose of the CFAR framework. Two hypotheses are shown with their corresponding PDF's: the noise- and the detection hypothesis. The threshold is the decision boundary to choose the most likely hypothesis e.g. if the measurement is classified as detection or noise. The areas under the PDF's represent the probabilities for the noise and detection hypothesis respectively. From the graph four possible cases can be extracted:

- Noise classified as detection: The "false alarm" case, where a detection is triggered by the mere presence of noise. It is represented by the area under the noise PDF which is right to the threshold (p_{FA}). The CFAR framework sets the threshold such that the false alarm rate (Equation 3.19) is constant on average (Equation 3.22).
- Object classified as detection: A SNR value that is is bigger than the threshold triggers the detection. The probability p_D, reflected by the area under the detection PDF right to the threshold.
- Noise classified as noise: Noise is correctly classified. This happens with the probability $1 - p_{FA}$ as it is represented by the area under the noise PDF which is left of the threshold.
- Object classified as noise: An object reflects not strong enough such that the SNR value is below the threshold. The corresponding probability is $1 - p_D$, represented by the area under the detection PDF which is left of the threshold.

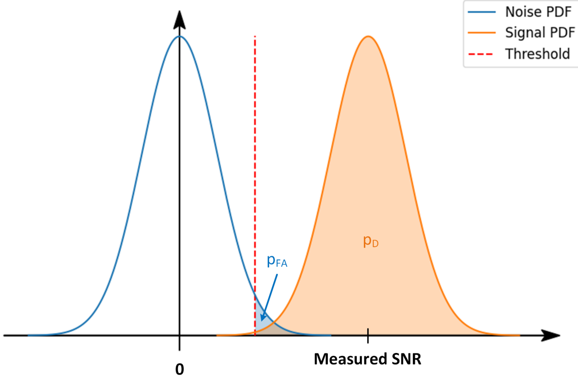


Figure 4.9.: Graphical illustration of the detection probability

An advantage of this method is, that the CFAR relevant parameters (N and α) are already included in the derivation. On the other hand the probability is always with respect to the used threshold. This requires it to be perfectly set - a condition which can never be guaranteed for different scenarios as has already been discussed in Section 4.2.3.

An important property of p_D is that it reflects the probability that the measured SNR is a detection. $p_D = 0$ would mean that the measured

SNR definitely will be classified as noise while $p_D = 1$ means that it definitely will be classified as detection. In the occupancy grid framework a detection probability of 0 would mean that the corresponding cell is certainly free. The presence of noise however doesn't necessarily indicate free space. That means a bin classified as noise is treated as certainly free space by the function $p_{PD}(\boldsymbol{b}_i)$. To handle this discrepancy, additional shaping is necessary. This will be done in the next step. Figure 4.10 shows an example of the unshaped, detection probability based representation with it's corresponding range angle map.

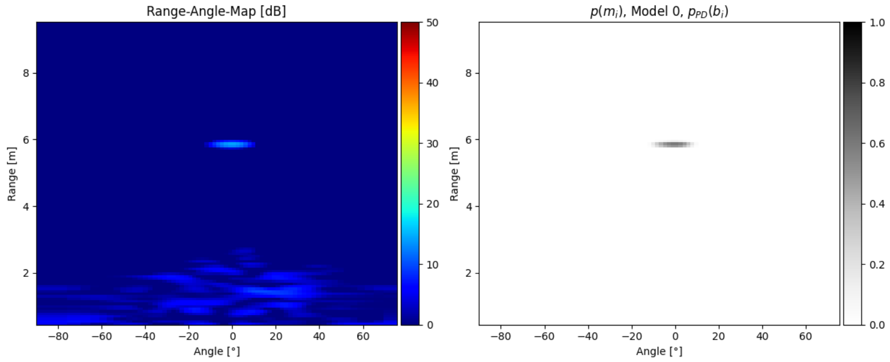
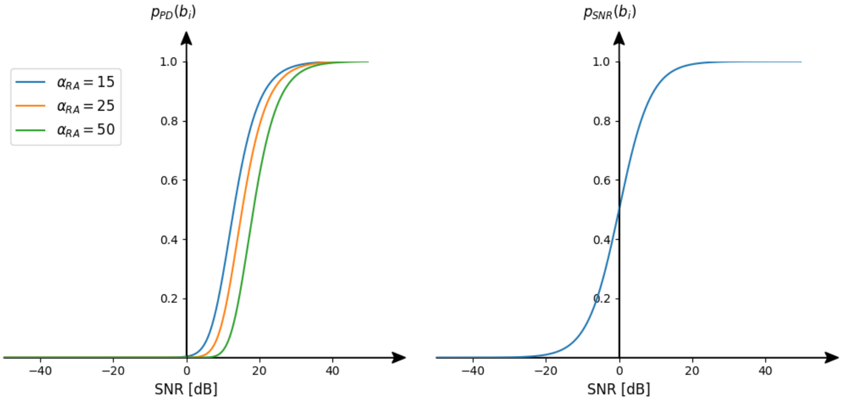


Figure 4.10.: $p_{PD}(\boldsymbol{b}_i)$ based probability representation with the corresponding range angle map.

Figure 4.11 shows a comparison of the two above described approaches. Both of them just give a rough transformation from a SNR value to a probability. Further shaping of this probabilities to incorporate knowledge how probable high- and low SNR values should be integrated into the final grid is necessary. In other words the credibility of a single measurement should be tune-able. When using $p_{PD}(\boldsymbol{b}_i)$ this is necessary anyway. It is done with the relation

$$p(\boldsymbol{m}_i) = p_X(\boldsymbol{b}_i) \cdot (k_{OCC} - k_{FREE}) + k_{FREE} \tag{4.6}$$

As there a two different proposals for transforming the SNR into a probability both will be evaluated in all models. That means that for $p_X(\boldsymbol{b}_i)$, $p_{SNR}(\boldsymbol{b}_i)$ (the SNR only based approach, Equation 4.4) as well as $p_{PD}(\boldsymbol{b}_i)$ (the detection probability based approach, Equation 4.5) might be used.



(a) $p_{PD}(\boldsymbol{b}_i)$

(b) $p_{SNR}(\boldsymbol{b}_i)$

Figure 4.11.: Comparison of $p_{PD}(\boldsymbol{b}_i)$ and $p_{SNR}(\boldsymbol{b}_i)$.

To adhere the laws of probability the parameters will be chosen from the interval

$$0.5 \leq k_{OCC} \leq 1 \tag{4.7}$$

$$0 \leq k_{FREE} \leq 0.5 \tag{4.8}$$

The parameter k_{FREE} tunes the weight for free space and k_{OCC} for occupied space respectively. This model has no explicit modeling for free space. The choice of k_{FREE} decides on how sure a SNR value in the vicinity of the noise floor should be considered as free. Choosing $k_{FREE} = 0$ would mean to interpret the absence of a detection as free space and $k_{OCC} = 1$ to consider a detection as certainly occupied. Figure 4.12 shows the shaped probability representations. By tuning the corresponding parameters noise is less weighted as free and detections are less weighted as occupied. In the last chapter the overall performance of both approaches will be evaluated.

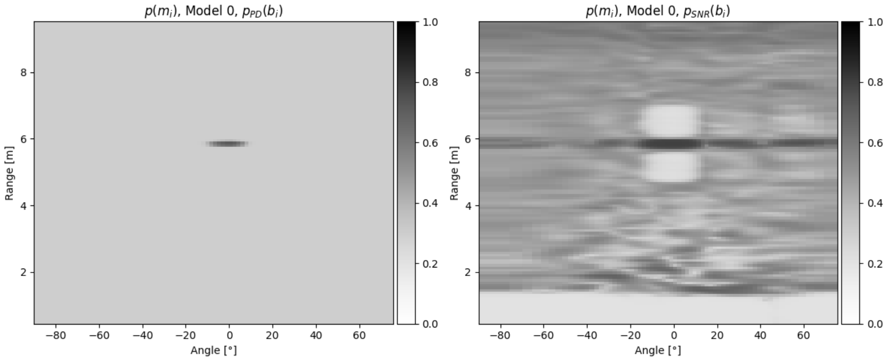


Figure 4.12.: Shaped probability representations. Left: $p_{PD}(\boldsymbol{b}_i)$, $k_{OCC} = 1$, $k_{FREE} = 0.3$. Right: $p_{SNR}(\boldsymbol{b}_i)$, $k_{OCC} = 0.8$, $k_{FREE} = 0.2$. The shaping transforms the values of $p_X(\boldsymbol{b}_i)$ into probabilities of occupancy that are more compliant to the occupancy grid framework.

Model 1: Detection Only

For model 0 all bins of the range angle map containing noise are considered as free space. To achieve higher quality gridmaps a more detailed analysis of the interpretation of noise is necessary. A problematic scenario can be seen in Figure 4.13. The left figure shows an image of the scene. The sensor is directed straight on a parked car. The right plot is the corresponding range angle map. The park car is well detected at a distance of approximately 2.5 meters. All around the reflecting surface of the car is noise. When generally treating noise as free, even the area behind the reflecting surface - which is inside the car - is considered free. However the waves sent by the sensor cannot penetrate this surface and therefore the area behind it should be considered non-informative. The reason why there is a signal at this bins is noise. Every bin in the range angle map carries at least noise no matter if the sent electro-magnetic waves can't even reach this certain point in space. The resulting probability representations can be seen in figure 4.14. Both representations show free space inside the parked car. For $p_{SNR}(\boldsymbol{b}_i)$ it seems not as severe as for $p_{PD}(\boldsymbol{b}_i)$ in this illustration though.

To overcome this problem noise in front of a detection is considered as free space, noise after the last detection is considered non-informative. The weight of the free space is tuned by k_{FREE}. Detections are considered as

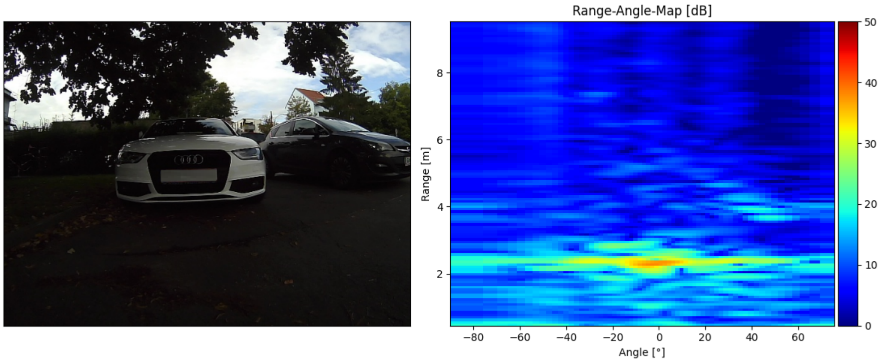


Figure 4.13.: Problematic scenario for model 0. Right: Image of the scene. Left: Corresponding range angle map

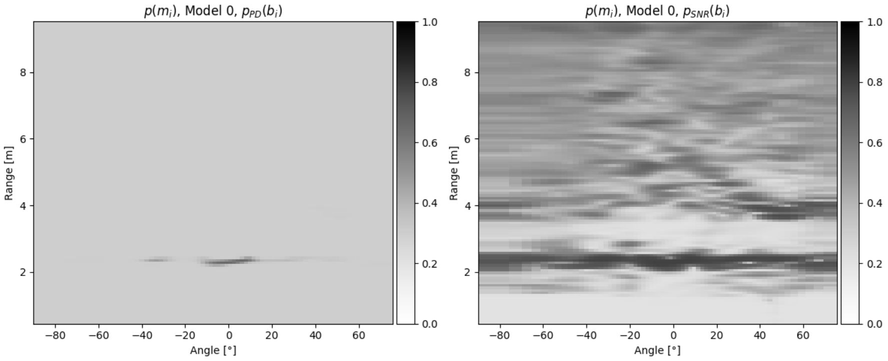


Figure 4.14.: Probability representations for the problematic scenario in Figure 4.13.

occupied space with the parameter k_{OCC}. This is very similar to the model used in [3]. However there are some major differences in the original use-case of the model:

- Originally the radar sensor is mechanically scanned with a preferably narrow main lobe, whereas the actual use case is a sensor with an antenna array.
- The radar sensor has been on a static position whereas the later conducted measurements will be with changing positions.
- The original model distinguishes between wide and narrow pulses when updating grid cells classified as free.

- Originally the model is used directly in a log-odds form, with a single parameter. Here it will be used in it's likelihood form with two parameters.
- The update function for occupied space in the original use-case is always just $p_{SNR}(\boldsymbol{b}_i)$. In this thesis the two cases $p_{SNR}(\boldsymbol{b}_i)$ and $p_{PD}(\boldsymbol{b}_i)$ will be evaluated.

The adapted model can be written as:

$$p(\boldsymbol{m}_i) = \begin{cases} p_X(\boldsymbol{b}_i) \cdot k_{OCC} & \text{if detection} \\ k_{FREE} & \text{if in front of detection} \\ 0.5 & \text{otherwise} \end{cases} \tag{4.9}$$

In case a bin $\boldsymbol{b}_i$ is classified as detection by the CFAR algorithm, the probability representation is scaled by k_{OCC}. All bins in front of a detection with the same angle are updated with a probability represented by k_{FREE}. Everything else is considered non-informative therefore 0.5. Figure 4.15 illustrates the detection based approach. Again for $p_X(\boldsymbol{b}_i)$, $p_{SNR}(\boldsymbol{b}_i)$ as well as $p_{PD}(\boldsymbol{b}_i)$ might be used. The difference to model 0 is however, that the transformation from a SNR into a probability is just used for detections.

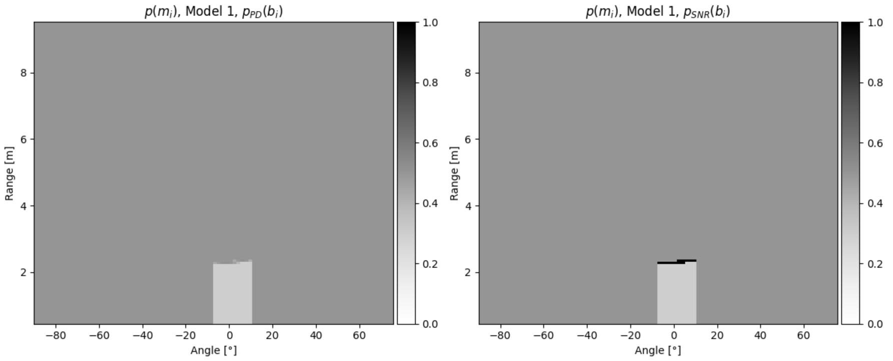


Figure 4.15.: Illustration of model 1. Left: $p_{PD}(\boldsymbol{b}_i)$, $k_{OCC} = 1$, $k_{FREE} = 0.3$. Right: $p_{SNR}(\boldsymbol{b}_i)$, $k_{OCC} = 1$, $k_{FREE} = 0.2$

Model 2: Incorporated Beam Pattern

Model 1 introduces a very big shortcoming. Without a detection nothing is considered as free space. Figure 4.16 illustrates a case where this is problematic. No detecions are in front of the sensor but most of the space could be considered free. This is a very important scenario as it represents one of the most likely use-cases in an automotive application; i.e. driving on a road. The probability representation for model 1 can be seen in Figure 4.17. In the shown scenario a car is detected to the sides of the path, and consequently the area in front of the car is considered free. The part that would be important for further path-planning - the empty space in front the sensor - is considered non-informative. To overcome this problem the connection between noise and free space has to be modeled in more details. The sensor's antenna array has it's highest sensitivity at an angle of 0° and it decreases further outward towards ±90°. The higher the sensor's sensitivity the more reliable noise can be considered as free space. The correlation of the sensor's sensitivity with the angle is given by the beam pattern. It is an antenna- (and therefore sensor-) specific characteristic and can be seen at the left side of Figure 4.18. The right side shows the transformed function $f_{FREE}(\phi, k_{FREE})$ that is used for updating free-space depending on the angle ϕ. The following manipulation steps are necessary for the transformation:

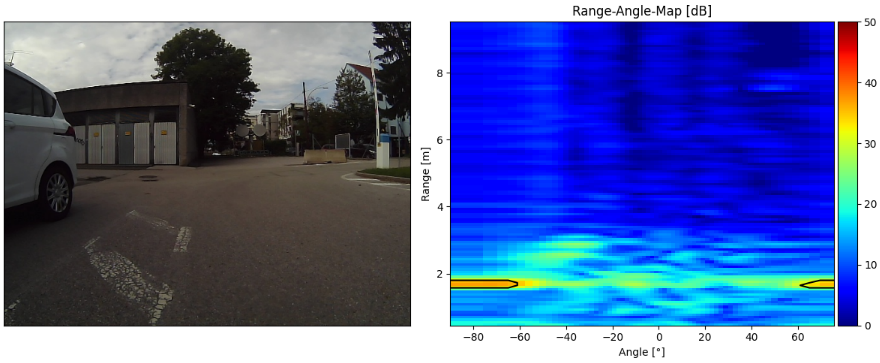


Figure 4.16.: Problematic scenario for model 1.

1. Transform the logarithmic scale beam pattern into a linear scale.

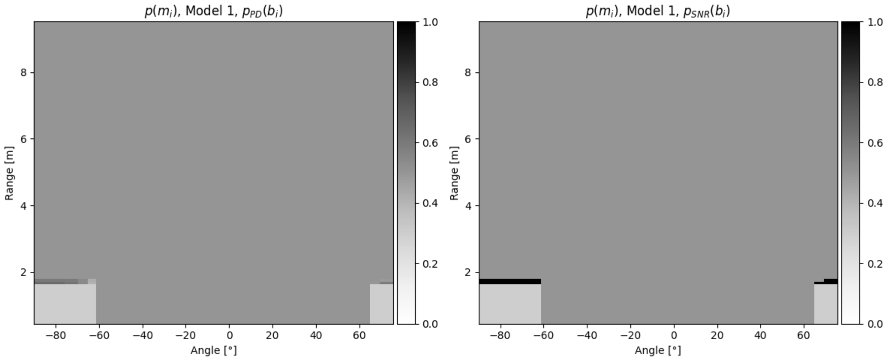


Figure 4.17.: Model 1 without obstacle in front of the sensor.

2. Invert the function. This is necessary as a high SNR in the beam pattern reflects a high certainty for the detection of objects. The more certain objects are detected, the more the absence of a detection indicate free space.
3. Scale and shift the function such that 90°≙0.5 (non-informative) and 0°≙k_{FREE} (maximal certainty that no object is missed). Again the parameter k_{FREE} can be used to tune the factor for weighting free space.

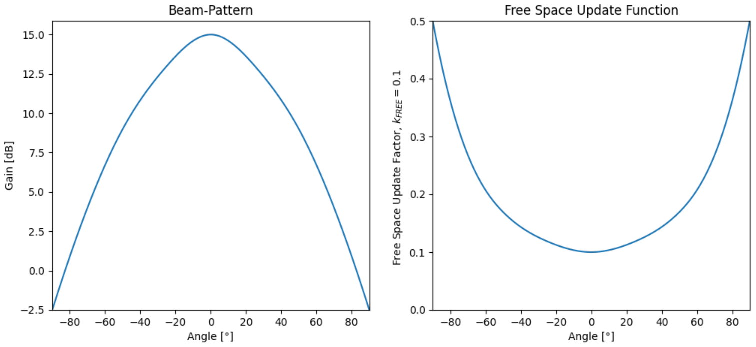


Figure 4.18.: Beam pattern (left) and derived free space update function with $k_{FREE} = 0.1$ (right).

Figure 4.19 shows the in Figure 4.16 depicted scenario with the beam pattern enhanced model. Now the space directly in front of the sensor, that has previously been marked as non-informative is now considered free. The influence of the update function on the mapping of free space is especially visible for higher angles.

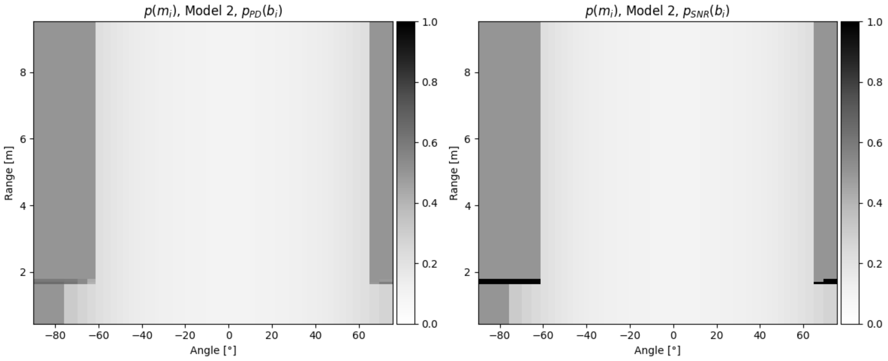


Figure 4.19.: Illustration of the beam pattern based sensor model.

The final model utilizing the beam pattern can be written as:

$$p(\boldsymbol{m}_i) = \begin{cases} p_X(\boldsymbol{b}_i) \cdot k_{OCC} & \text{if detection} \\ f_{FREE}(\phi, k_{FREE}) & \text{if in front of- or no detection} \\ 0.5 & \text{otherwise} \end{cases} \tag{4.10}$$

As for the two models before for $p_X(\boldsymbol{b}_i)$, $p_{SNR}(\boldsymbol{b}_i)$ as well as $p_{PD}(\boldsymbol{b}_i)$ might be used. Like for model 1 it will only transform the SNR of detections.

4.3.2. Position Translation

Each bin of the probability representation is referenced with polar, local sensor coordinates. They have to be translated into Cartesian global grid coordinates. Figure 4.20 shows exemplary how the sensor's FOV might be placed into the global occupancy grid. It is described by the range r and the angle ϕ. The global position of the sensor is given by the coordinates x_S and y_S. It's orientation by the bearing angle Θ. Those values are provided by the mapping-package used within ROS.

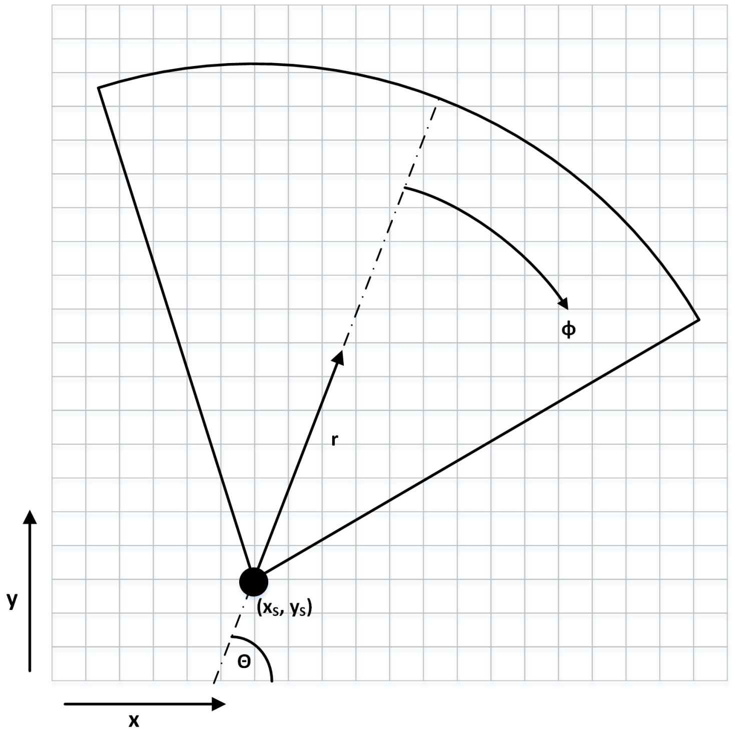


Figure 4.20.: Position translation overview.

The global position of a single bin in the range angle map is calculated according to

$$x_i = r \cdot sin(\phi + \Theta) + x_S \tag{4.11}$$

$$y_i = r \cdot cos(\phi + \Theta) + y_S \tag{4.12}$$

4.3.3. Integration

To integrate the local probability representation into the global grid map first the translated bin positions have to be mapped to grid-cells. The correspondence can simply be calculated using the formulas

$$m_X = \left\lfloor \frac{x_i}{R} + x_{START} \right\rfloor \tag{4.13}$$

$$m_Y = \left\lfloor \frac{y_i}{R} + y_{START} \right\rfloor \tag{4.14}$$

R is the resolution of the occupancy grid. It will be set to a reasonable value in terms of necessary map resolution, sensor capabilities and empirical values from related publications. x_{START} and y_{START} can be used to set the initial robot position with respect to the scenario. The number of grid cells can be set using x_{SIZE} and y_{SIZE} for the corresponding dimension.

The local representation will be transformed into it's log-odds form using Equation 3.37. As final step the global map is updated by integrating all bins of the local occupancy representation that fall into their corresponding global grid-cells. The update of the global map can be done as described in Equation 3.39.

5. Evaluation

In this chapter the performance of the developed sensor models should be evaluated. In order to do this, occupancy grids are created from measurements using the developed models. The performance evaluation is based on the quality of the resulting gridmaps.

At first the used robot platform with it's mounted sensors is shown and the used components especially the radar sensor are specified. Next the choice of the measurement scenarios is discussed. The scenarios have been chosen in a way that certain properties of the radar sensor could be examined, and the influence on the resulting gridmap derived. After that the choice of the parameters used for the processing pipeline is explained.

As a basis for assessing the quality of the final maps reference occupancy grids generated with a laser scanner are used. To further formalize the quality of the results, criteria are introduced that generally describe the map quality. More specifically they categorize features that caused a reduced quality. This criteria are applied on the occupancy grids created from the measurements. It is investigated what effects of the sensor caused certain criteria to occur. Finally there will be a discussion of the results.

5.1. Robot Platform

To perform measurements that are close to a real use-case they needed to be dynamic. The robot platform has been steered through the different scenarios on certain paths. Therefore it needed to be mobile and as well carry and connect all the sensors necessary for the measurements. Important at this point are the localization task it needed to perform as well as the reference occupancy grid creation using a laser-scanner. For a more detailed explanation Section 4.1 can be revisited. Figure 5.1 shows the setup, pointing out the important components for the measurement as described in section 4.1. The used components of the setup were:

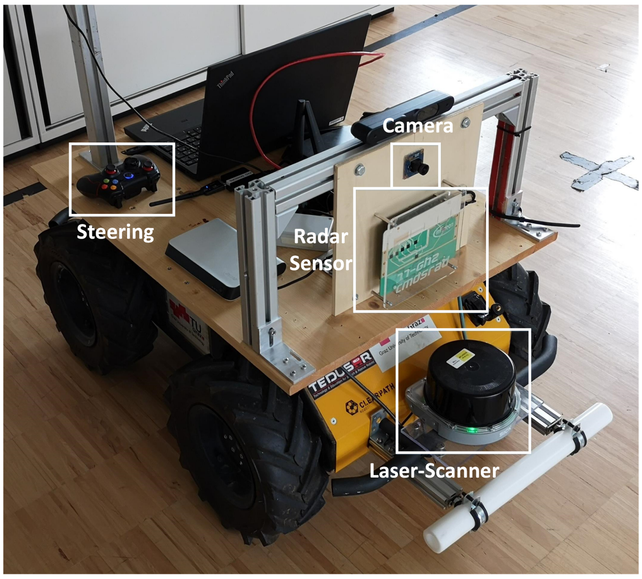


Figure 5.1.: Robot platform with the components used for the measurements.

Robot Platform: Clearpath Husky UGV[1] with custom top-plate.

ROS: Version Kinetic[2]

Mapping Package: Hector Mapping[3], Resolution: 0.1cm, Update Factor Free: 0.4, Update Factor Occupied: 0.7. A cell-size of 10cm has been used. This package was used over Gmapping because it was already fully functional on the used platform.

Laser Scanner: SICK MRS1000[4]

Steering: USB Game pad, EasySMX ESM 9013 Wireless 2.4G Game Controller[5].

[1] clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/
[2] wiki.ros.org/kinetic
[3] wiki.ros.org/hector_slam
[4] sick.com/ag/en
[5] easysmx.com/products/easysmx-esm-9013-wireless-gaming-controller

5.1.1. Radar Sensor

The radar sensor will be given special attention at this point, as it is the most important part of the setup. The sensor RadarLog[6] of the company Inras has been used. It is special since it can provide raw data measurements. The antenna-fronted has two transmit and 16 receive antennas and operates at 77 GHz. The full sensor configuration can be seen at Table 5.1.

Parameter	Value	Parameter	Value
N	256	N_{CYCLES}	64
f_{START}	$76GHz$	f_{STOP}	$78GHz$
T_P	$200\mu s$	T_{INT}	$200ms$
T_{SW}	$128\mu s$	T_{RD}	$32\mu s$
$N_{CHANNELS}$	16	N_{TXSEQ}	1
f_s	$2MHz$		

Table 5.1.: Radar sensor configuration.

The choice of f_{START} and f_{STOP} results in a sweep-bandwidth of $B_{SW} = 2\,GHz$. T_{RD} is the ramp-down time after the up-chirp. It's value has been empirically derived. N_{TXSEQ} is the number of sequences that has been transmitted during a chirp. T_{INT} is the overall update frequency of the sensor. It is set such that a overall frame rate of 5 Hz is achieved. This may seem rather low compared to commercial sensors that work with rates of up to 25 Hz, but these sensors don't transmit raw data. Frame rates that high proved impossible with the used sensor, because they resulted in corrupted data. The most probable reason for this is that the sensor's internal components are not able to handle the data traffic. Apart from that, the resulting data-files quickly get a size where further processing gets difficult (especially the conversion into the final python pickle format) even with the low frame rate. As the maximum speed of the Husky platform is approximately $1m/s$, 5 Hz is easily sufficient for providing measurements for a decent occupancy mapping.

[6] inras.at/produkte/radarlog.html

5.2. Measurement Scenarios

The measurement scenarios are a key component for an analysis of the developed framework. They provide real data to test the derived models on. By choosing different setups for every scenario a more detailed influence of environment and sensor properties on the resulting gridmaps can be done. Therefore two different types of measurements have been conducted. The first type were regulated scenarios. They aimed to provide measurements in as much as possible controlled environments. This scenarios have been chosen such that certain properties of the sensor can be examined. The second type were real-world scenarios. For them a set of relevant real-world environments have been chosen. They should give a more detailed insight in the overall performance for later applications.

5.2.1. Regulated Scenarios

For this environments simple scenarios were created with a previously defined number and placement of objects. To be able to perform the measurements without interference of the environment with respect to noise and reflections a free area had been chosen. Due to availability of the location as well as the robot platform two different areas have been used for this purpose as shown in Figure 5.2. Two major factors will be investigated: the influence of the RCS and the ability of the sensor to detect objects behind objects. A total number of six objects have been chosen for the measurements. To cover the necessary aspects three different materials (plastic, wood, metal) and 2 different sizes (small, big) have been used. Figure 5.3 shows pictures and Table 5.2 a more detailed description of the used objects.

For investigating the influence of the RCS occupancy grids of single objects have been created. As described in Section 3.1.1 two parameters of the RCS depend on the object itself: size and material. To cover this two aspects for all six objects an occupancy grid has been created by approaching it straight. The third RCS-influencing parameter, the directivity, depends on the relative position between object and sensor. To cover this aspect, for all six objects an occupancy grid was created by approaching it straight, but shifted by a certain distance. Figure 5.4a illustrates this measurements.

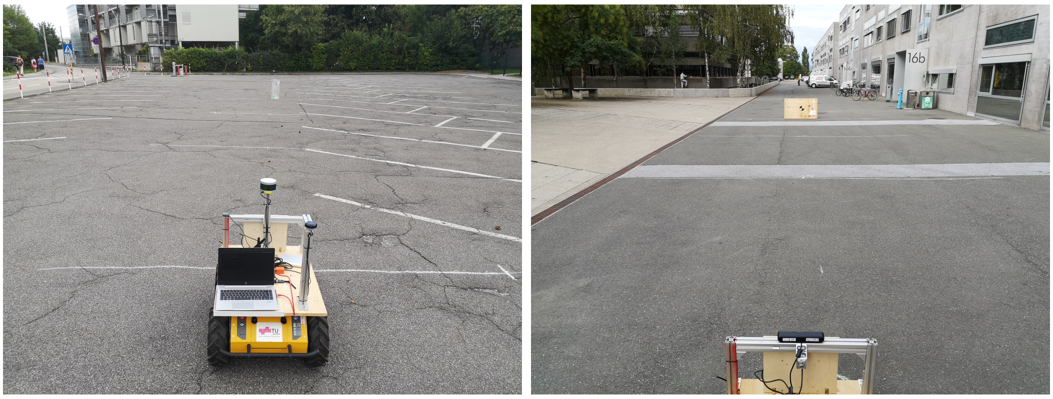


Figure 5.2.: Measurement areas for the regulated scenarios. Left: Area 1. Right: Area 2.

Description	Size [cm]	Figure
Metal pole	6x4x100	5.3a
Plastic pole	⌀7.5x106	5.3b
Wood pole	4.5x4.5x82	5.3c
Metal box	76x37x38	5.3d
Plastic box	60x40x22	5.3e
Wood box	132x80x5	5.3f

Table 5.2.: Measurement objects Description.

For investing the sensor's capability to detect something behind an object two objects are necessary. Therefore occupancy grids were created by placing two of the three big objects behind each other with two different distances in all possible combinations by approaching the first object straight. The principle can be seen in Figure 5.4b.

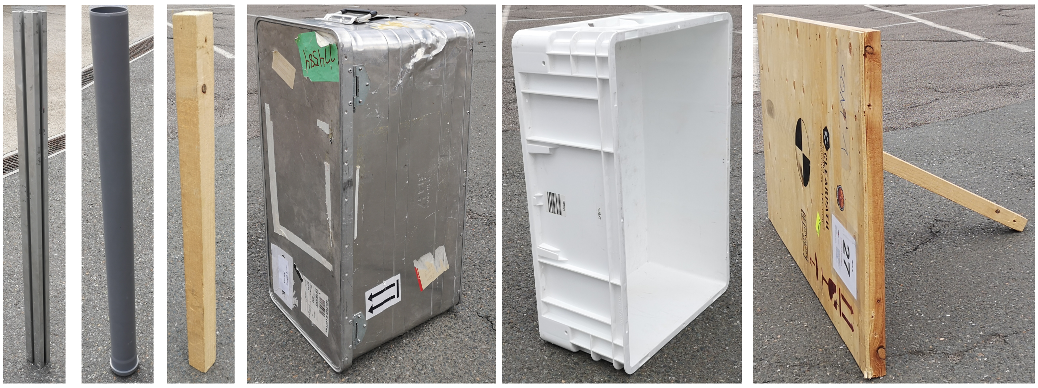
Tables 5.3 and 5.4 summarize the conducted regulated experiments.

5.2.2. Real-World Scenarios

Figure 5.5 shows the environments for the real-world scenarios. Four different environments have been chosen, three of them automotive related scenarios and one more general scenario. Figure 5.5a shows a scene where

Nr.	Object	Offset	Area	Localization
1	Metal pole	0m	1	SLAM
2	Metal pole	2m	1	SLAM
3	Metal pole	4m	1	SLAM
4	Plastic pole	0m	1	SLAM
5	Plastic pole	2m	1	SLAM
6	Plastic pole	4m	1	SLAM
7	Wood pole	0m	1	SLAM
8	Wood pole	2m	1	Odometry
9	Wood pole	4m	1	Odometry
10	Metal box	0m	2	Odometry
11	Metal box	2m	2	Odometry
12	Metal box	4m	2	Odometry
13	Plastic box	0m	2	Odometry
14	Plastic box	2m	2	Odometry
15	Plastic box	4m	2	Odometry
16	Wood box	0m	2	Odometry
17	Wood box	2m	2	Odometry
18	Wood box	4m	2	Odometry

Table 5.3.: Single object measurements scenarios. The measurement setup can be seen in Figure 5.4a. The robot platform approaches the object shifted by the given offset.



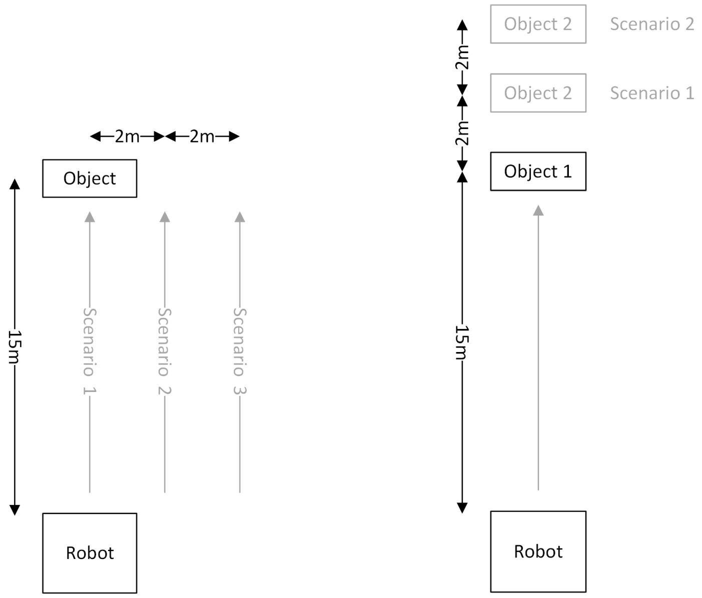
(a) (b) (c) (d) (e) (f)

Figure 5.3.: Objects used for the measurements described in Table 5.2. (a) Metal pole; (b) Plastic pole; (c) Wood pole; (d) Metal box; (e) Plastic box; (f) Wood box.

cars are parked along a road. It is a classic ”driving along a road scenario”. The second environment is shown in Figure 5.5b. It poses a classic parking lot scene. In the third automotive scenario a single car is examined. It can be seen in Figure 5.5c. The last scenario is a scene taken at the campus, shown at Figure 5.5d. It contains mainly trees and concrete with little metal surfaces.

Nr.	Object 1	Object 2	Offset	Area	Localization
1	Metal box	Plastic box	2m	1	SLAM
2	Metal box	Plastic box	4m	1	SLAM
3	Metal box	Wood box	2m	1	SLAM
4	Metal box	Wood box	4m	1	SLAM
5	Plastic box	Metal box	2m	1	SLAM
6	Plastic box	Metal box	4m	1	SLAM
7	Plastic box	Wood box	2m	1	SLAM
8	Plastic box	Wood box	4m	1	SLAM
9	Wood box	Metal box	2m	1	SLAM
10	Wood box	Metal box	4m	1	SLAM
11	Wood box	Plastic box	2m	1	SLAM
12	Wood box	Plastic box	4m	1	SLAM

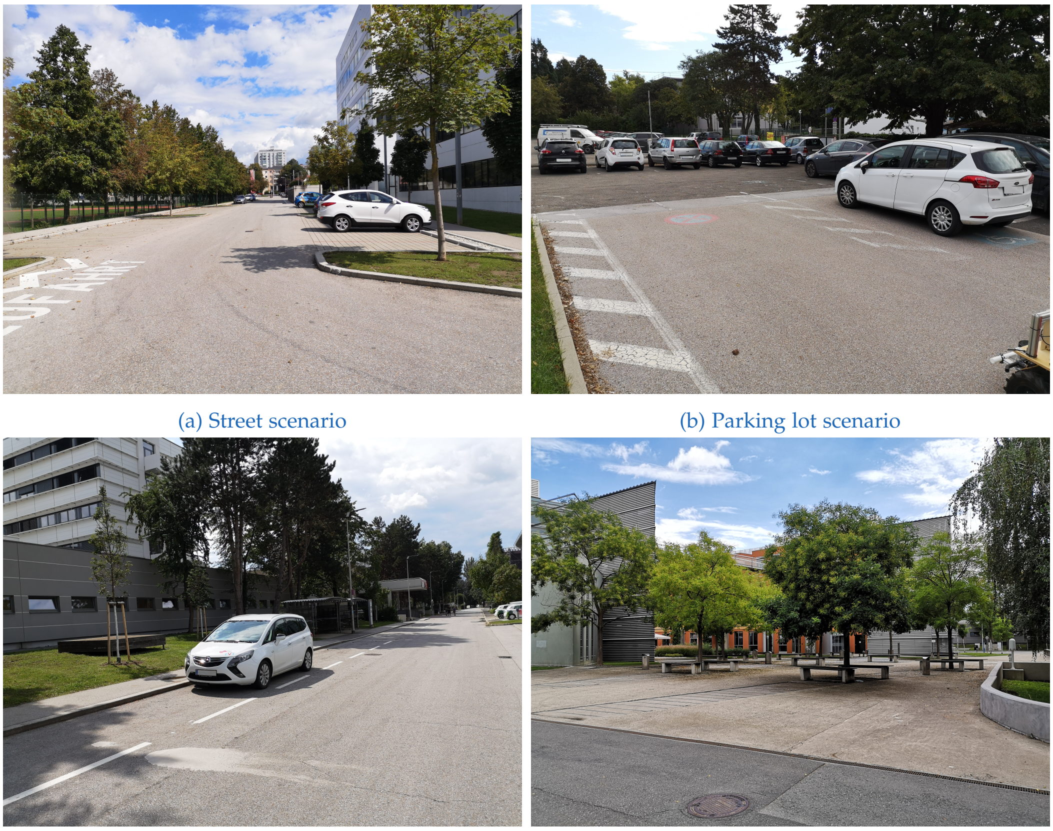
Table 5.4.: Two object measurements scenarios. The measurement setup can be seen in Figure 5.4b. The offset describes the distance between the two objects.



(a) Single object scenarios.

(b) Two object scenarios.

Figure 5.4.: Illustration of the regulated scenarios.



(a) Street scenario

(b) Parking lot scenario

(c) Single car scenario

(d) Campus scenario

Figure 5.5.: Real-world scenario environments.

5.3. Processing Parameters

The following section will briefly describe the used parameters for the processing. The theoretical background has already been discussed in Chapter 4.

5.3.1. Radar Signal Processing

r_{CMIN}, r_{CMAX}: Using Equation 3.11 and the sensor configuration given in Table 5.1 the maximum achievable range is approximately $r_{MAX} \approx 10m$. Naturally the minimum range is $r_{MIN} = 0m$. Figure 5.6 shows a full representation of the range doppler and range angle map without cropped range dimension. Very high SNR components in the vicinity of $r = 0$ are visible. They are a result from DC-noise in the receiver chain which is characteristic for radar sensors. To avoid this high noise components a feasible value for cropping the minimum range is $r_{CMIN} = 0.5m$. As there are no high noise components at high ranges the upper end of the range dimension won't be cropped.

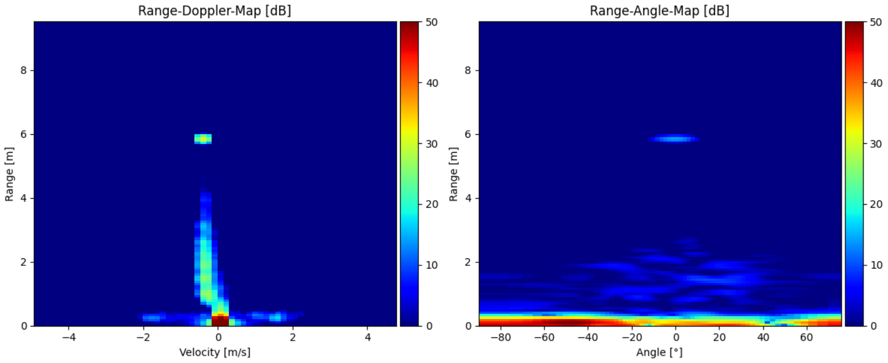


Figure 5.6.: Range doppler and range angle map without cropped range dimension.

N_{WIN}, N_{GUARD}: The range angle map of Figure 5.6 shows a peak at a distance of about $6m$. It has a width (in the range-dimension) of approximately $0.5m$. To obey the assumption of the CFAR-algorithm that the neighboring cells for estimating σ_w^2 only contain noise (see Section 3.1.5) a minimum total number of guard cells of $0.5m/\Delta r \approx 7$ is necessary. Considering leading and lagging part of the window $N_{GUARD} = 4$ is chosen. Assuming detections will be spaced with at least one meter and still providing a feasibly low false-alarm rate as well as a feasible accurate noise estimate $N_{WIN} = 10$ is chosen.

$\mathbf{N_{RANGE}}$: As already suggested calculating $r_{MAX}/\Delta r$ gives the most feasible value. By using the values of Table 5.1 as well as Equations 3.11 and 3.12 it is set to $N_{RANGE} = 256$ which perfectly reflects the configured number of samples N for a single sweep.

$\mathbf{N_{ANGLE}}$: Physically the resolution is set by the number of used antennas, therefore the lowest choice would be $N_{ANGLE} = 16$. However the experiments showed that it makes sense to set it higher. A value of $N_{ANGLE} = 64$ turned out to be a feasible choice since peaks are more distinct but the angular dimension is not overly dense described.

$\boldsymbol{\alpha_{RA}}$: A set of values has been empirically derived. For that purpose the parking lot and the campus scenario (Figures 5.5b and 5.5d) were chosen as reference as they provide two key and yet contrary applications. Table 5.5 shows the values for α_{RA} that have proven to be feasible choices. This parameters are tested for the parking lot and the campus scenario for every model together with the parameter sets for k_{FREE} and k_{OCC} in the next section. The value for α_{RA} with the best results has then been further applied for the remaining scenarios for the corresponding model.

$\boldsymbol{\alpha_{RA}}$	15	25	50

Table 5.5.: Parameter candidates for α_{RA}.

Note: As the parameters α_{RD} and $N_{VELOCITY}$ do not influence the range angle map, but only the object list which is just used for illustration purposes in this thesis, they weren't mentioned at this point.

5.3.2. Occupancy Grid Creation

R: According to Equation 3.12 the range resolution with the used radar sensor configuration is $\Delta r \approx 8cm$. This as well as approaches in the literature suggest that a cell size of $R = 10cm$ is a feasible choice.

$\mathbf{x_{START}}$, $\mathbf{y_{START}}$, $\mathbf{x_{SIZE}}$, $\mathbf{y_{SIZE}}$: As those parameters simply influence the visualization of the occupancy grid, their choice is of minor importance for the processing. The starting position is usually in the middle of the grid $x_{START} = y_{START} = 0.5$. A general grid size of $x_{SIZE} = y_{SIZE} = 1500$ (which results in a grid of 150mx150m for the chosen cell size) is used. For each scenario the plotted occupancy grid is cropped to the area containing measurement data. The big size of the gridmap has been chosen because a uniform value eased the configuration of the processing framework.

$\mathbf{k_{FREE}}$, $\mathbf{k_{OCC}}$: The quality of the final gridmap will strongly depend on the choice of those parameters. As the parameters α_{RA}, k_{FREE} and k_{OCC} are tightly connected, the final values are derived by using the values for α_{RA} in Table 5.5 together with parametric sweeps for k_{FREE} and k_{OCC}. The intervals used can be seen in Table 5.6. The sweeps were conducted for $p_{SNR}(\boldsymbol{b}_i)$ as well as $p_{PD}(\boldsymbol{b}_i)$ with all different modeling approaches described in Section 4.3.1 and have been applied to the parking lot and the campus scenario. The parameters with the best map quality have been chosen and can be seen at Table 5.7. The quality assessment of the maps was based on the criteria introduced in section 5.5. The interval for $p_{PD}(\boldsymbol{b}_i)$ is smaller and more densely sweeped than the interval for $p_{SNR}(\boldsymbol{b}_i)$. This is because $p_{SNR}(\boldsymbol{b}_i)$ has several points which are similar in quality over the whole range of k_{FREE} and k_{OCC} (Equations 4.7 and 4.8) whereas $p_{PD}(\boldsymbol{b}_i)$ yields meaningful results only in the given, restricted interval.

	$p(\boldsymbol{b}_i)$	**Start**	**Stop**	**Step**
$\mathbf{k_{OCC}}$	$p_{PD}(\boldsymbol{b}_i)$	0.4	0.5	0.01
	$p_{SNR}(\boldsymbol{b}_i)$	0	0.5	0.1
$\mathbf{k_{FREE}}$	$p_{PD}(\boldsymbol{b}_i)$	0.9	1	0.01
	$p_{SNR}(\boldsymbol{b}_i)$	0.5	1	0.1

Table 5.6.: Parameter sweeps for k_{FREE}, k_{OCC} and the two possibilities of $p(\boldsymbol{b}_i)$.

PR	Model	α_{RA}	$\mathrm{k}_{\mathrm{FREE}}$	$\mathrm{k}_{\mathrm{OCC}}$
$p_{PD}(\boldsymbol{b}_i)$	0	25	0.49	1
	1	15	0.45	1
	2	15	0.49	1
$p_{SNR}(\boldsymbol{b}_i)$	0	15	0.3	0.65
	1	25	0.25	0.8
	2	15	0.45	1

Table 5.7.: Parameter choices for the possibilities of $p(\boldsymbol{b}_i)$ and models.

5.4. Reference Occupancy Grids

The robot platform has been equipped with a laser-scanner which has been used to create occupancy grids for reference purposes. The following sections will briefly show and describe this reference gridmaps.

Note: In the reference gridmaps as well as in the radar generated occupancy grids the red line marks the path of the robot by means of the position data used for the grid creation. The red cross indicates the starting point.

5.4.1. Regulated Scenarios

Unfortunately the reference grids for the regulated scenarios are of quite poor quality and therefore couldn't really be used. However as the desired result is anyway known from the object sizes in Table 5.2 and the setup itself (Figure 5.4) this is of minor importance. One main reason for the poor quality is, that the measurements have been conducted consecutively without restarting the platform. This was necessary because otherwise the battery of the platform as well as the radar-sensor would have been depleted too fast. As a result, the reference grid often has fragments of previously measured objects as can be seen in Figure 5.7.

The second main reason why the reference grids for the regulated scenarios are often not usable is, that the positions that have been derived by the SLAM algorithm of the hector map very often diverge severely. The left part of Figure 5.8 shows the SLAM generated positions for such a case. It is

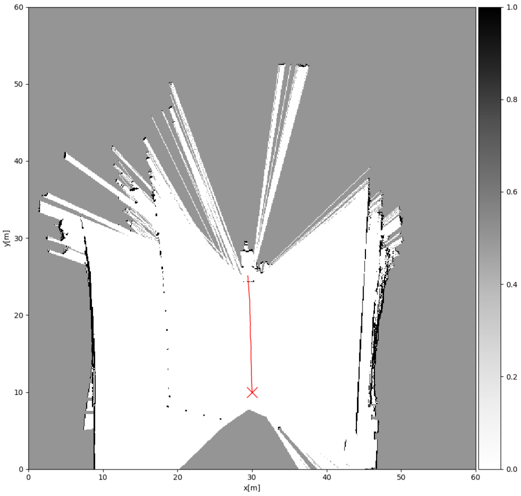


Figure 5.7.: Metal box/Plastic box 2m reference gridmap (Nr. 1 in Table 5.4).

clearly visible that the positions are completely wrong. This malfunction has quite certainly to do with the second measurement area used for the regulated scenarios. Apparently some features of the space in combination with the continuous driving back- and forth between the starting point and the object caused the position correction to fail. To minimize the influence of those errors, the positional data generated by the platforms odometry has been used instead. Unfortunately the quality of this data is not as accurate as a good SLAM-estimate as can be seen in the right side of Figure 5.8. This also reduces the quality of the radar-generated occupancy grid. However as the paths and setups for this scenarios are really simple and most objects are detected when the robot is rather close to the object, the measurements still bear useful information. Table 5.3 and 5.4 show on which measurement area a certain scenario has been conducted and if the SLAM- or odometry generated data has been used.

5.4.2. Real-World Scenarios

The generation of the reference grids for the real-world scenarios worked generally pretty well. They are shown in the Figures 5.9 to 5.12. The left

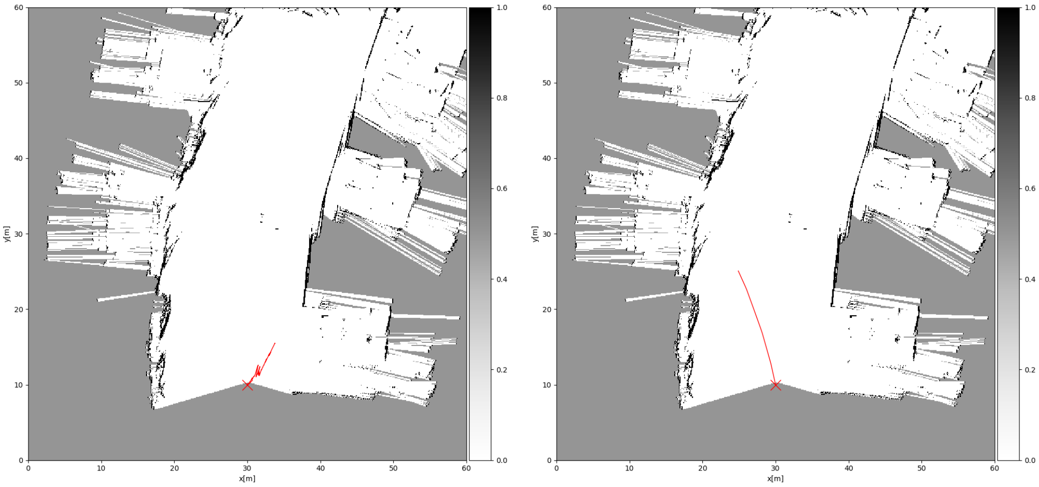


Figure 5.8.: Metal box $0m$ reference gridmap (Nr. 10 in Table 5.3. Ground truth: Figure 5.4a). Left: SLAM positions. Right: Odometry positions.

part of the figure always shows the approximate path, which was inserted by hand. The right part shows the actual positional data of the scenario.

The single car scenario (Figure 5.5c) and the street scenario (Figure 5.5a) show the expected map. In both cases the parking cars are the main features. They are clearly mapped, together with some features of the environment like trees and small buildings. The estimated paths match the recorded paths really well.

The campus scenario (Figure 5.5d) also worked really well. The trees, benches and surrounding walls are correctly mapped. The only discrepancy is a small hitch in the path shortly after the starting point ($x \approx 40m$ and $y \approx 10m$). It is probably also the reason for the fragments of the curved wall on the right end of the map.

Most problematic is the parking lot scenario (Figure 5.5b). The positional data provided by the SLAM algorithm (right side of Figure 5.12) completely deviates from the actual path which results in a completely corrupted map at some point. The car right to the starting point is correctly mapped. As the path after the starting point is slightly downwards, just the tires of the cars opposite to the starting point are visible. However their locations are

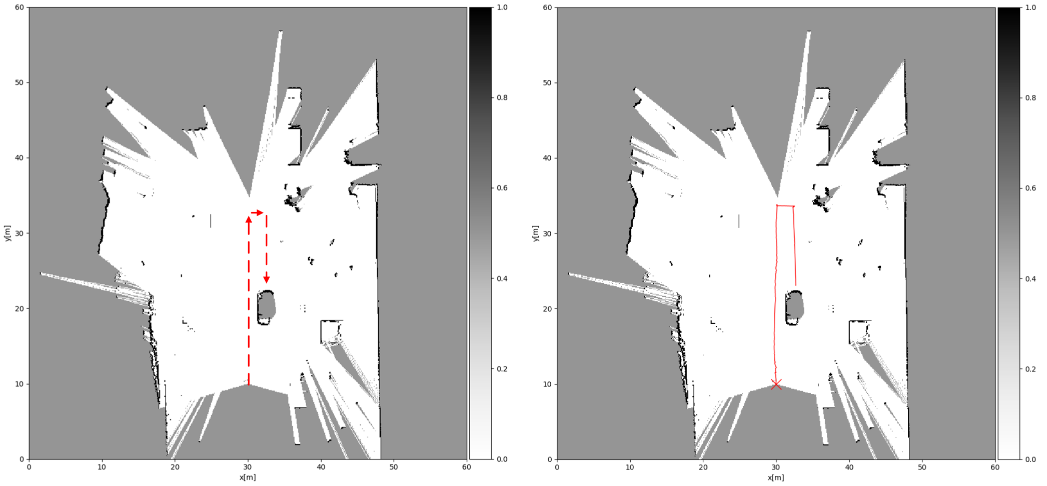


Figure 5.9.: Single car scenario reference grid. Left: Illustrated path. Right: SLAM positions.

correctly mapped. The cars further to the right of the starting point are not really mapped at all. To compensate for the wrong path again the positional data provided by the odometry has been used. It can be seen at figure 5.13. The estimated path clearly deviates from the expectation. Considering the nature of this scenario (big metallic reflective surfaces) the choice as one of the two scenarios for which the parameters are optimized is still valid. Because of the sensors properties, the resulting radar-generated grid will just be skewed around the corrupted path.

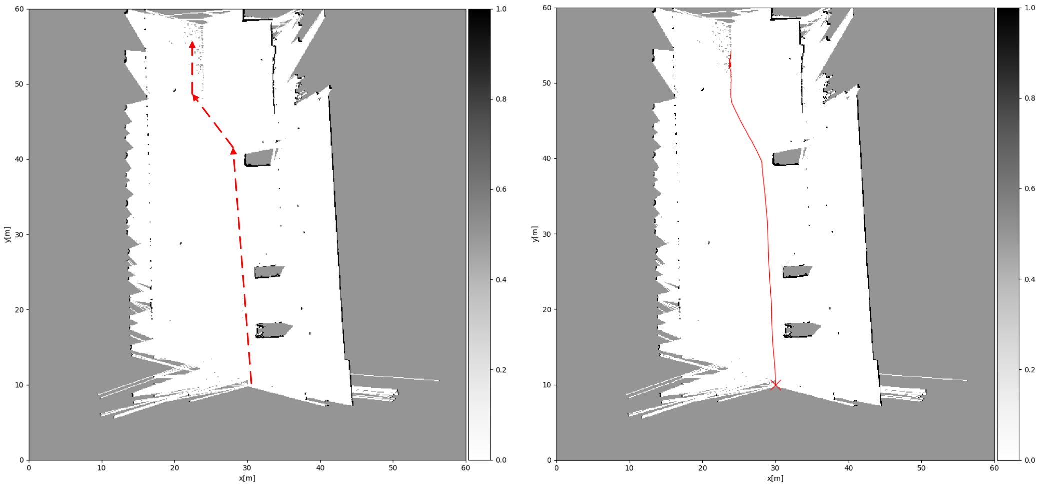


Figure 5.10.: Street scenario reference grid. Left: Illustrated path. Right: SLAM positions.

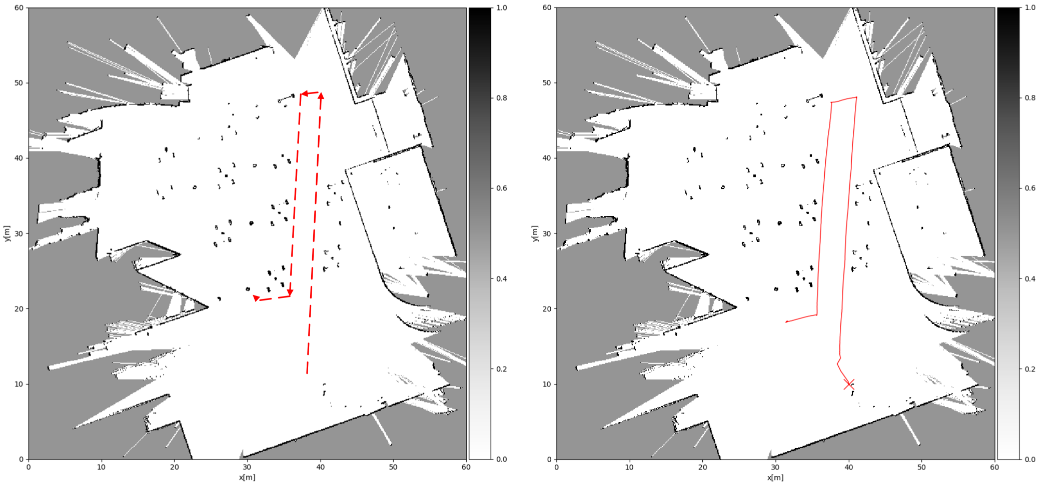
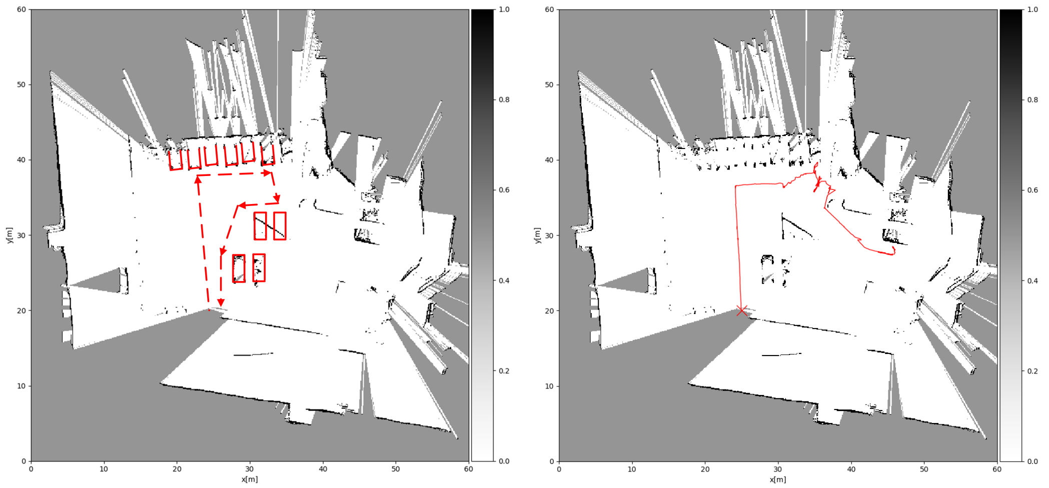


Figure 5.11.: Campus scenario reference grid. Left: Illustrated path. Right: SLAM positions.



(a)

(b)

Figure 5.12.: Parking lot scenario reference grid. (a) Illustrated path. Because the mapping algorithm failed at some point, the positions of the parked cars are also illustrated in order to have a ground truth reference for this scenario. (b) SLAM positions.

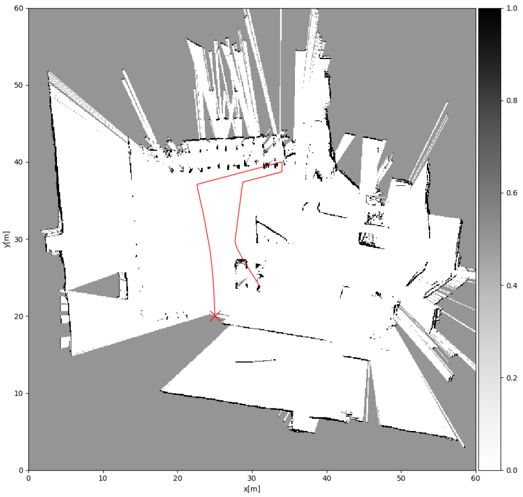


Figure 5.13.: Parking lot scenario reference grid. The map is created by the hector-map package, but the positions provided by the robot's odometry are depicted.

5.5. Metric

The radar-generated maps have been assessed according to criteria which aim to describe the quality of the gridmap as well as the type of deviation from the desired output. The assessments have been done subjectively by the author and therefore leave room for interpretation and further discussion. However this process gives a good overview of the evaluated data and the overall performance of the used methods.

For the final evaluation a total of approximately 2200 gridmaps have been created for the parametric sweeps (6 modeling approaches x 121 combinations of k_{FREE} and k_{OCC} x 3 values for α_{RA}) and further approximately 200 gridmaps (34 scenarios x 6 sets of final parameters) have been created by applying the optimized parameters to all scenarios. Because of this huge number of plots just a few examples will be added to the thesis for illustration and explanation purposes. The whole set of plots can be requested from Paul Meissner. As reference for the assessments the laser-generated grids (especially for the real-world scenarios), knowledge about the setup (especially for the regulated scenarios) and general knowledge of the scene from the given photos and the camera-stream will be used.

One big part of the map's quality will be determined according to the presence of the desired features. For the regulated scenarios this are the placed objects. For the street, single car and parking lot scenario this features are the cars. For the campus scenario this are the benches, trees and the surrounding walls. The second big part of the map's quality is the mapping of free space. In total there are five criteria for the final plots:

- **Over-Estimated Occupancy/Objects (OO):** Features are mapped too excessively. Objects are represented too big in terms of occupied cells. Also cells in the vicinity of an object can be occupied as a result of fragments (due to processing, sensor characteristics etc.).
- **Under-Estimated Occupancy/Objects (UO):** Features are mapped insufficiently. Known objects are represented too small or not mapped at all. The corresponding grid-cells are marked as free space.
- **Over-Estimated Free Space (OF):** Cells that are mapped as free space should be marked as unknown. The corresponding cells are highly

unlikely reached by the electromagnetic waves of the sensor (e.g. inside cars).

- **Under-Estimated Free Space (UF):** Cells that should be mapped as free space are mapped as unknown. This criteria is particularly evaluated in the vicinity of the driven path as it has not been occupied for sure.
- **Noise (N):** The map has random cells which are classified as occupied.

If a criteria is present in a certain scenario it has been rated according to the occurrence's severeness. The final categorized scenarios for all combinations can be found in Appendix A.

5.6. Results

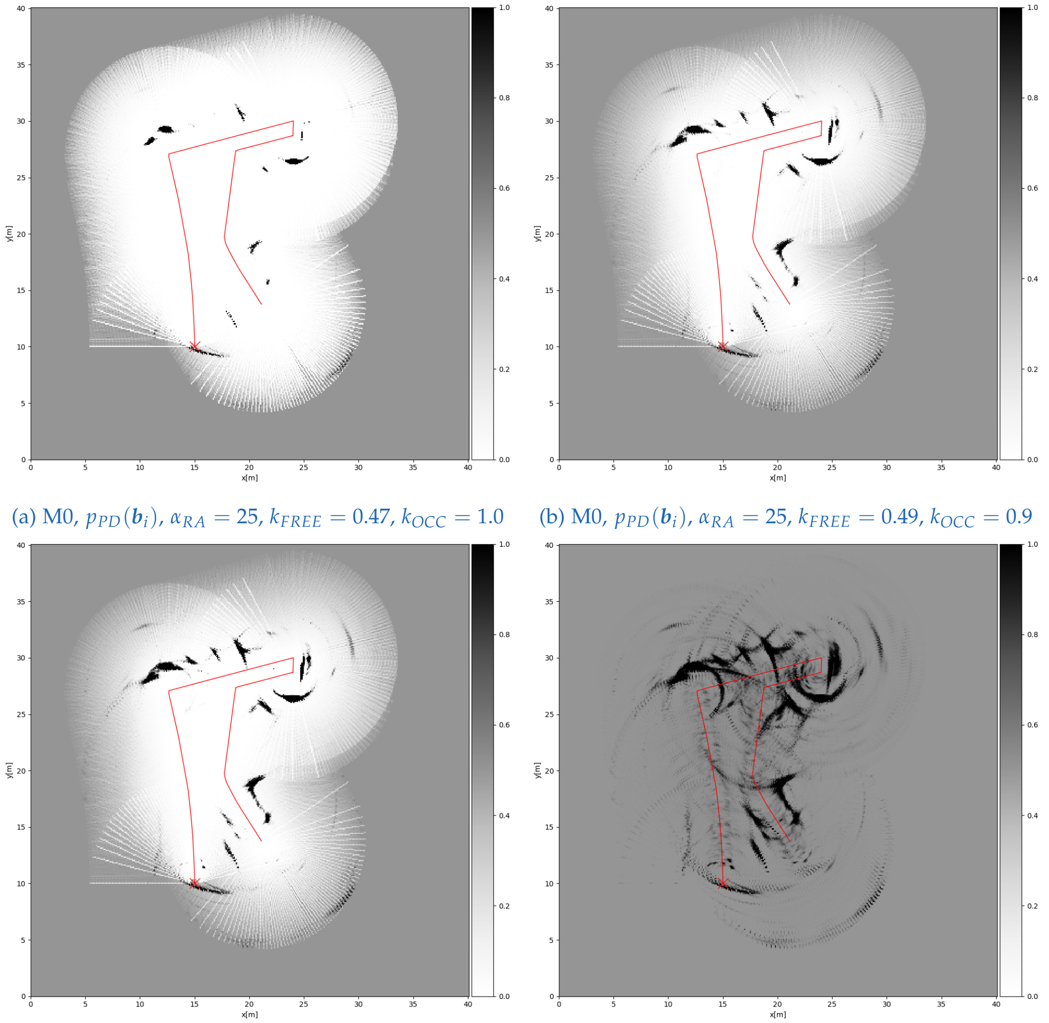
5.6.1. Influence of α_{RA}, k_{FREE} and k_{OCC}

To get a better understanding how the used parameters in Table 5.7 shape the resulting gridmap their influence will be discussed in this Section. The necessary parameter choice reflects back on the properties of $p_X(\boldsymbol{b}_i)$ as well as the properties of the model. For illustration purposes in Figure 5.14 and 5.15 only grid-maps of model 0 using $p_{PD}(\boldsymbol{b}_i)$ will be used. This is because $p_{PD}(\boldsymbol{b}_i)$ provided slightly better results than $p_{SNR}(\boldsymbol{b}_i)$ and maps created using model 0 contain the most information. However the principles apply to all other cases as well. Effects that occur on the shown maps will be discussed later.

Figure 5.14 first compares the influence of k_{FREE} and k_{OCC} for the same α_{RA}. Figure 5.14c shows the map for the optimized parameter set. As k_{FREE} tunes the weight for free space, it also tunes the compensation of noise and previously made detections. Consequently a higher value for k_{FREE} causes objects to be smaller or even to completely vanish. The resulting gridmap is sensitive to the choice of this parameter. A change of 0.02 already yields completely different results. This is illustrated in Figure 5.14a. On the contrary Figure 5.14d shows a map using $k_{FREE} = 0.5$. This means that noise is generally treated as non-informative. Naturally this map has no free space. All SNR values are put into the map and never eradicated since $p_X(\boldsymbol{b}_i)$ never goes below 0.5. Noise or object fragments are not removed as well. The choice of k_{OCC} is less important. In Figure 5.14b the value deviates by 0.1 to the optimized case and almost no difference is visible. Occupancy is emphasized a little more. However increasing k_{OCC} has a similar effect than increasing k_{FREE}.

Reducing α_{RA} from the optimized case as shown in Figure 5.15b, shows a similar effect like increasing k_{FREE}. The threshold for considering a SNR value as detection gets lower and therefore more values are considered a detection. Consequently a lower k_{FREE} with a simultaneously lower α_{RA} yields a similar result as the optimized case (Figure 5.15a). The difference between those two cases can be seen at the borders, where for the reduced k_{FREE} the space is weighted more as free space. Increasing α_{RA} has the opposite effect. Objects are smaller or even vanish (Figure 5.15c). Figure 5.15d shows that even for higher values of α_{RA} objects are detected but over-compensated by the free space estimation.

The change of the parameters has the same effects for $p_{PD}(\boldsymbol{b}_i)$ and $p_{SNR}(\boldsymbol{b}_i)$. However Figure 5.16 shows that the parameter choice for $p_{SNR}(\boldsymbol{b}_i)$ is much more delicate, especially for k_{OCC}. Figure 5.16b shows the optimized case. Just slightly increasing k_{OCC} by 0.05 causes the map to show way too much occupied areas and noise (Figure 5.16c). Slightly decreasing it by 0.05 shows a map that has too much free space (Figure 5.16a). Another important difference between $p_{PD}(\boldsymbol{b}_i)$ and $p_{SNR}(\boldsymbol{b}_i)$ is, that $p_{SNR}(\boldsymbol{b}_i)$ has several points with similar results over the whole parameter range (Equations 4.8 and 4.7). A comparison of the Figures 5.16a and 5.16d shows an example for this. Here k_{FREE} is increased with the same value k_{OCC} is decreased. For $p_{PD}(\boldsymbol{b}_i)$ no such behavior can be observed. Values of $k_{FREE} < 0.4$ cause the map to show almost no objects like shown in Figure 5.17.



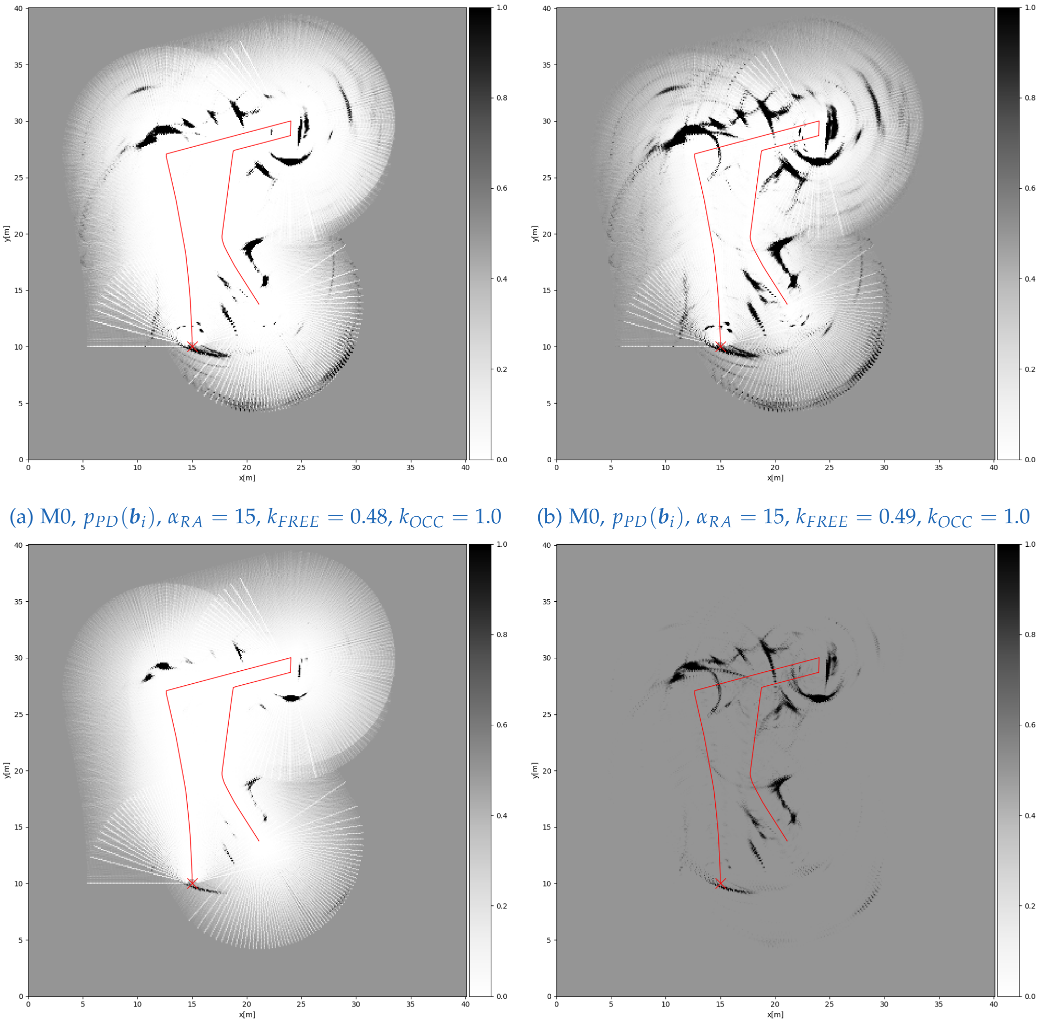
(a) M0, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 25$, $k_{FREE} = 0.47$, $k_{OCC} = 1.0$

(b) M0, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 25$, $k_{FREE} = 0.49$, $k_{OCC} = 0.9$

(c) M0, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 25$, $k_{FREE} = 0.49$, $k_{OCC} = 1.0$

(d) M0, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 25$, $k_{FREE} = 0.5$, $k_{OCC} = 1.0$

Figure 5.14.: Parking lot scenario (Ground truth: Figure 5.12a). Parameter influence for $p_{PD}(\boldsymbol{b}_i)$ and model 0 for a single $\alpha_{RA} = 25$. Decreasing k_{FREE} causes objects to be too small, increasing it yields a higher amount of unwanted fragments. Tuning k_{OCC} has the opposite effects of tuning k_{FREE}.



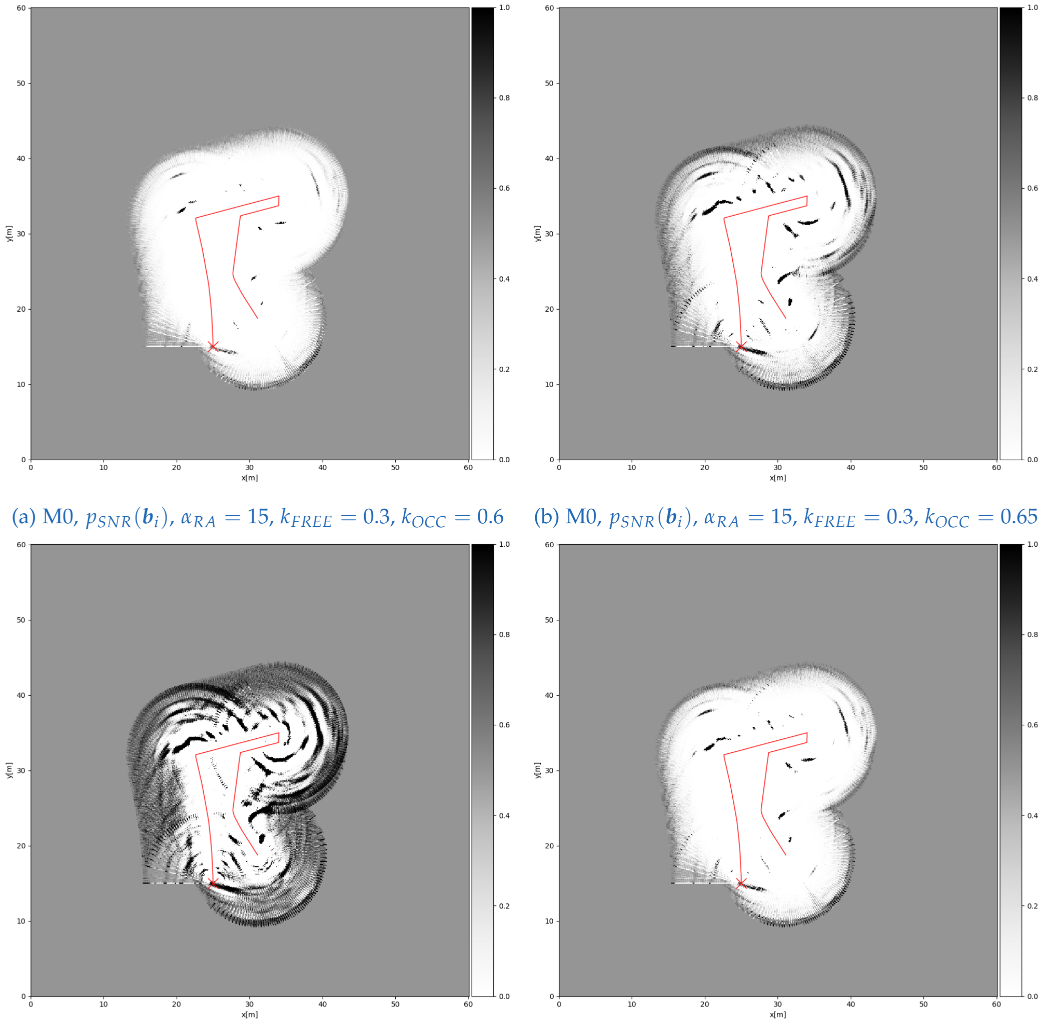
(a) M0, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.48$, $k_{OCC} = 1.0$

(b) M0, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.49$, $k_{OCC} = 1.0$

(c) M0, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 50$, $k_{FREE} = 0.49$, $k_{OCC} = 1.0$

(d) M0, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 50$, $k_{FREE} = 0.5$, $k_{OCC} = 1.0$

Figure 5.15.: Parking lot scenario (Ground truth: Figure 5.12a. Parameter influence for $p_{PD}(\boldsymbol{b}_i)$ and model 0 for different α_{RA}. Decreasing α_{RA} has the same effect as decreasing k_{FREE} and vice versa.



(a) M0, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.3$, $k_{OCC} = 0.6$

(b) M0, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.3$, $k_{OCC} = 0.65$

(c) M0, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.3$, $k_{OCC} = 0.7$

(d) M0, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.25$, $k_{OCC} = 0.65$

Figure 5.16.: Parking lot scenario (Ground truth: Figure 5.12a). Parameter influence for $p_{SNR}(\boldsymbol{b}_i)$ and model 0. Parameter changes cause the same effects as for $p_{PD}(\boldsymbol{b}_i)$, with the difference that $p_{SNR}(\boldsymbol{b}_i)$ is more sensitive to the parameters.

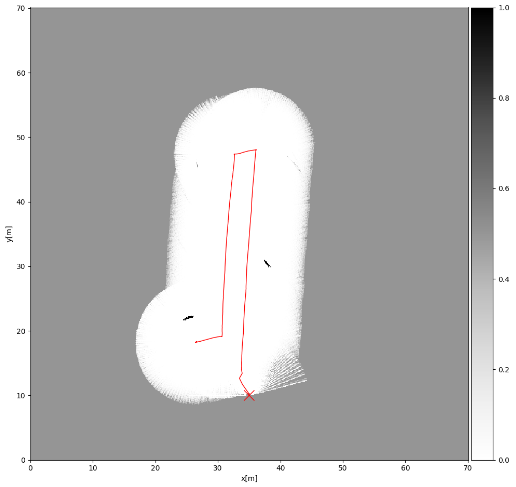


Figure 5.17.: Campus scenario (Ground truth: Figure 5.11). M0, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 25$, $k_{FREE} = 0.4$, $k_{OCC} = 1.0$. For $p_{PD}(\boldsymbol{b}_i)$ values for $k_{FREE} < 0.45$ yield maps with too much free space.

5.6.2. Detailed Assessment

This section will present a detailed assessment of the proposed sensor models using criteria given in Section 5.5. Detailed results from all conducted experiments can be found in the Appendix A.

Over-Estimated Occupancy/Objects

The effect that corrupts the radar-generated occupancy grids most severe is that some objects are estimated way too big. It the effect that is most difficult to handle. This effect occurs for objects that are highly reflective and large like the used metal box. Figures 5.21 to 5.23 show that this effect is very strong for all models no matter what choices of $p(\boldsymbol{b}_i)$ are made. Objects with smaller RCS don't have this kind of problem as shown in Figure 5.19.

Over-estimation has multiple causes. The first major cause is the choice of α_{RA}. As already discussed it is impossible to use a value that fits for all kinds of materials and object sizes equally well. The chosen thresholds are set such that the average detection performance is maximized. This however means that for objects with a high RCS α_{RA} is set too low. The corresponding projection in the range angle map appears too large. An example is pictured in Figure 5.20. It shows that due to the peak's high amplitude, the side lobe's amplitudes exceed the CFAR-threshold. Therefore a detection is triggered along the whole angle for the affected ranges. This phenomenon is the reason why a lot of maps show concentric circles around some objects. Figure 5.20a shows that this effect already starts at bigger distances, and becomes worse the further the sensor moves to the object. One solution for this problem would be choosing a higher value for α_{RA}. But as already shown in Section 5.6.1 this would require a way more detailed examination of the parameter space for k_{FREE} and probably also for k_{OCC}. Another way to address this problem would be to increase the angular resolution. This could be done by sub-space angle estimation methods like mentioned in Section 3.1.6 or by using a sensor that provides a better angular resolution.

A special case of this criteria occurs for strong reflecting objects that are close to $\pm 90^\circ$. The part of the peak that exceeds the maximum angle gets projected back to the other end of the axis as result of the FFT's circular

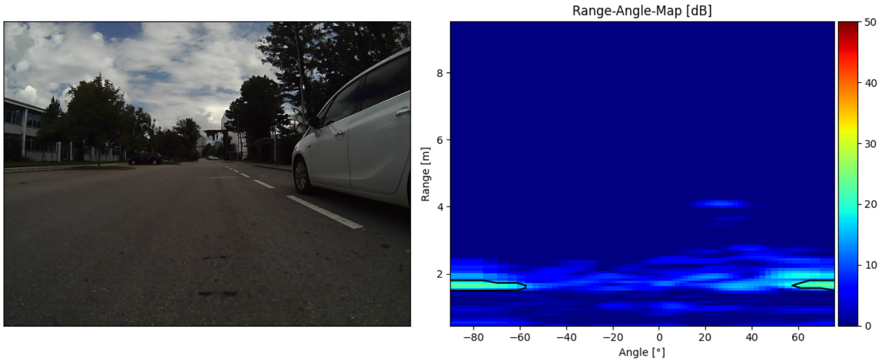


Figure 5.18.: Single car scenario. $\alpha_{RA} = 25$. The detection on the left side gets projected back to the right side.

convolution. This happens for example when driving next to a parking car like shown in Figure 5.18. Fragments opposite of the detected surface can be observed in the corresponding grid map (Figure 5.33a at $x \approx 17m$ and $x \approx 15m$). The only way this effect can be reduced, is by reducing over-estimation utilizing the measures described above.

The second major cause for over-estimated objects are multi-path reflections. They manifest as targets with different velocities and therefore can clearly be seen at at the range doppler maps of Figure 5.20. The further the sensor moves to the object the more clearly the effect occurs. At Figure 5.20a a single multi-path reflection already shows up at a distance of approximately $9m$. This additional target already appears as a detection on the range angle map and therefore corrupts the resulting grid map. At Figure 5.20b the effect becomes severely worse, and creates fragments behind the detected object, that are clearly visible in the final occupancy grid (Figure 5.22a at $x \approx 15m$ and $y \approx 20m$). The occupied space behind the metal box in Figures 5.21 to 5.23 is always a result of multi-path reflections. This can just be seen examining the range doppler maps as the space could also be occupied because of the second object behind the metal box.

The fact that the detections in the range angle map originate from objects in the range doppler domain with different velocities clearly shows that this detections are not because of the wood box behind the metal box. The wood box would be detected at the same velocity as the metal box i.e. the

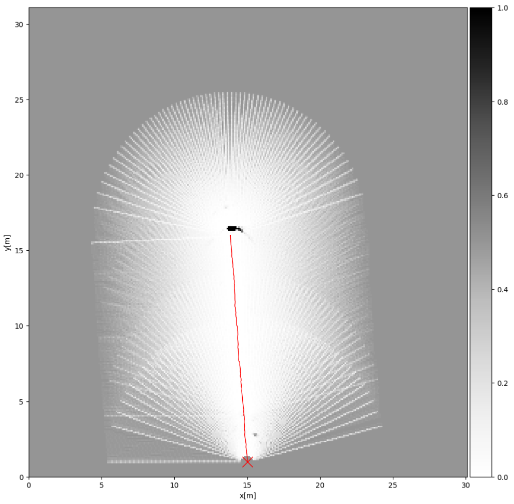
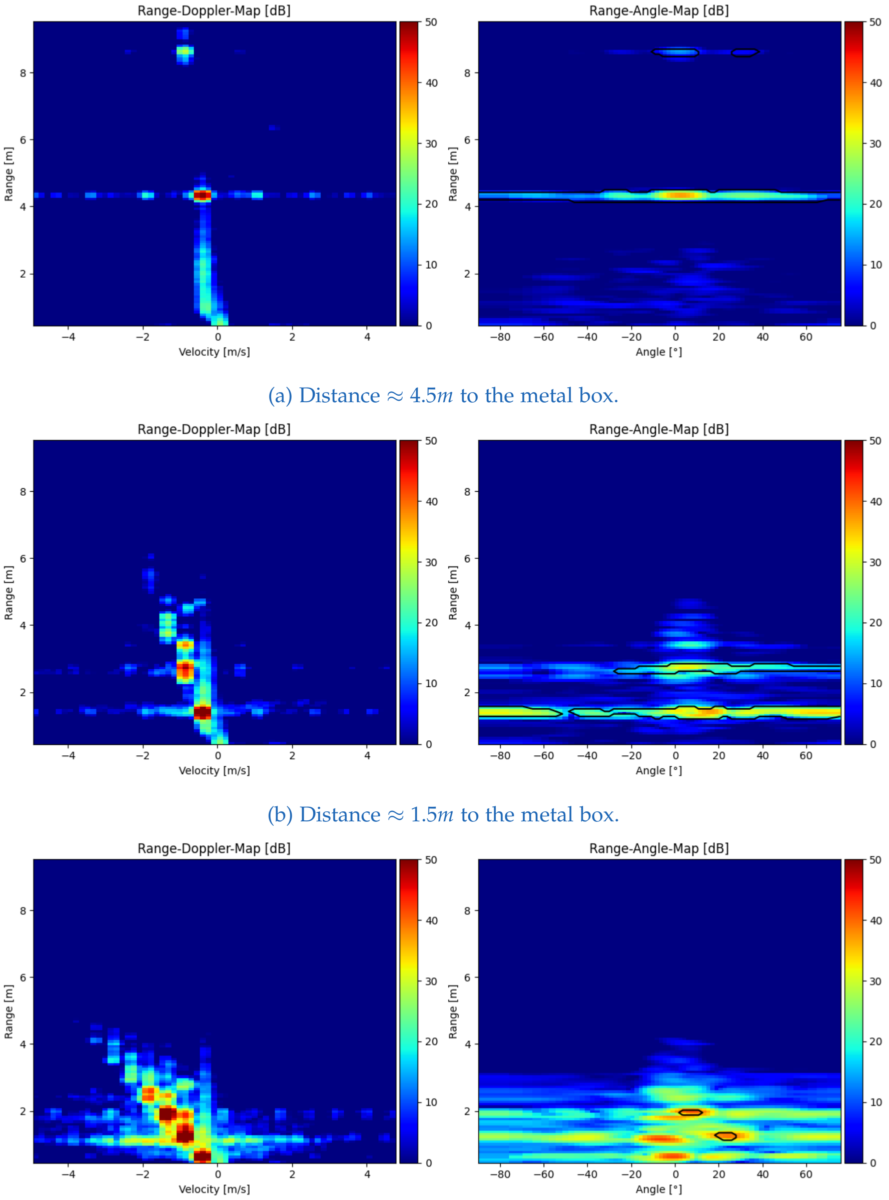


Figure 5.19.: Plastic pole om scenario (Nr. 4 in Table 5.3. Ground truth: Figure 5.4a). M0, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 25$, $k_{FREE} = 0.49$, $k_{OCC} = 1.0$. The object is mapped adequately because α_{RA} fits it's RCS.

platform's velocity. The over-estimation as a result of multi-path reflections occurs more intensively for strong reflecting objects. The reflected energy is higher and therefore more likely to traverse back by multiple paths. To compensate this effect just velocities of the range doppler map around the vehicles velocity could be used to generate the range angle map. This however would just remove the multi-pah reflections if the vehicle is moving. It also would probably lead to a worse detection of objects that are not directly in front of the sensor as their radial velocity changes when the sensor moves towards them.

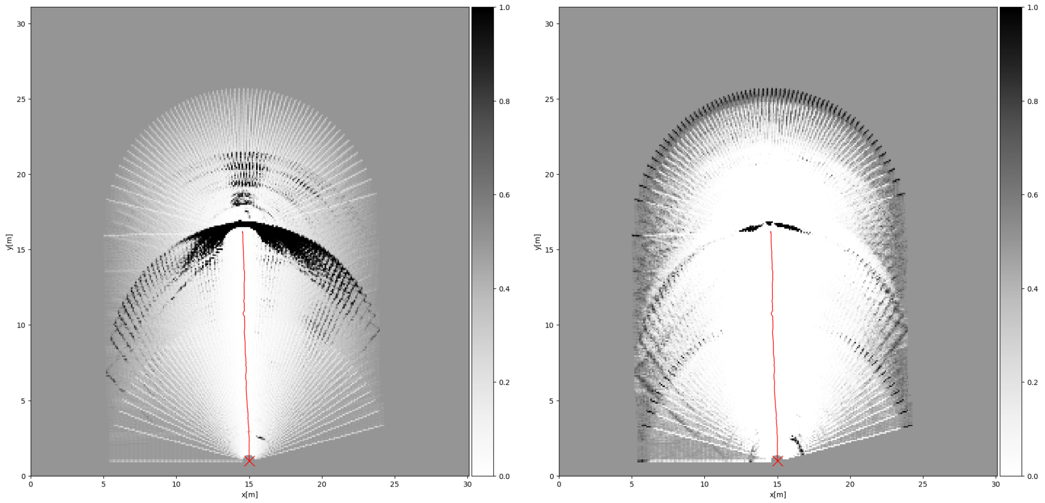


(a) Distance ≈ $4.5m$ to the metal box.

(b) Distance ≈ $1.5m$ to the metal box.

(c) Distance ≈ $0.75m$ to the metal box.

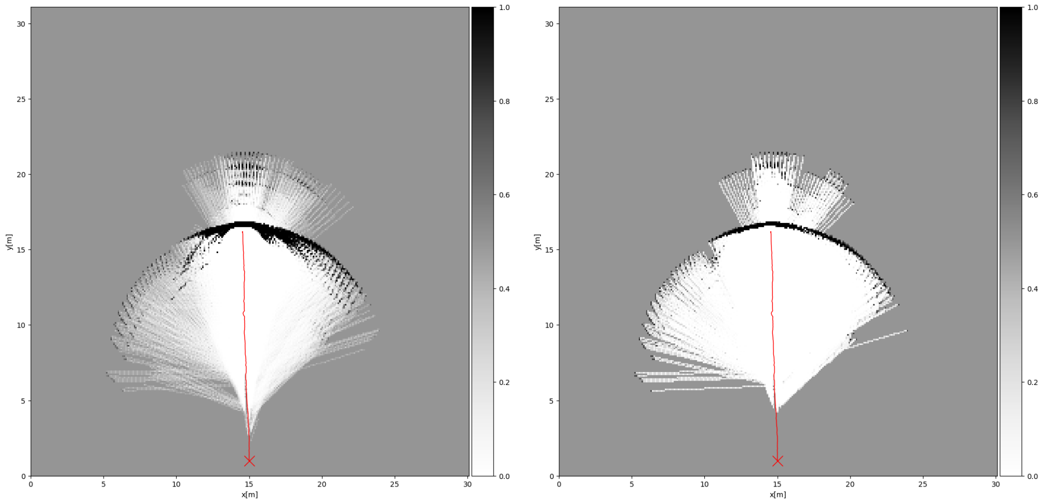
Figure 5.20.: Range doppler- and range angle maps for the Metal/Wood $2m$ scenario (Nr. 3 in Table 5.4, $\alpha_{RA} = 25$). The different objects in the range angle map are a result of multi-path reflections which can bee seen in the corresponding range doppler map.



(a) M0, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 25$, $k_{FREE} = 0.49$, $k_{OCC} = 1.0$

(b) M0, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.3$, $k_{OCC} = 0.65$

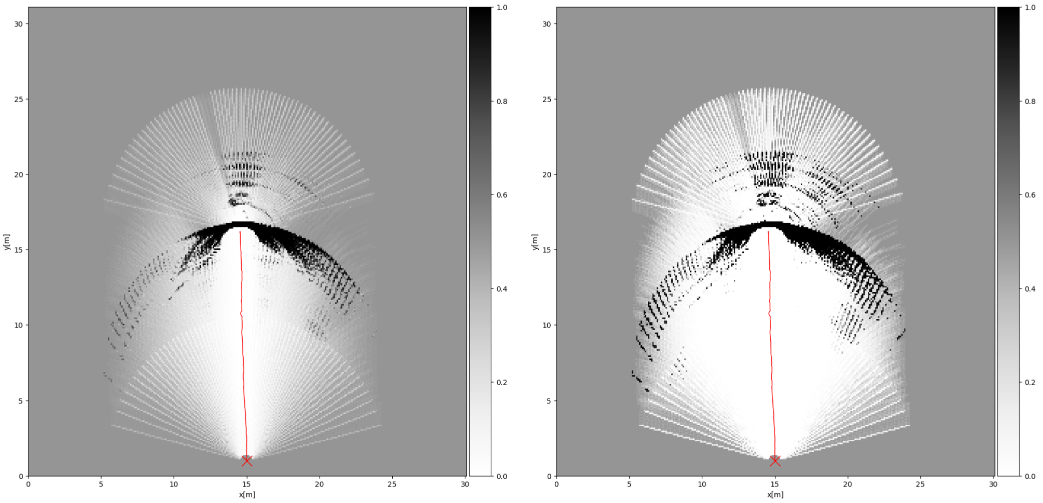
Figure 5.21.: Example of an over-estimated occupancy case for model 0. Metal box/Wood box $2m$ scenario (Nr. 3 in Table 5.4. Ground truth: Figure 5.4b). The metal box is mapped far too large because of it's big RCS.



(a) M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

(b) M1, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 25$, $k_{FREE} = 0.25$, $k_{OCC} = 0.8$

Figure 5.22.: Example of an over-estimated occupancy case for model 1. Metal box/Wood box $2m$ scenario (Nr. 3 in Table 5.4. Ground truth: Figure 5.4b). The metal box is mapped far too large because of it's big RCS.



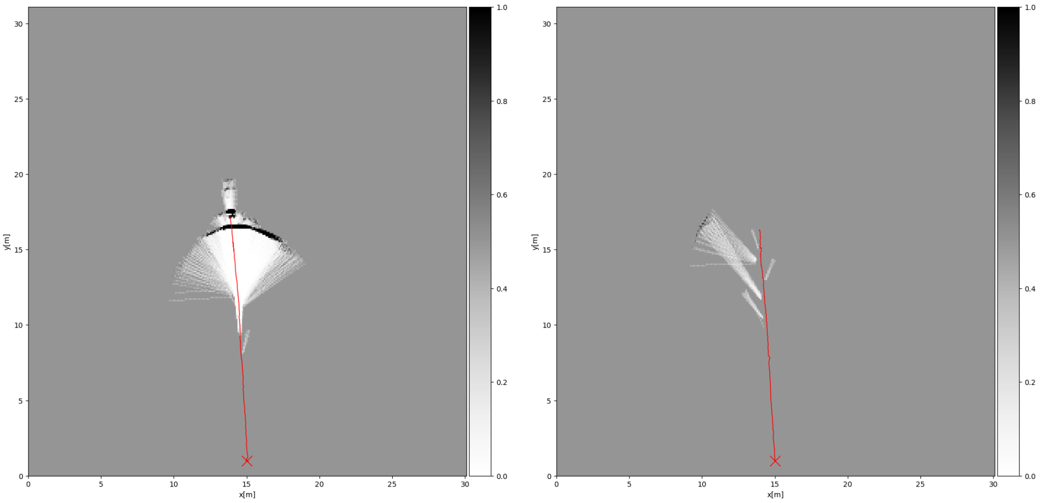
(a) M2, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.49$, $k_{OCC} = 1.0$ (b) M2, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

Figure 5.23.: Example of an over-estimated occupancy case for model 2. Metal/Wood 2m scenario (Nr. 3 in Table 5.4. Ground truth: Figure 5.4b). The metal box is mapped far too large because of it's big RCS.

Under-Estimated Occupancy/Objects

One cause for the absence of objects has already been shown and explained in Section 5.6.1 and can be seen in Figures 5.15c and 5.14a: α_{RA} is set too high or k_{FREE} is set too low respectively. As the parameter choice strongly depends on the reflectivity and therefore the RCS of objects, a RCS that is too small is often the root cause for missing or too small mapped objects. To investigate this effect the resulting grids of the three small objects are shown in Figures 5.24 to 5.26. Model 1 is used for the illustration because the compensation of occupancy by free space is minimal. Also the free space in front of an object shows the distance from which on an object is detected. The metal pole in Figure 5.24a is clearly recognized and even over-estimated when approached straight. When approaching it 4m shifted in Figure 5.24b, the pole is still visible. The detection is not that strong though but still an over-estimation effect can be seen. The wood pole yields a contrary picture. When approached straight in Figure 5.26a it is recognized with little to no over-estimation. Approached by a shifted distance of 2m (Figure 5.26b) the pole isn't recognized at all. This is because wood is a poor reflector of electromagnetic waves and therefore has a significantly smaller RCS than the metal pole - even if they have approximately the same size and shape. The plastic pole shows that its RCS lies between the metal and the wood pole. It is detected for the 0m (Figure 5.25a) and 2m (Figure 5.25b) case, but in case of approaching it shifted by 4m it is not visible at all.

Another cause for the absence of objects occurs for the two object scenarios. For some scenarios the sensor was clearly able to detect the 2nd object. As already shown and explained in Section 5.6.2, the sensor is not able to penetrate metal. The occupied space that is visible behind the metal box is just a result of multi-path fragments. For the other two cases - the plastic box and the wood box - the experiments showed that the sensor was able to detect the 2nd object in most cases. Figure 5.27 shows that the plastic box has been detected behind the wood box in the 2m case as well as in the 4m case, whereas the plastic box was just detected in the 2m case. The reason for this is because the surface of the wood box is significantly bigger and thicker and therefore harder to penetrate. The metal box behind the two other objects is always detected except for the Wood box/Metal box 4m scenario for the same reason (see Figures 5.29 and 5.30).



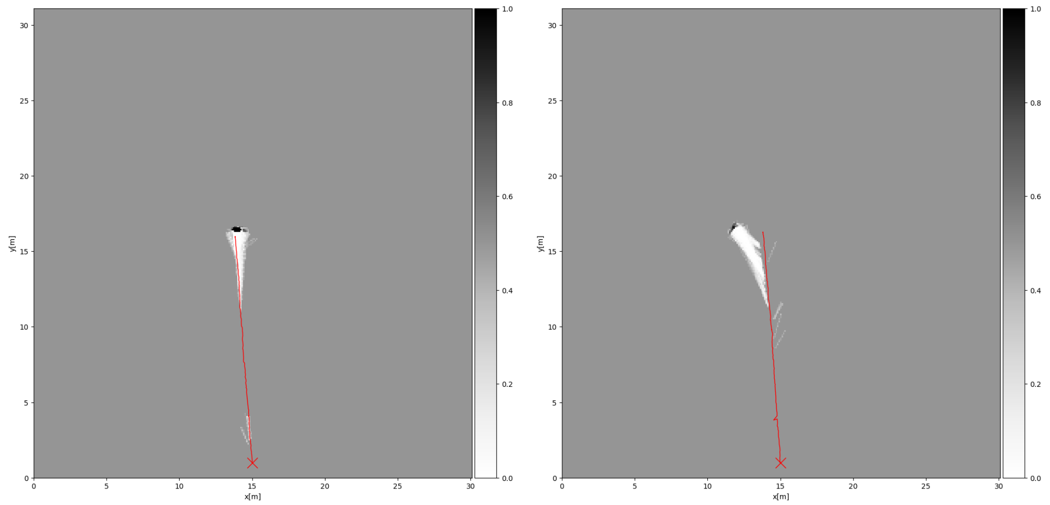
(a) Metal pole $0m$ scenario, M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

(b) Metal pole $4m$ scenario, M1, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

Figure 5.24.: Approached directly the metal pole is visible and clearly over-estimated. The occupied space that appears as a second object behind it is a result of multi-path reflections of the metal pole. When the pole is approached shifted by 4m it is still visible, but the representation is distorted. Ground truth: Figure 5.4a.

A special case for an under-estimated object can be seen at figure 5.31. When using $p_{PD}(\boldsymbol{b}_i)$ the first object is clearly visible (figure 5.30a) whereas when using $p_{SNR}(\boldsymbol{b}_i)$ it almost vanishes (figure 5.31a). This effect occurs for model 0 as well as model 1. For model 1 it is not that severe though. Looking at figure 4.8 shows that $p_{SNR}(\boldsymbol{b}_i)$ has a margin that is considered almost certainly as free up until approximately 1m. This margin results from the noise floor estimation of the CFAR. The high DC-noise components also visible in figure 5.6 cause the noise estimate to be too high. And even if the range dimension is cropped this high noise still influences the SNR estimate by the CFAR's lagging window. This problem can be solved by further cropping the usable range and therefore setting r_{CMIN} to a higher value e.g. 1m.

Interestingly a lot of objects are mapped too small for the approach using model 2. This can be seen for example by comparing the figures 5.32b and 5.34b. SNR measurements that are below the CFAR-threshold are rejected and therefore objects are less dense. Simultaneously occupied space is

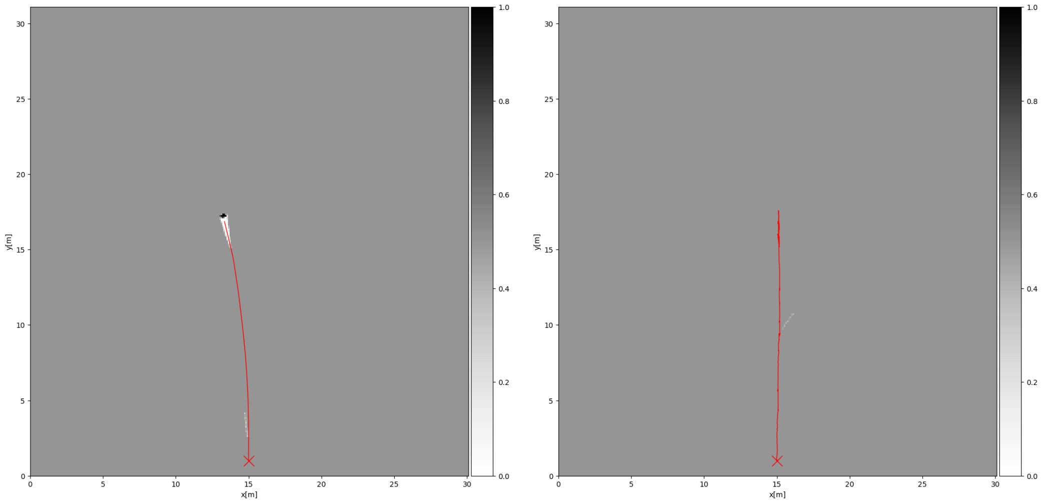


(a) Plastic pole $0m$ scenario, M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

(b) Plastic pole $2m$ scenario, M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

Figure 5.25.: Approached directly the plastic pole is clearly visible and a little over-estimated. When the pole is approached shifted by 2m it is still visible. When it is approached shifted by 4m it is not visible at all. Ground truth: Figure 5.4a.

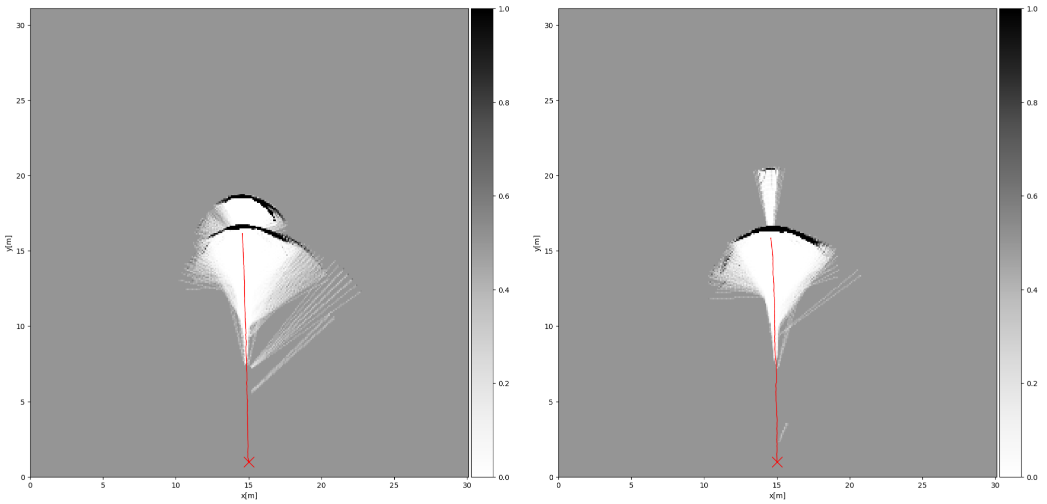
compensated by noise that is considered as free space according to equation 4.10. The sum of this two effects add up to the worse results.



(a) Wood pole $0m$ scenario, M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

(b) Wood pole $2m$ scenario, M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

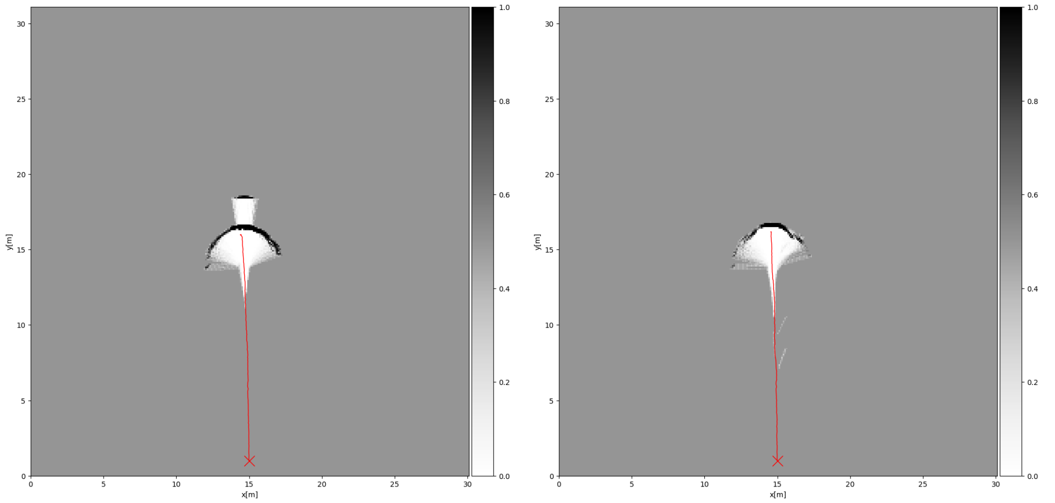
Figure 5.26.: Approached directly the wood pole is visible. When the pole is approached shifted by 2m it is not visible on the map at all. Ground truth: Figure 5.4a.



(a) Plastic box/Wood box $2m$ scenario, M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

(b) Plastic box/Wood box $4m$ scenario, M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

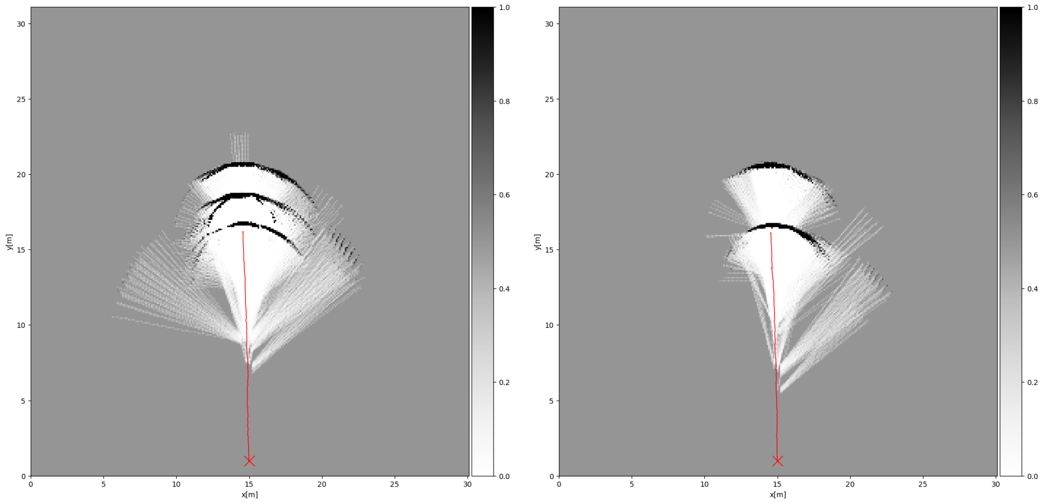
Figure 5.27.: The wood box is visible behind the plastic box for both offsets. Ground truth: Figure 5.4b.



(a) Wood box/Plastic box $2m$ scenario, M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

(b) Wood box/Plastic box $4m$ scenario, M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

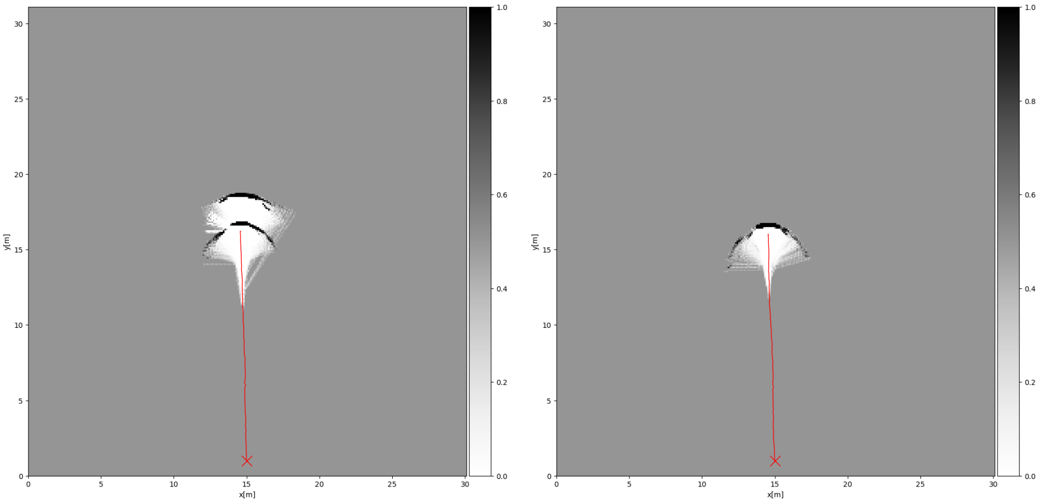
Figure 5.28.: The plastic box is just visible when it is placed 2m behind the wood box. Ground truth: Figure 5.4b.



(a) Plastic box/Metal box $2m$ scenario, M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

(b) Plastic box/Metal box $4m$ scenario, M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

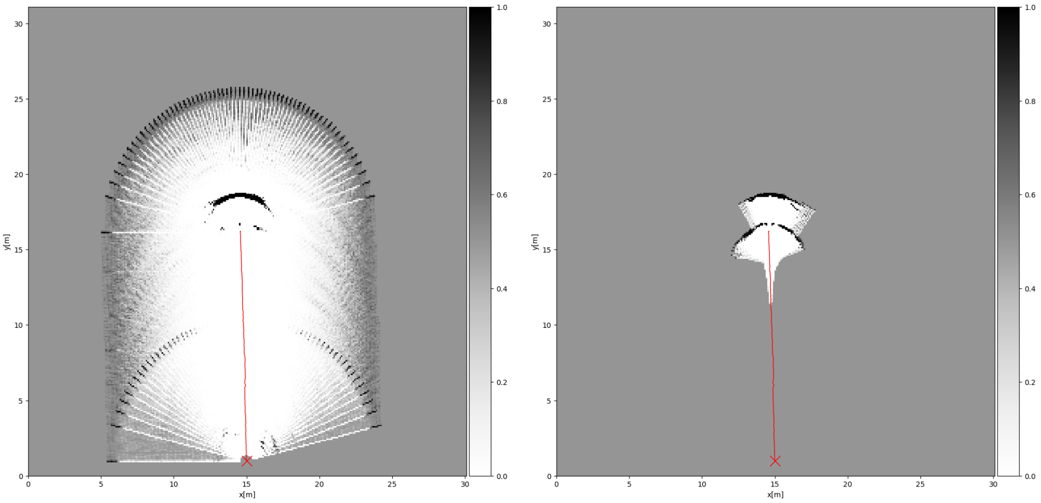
Figure 5.29.: The metal box is visible behind the plastic box for both 2m and 4m offset. The third object in the 2m scenario is due to multi-path reflections of the metal box. Ground truth: Figure 5.4b.



(a) Wood box/Metal box $2m$ scenario, M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

(b) Wood box/Metal box $4m$ scenario. M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

Figure 5.30.: The metal box is just visible behind the wood box with an offset of 2m but not with an offset of 4m. Ground truth: Figure 5.4b.



(a) M0, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.3$, $k_{OCC} = 0.65$

(b) M1, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.25$, $k_{OCC} = 0.8$

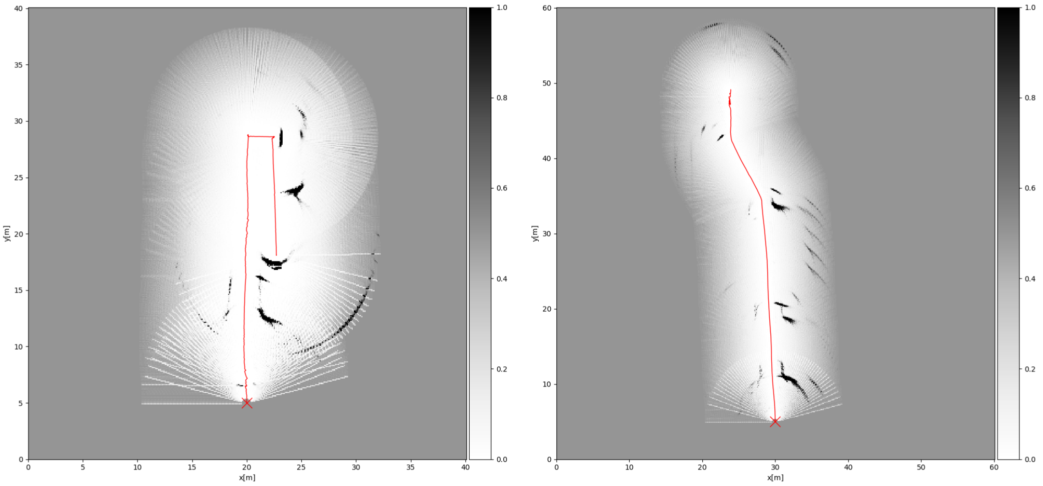
Figure 5.31.: Wood box/Metal box $2m$ scenario (Nr. 9 in Table 5.4). Parts of objects are missing due to the noise estimation of the CFAR algorithm. Ground truth: Figure 5.4b.

Over-Estimated Free Space

Determining which areas should be unknown for the given scenarios is generally difficult. As already shown, it is often possible that the sensor penetrates certain objects. In other cases the sensor might look beyond objects because they are not very high, the sensor has a beam-width of $\approx 10°$ in elevation and was mounted approximately 0.5m above the ground. Therefore for assessing over-estimation of free space the focus has been put on the scenarios with cars. The sensor cannot penetrate the metal surface of a car and therefore the inside should be marked as unknown. That means the single car, street and parking lot scenarios are important for this criteria. Figures 5.32 to 5.34 show the occupancy grids using $p_{PD}(\boldsymbol{b}_i)$ with all models for the single car and the street scenario. The parking lot grid maps can be found in Section 5.6.3.

The best results with respect to this criteria are achieved using model 1 as free space is mapped very conservatively. In most cases the contours of the cars are clearly visible especially in the parking lot scenario (Figure 5.37b). In some cases there is a slight over-estimation of the free space as a result of a over-estimation of the car's surfaces (e.g. Figure 5.33a at $x \approx 22m$ and $y \approx 15m$.

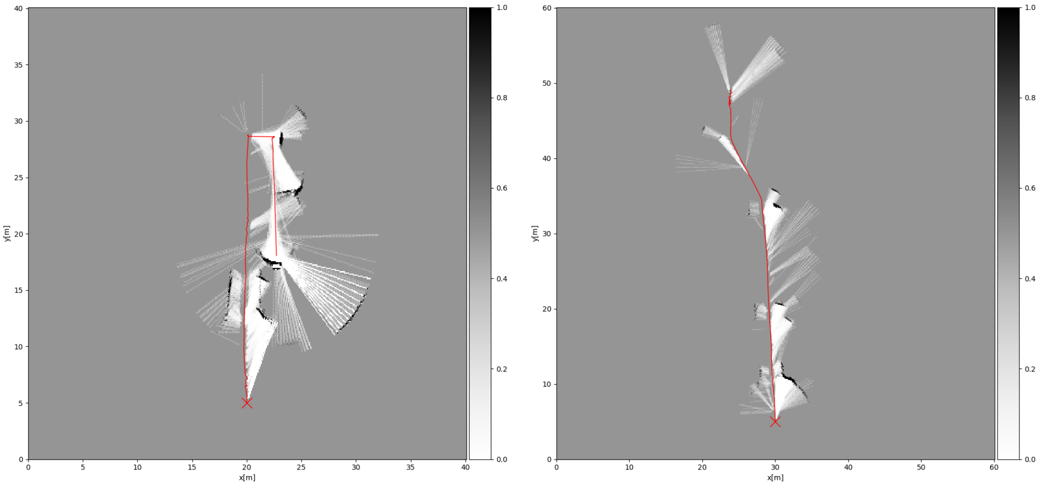
As expected the inside of the cars for model 0 are marked as free because no detection is generally considered as free space. Interestingly the results for model 2 deviate from the expectations. The modeling approach should have improved the free space over-estimation by considering noise towards the borders of the FOV less certain as free space. However free space is almost as over-estimated as for model 0. This is a strong indication that the used function for $f_{FREE}(\phi, k_{FREE})$ (Equation 4.10) is not increasing fast enough towards 0.5 as the angle goes towards $\pm 90°$. Also the parameter space for k_{FREE} could be explored more densely. This is an additional explanation for the potential under-estimation of occupancy for model 2 mentioned in Section 5.6.2.



(a) Single car scenario, M0, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 25$, $k_{FREE} = 0.49$, $k_{OCC} = 1.0$

(b) Street scenario, M0, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 25$, $k_{FREE} = 0.49$, $k_{OCC} = 1.0$

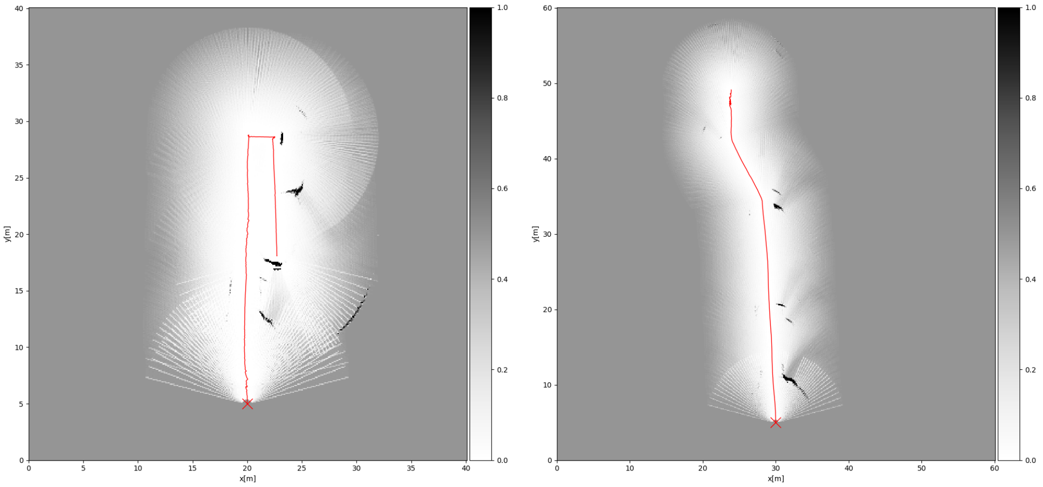
Figure 5.32.: Model 0 final occupancy grids (Ground truth: Figures 5.9 and 5.10). The space inside the cars for both scenarios should not be mapped as free space.



(a) M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

(b) M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

Figure 5.33.: Model 1 final occupancy grids (Ground truth: Figures 5.9 and 5.10). Unlike for model 0 and model 2 there is almost no over-estimation of free space inside the cars.



(a) M2, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.49$, $k_{OCC} = 1.0$

(b) M2, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.49$, $k_{OCC} = 1.0$

Figure 5.34.: Model 2 final occupancy grids (Ground truth: Figures 5.9 and 5.10). The space inside the cars for both scenarios should not be mapped as free space. However the effect is not as strong as for model 0.

Under-Estimated Free Space

Under estimation of free space happened only when model 1 was used. This is obvious as it reflects the shortcoming of this approach as discussed in Section 4.3.1. For areas including a lot of objects with a high RCS it still achieves decent results as can be seen at the parking lot scenario in Figure 5.37b. Still this model serves more as an analytical tool than as a final model. Figures 5.25 and 5.26 show examples where free space is not recognized, even along the robots path. In Figure 5.26b there is even no free space due to the lack of detections.

Noise

A lot of spots that are mistakenly classified as occupied and look like noise at the first place can be traced back to effects that are connected to the over-estimation of objects explained in Section 5.6.2. Therefore the noise criteria is in some cases closely related to this over-estimation. However there are some cases where the presence of noise fragments has other reasons.

One of them can be seen in Figures 5.36c and 5.36d. Both figures show noise at all borders of the measured space. This is related to the properties of $p_{SNR}(\boldsymbol{b}_i)$. Figure 5.35 helps to understand the cause of this phenomenon. Even for SNR values in the vicinity of the noise floor a probability of occupancy close to 1 is assigned. For example the slight peak in the range angle map at a range of $\approx 9m$ and an angle of $30°$ is assigned a probability of ≈ 0.9. The cause of this lies in the function of $p_{SNR}(\boldsymbol{b}_i)$ (Equation 4.4). Figure 4.11 already indicated this behavior. $p_{SNR}(\boldsymbol{b}_i)$ is already fully ascending for a value of 0dB while $p_{PD}(\boldsymbol{b}_i)$ starts to ascend at much higher SNR values (depending on α_{RA}).

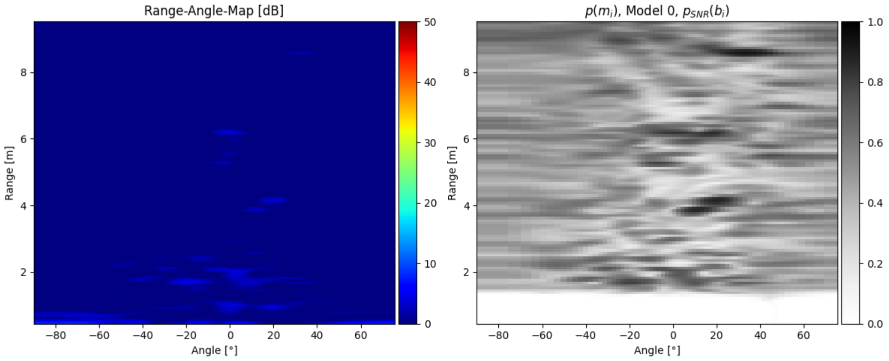


Figure 5.35.: Campus scenario. M0, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0$, $k_{OCC} = 1.0$. $p_{SNR}(\boldsymbol{b}_i)$ causes high probabilities of occupancy at the borders of the sensor's FOV.

5.6.3. Final Radar-Generated Occupancy Grids

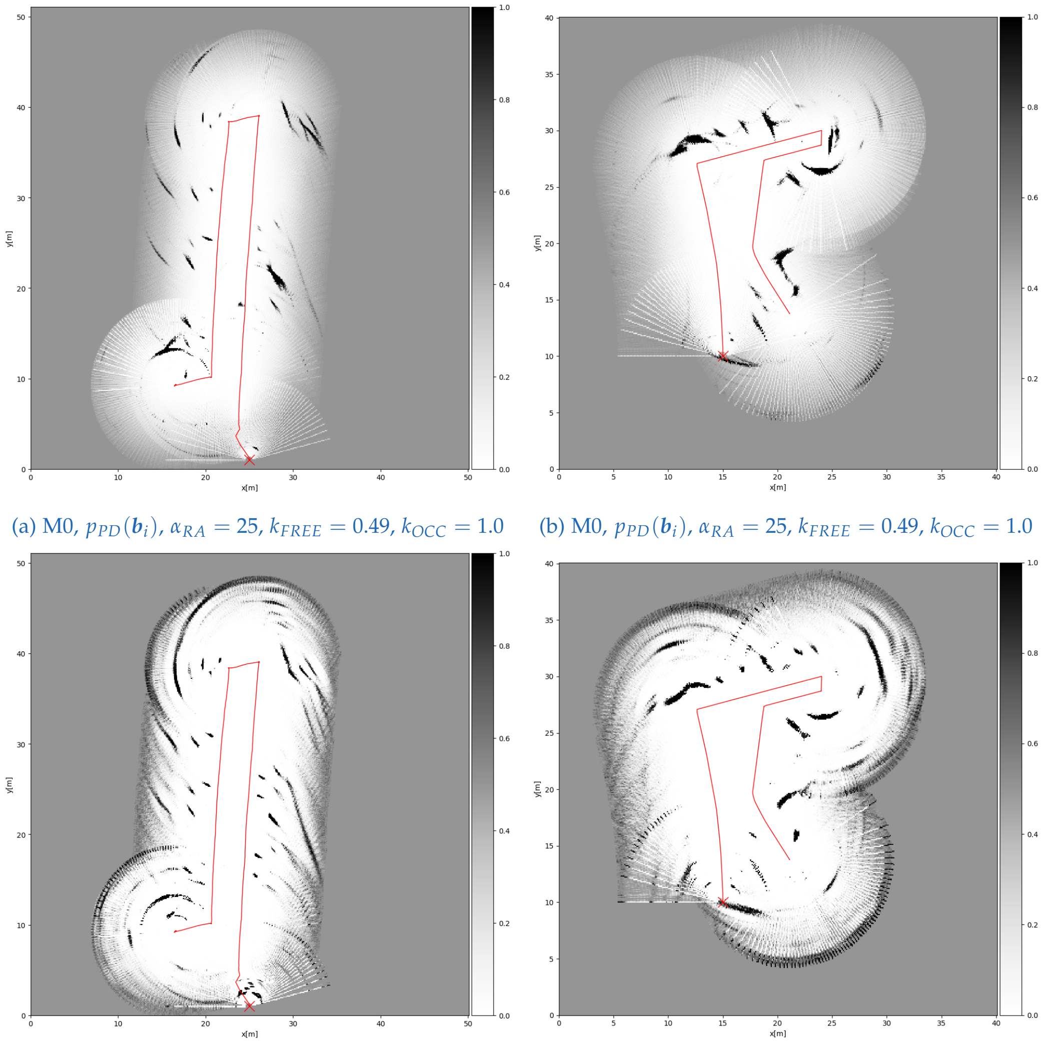
Figures 5.36 to 5.38 show the grids of the campus and the parking lot scenario with the optimized parameters according to Table 5.7. The important features in both scenarios are visible in principle (the benches, trees and walls in the campus scenario and the cars in the parking lot scenario). All figures show some of the criteria and the corresponding effects described above.

The campus scenario using model 0 in Figure 5.36 shows that the detection of the trees is rather difficult. Just a few of them are detected between the benches. The scene also includes two wastepaper baskets which have a fully metallic surface (one at $x \approx 15m$ and $y \approx 12m$, the second at $x \approx 27m$ and $y \approx 20m$). They are mapped too big and too dense because of the high reflective nature of the object as described in Section 5.6.2. The cars in the parking lot scenario using model 0 are in principal detected. However there is a huge gap between the detection of cars that are faced straight (e.g. before the first right turn in the path) and that are at the borders of the the sensor's FOV (e.g. after the first right turn to the left side of the path). This perfectly shows the strong dependence of the RCS on the angle as described in Section 5.6.2. Characteristically for model 0 is also the free space behind the detected surfaces (inside the cars), which is not desired and has been explained in Section 5.6.2. Figure 5.36b shows two fragments at $x \approx 17m$ and $y \approx 15m$.

The right one is a product of an object being projected over the edges of the angular dimension as described in Section 5.6.2. This can be concluded as the fragment is parallel to the parking car. The left one is probably a result of the odometry's position drift and therefore a measurement of the same parking car. With the choice of $k_{FREE} = 0.49$ for $p_{PD}(\boldsymbol{b}_i)$, noise is treated as "almost non-informative" and with $k_{OCC} = 1.0$ detections are rated reliable. The parameter choices for $p_{SNR}(\boldsymbol{b}_i)$ can't be explained that intuitively. The $p_{SNR}(\boldsymbol{b}_i)$ based approach shows characteristic higher noise and occupancy values as described in Section 5.6.2.

In a lot of real-world scenarios the approach of model 1 will not produce usable results because mapping of free space is neglected in a lot of cases. In this thesis it served more as an analytical tool because it just focuses on detections. Because the parking lot scenario has a lot of highly reflective objects, the results for this scenario are still usable. Section 5.6.2 already explained, that free space is mapped too sparse. Noise is interpreted more distinguished as free space with this modeling approach and therefore k_{FREE} can be set lower compared to model 0. However model 1 is interesting in another context as well. It somehow shows why using the object list for creating occupancy grids is difficult. The mapping of free space depends on the presence of detections. So model 1 and the object list have the same basic idea.

Model 2 shows the results as explained in Section 5.6.2. Unfortunately the results don't meet the expectations. Objects are not mapped as distinct as for model 0 and 1 but free space is still over-estimated. This indicates that this approach needs further investigation as it should be able to perform better than both approaches.



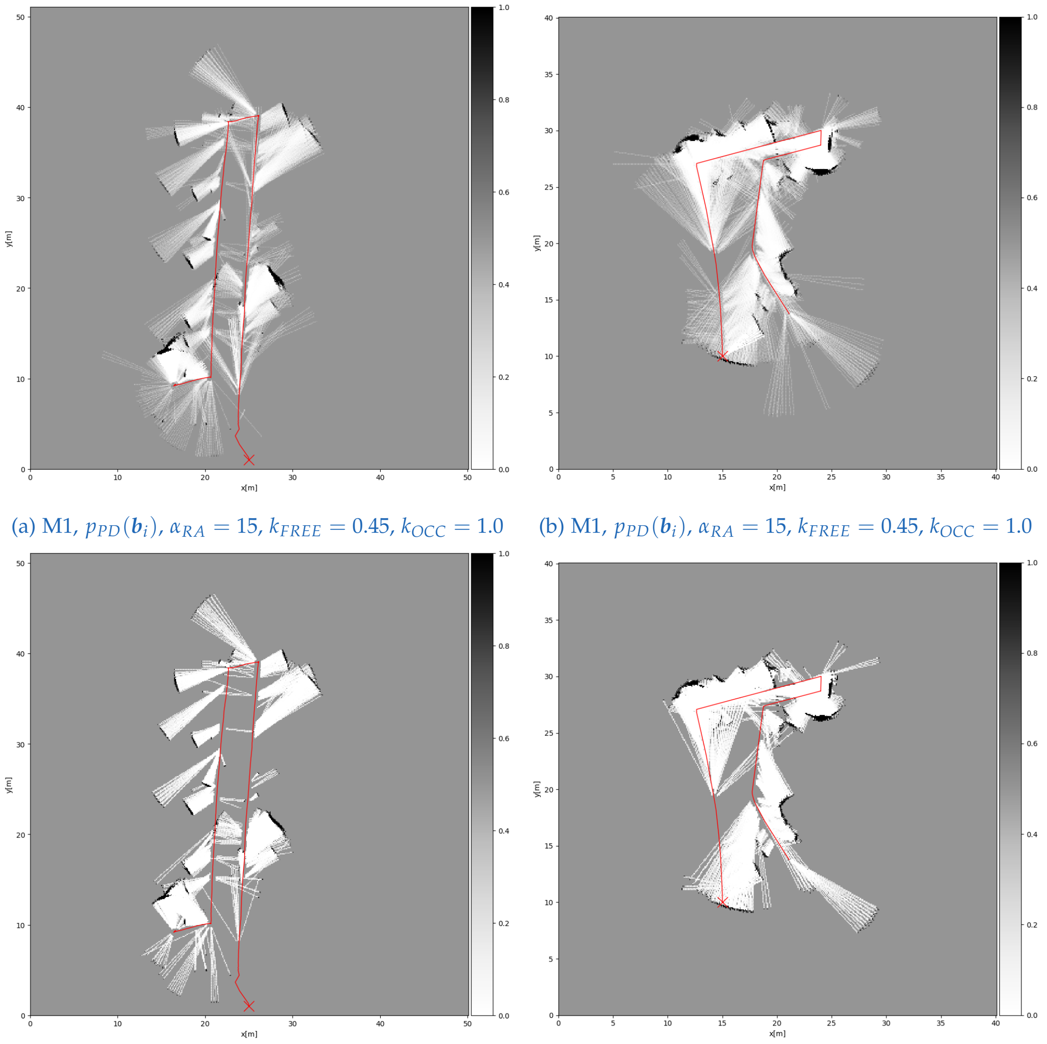
(a) M0, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 25$, $k_{FREE} = 0.49$, $k_{OCC} = 1.0$

(b) M0, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 25$, $k_{FREE} = 0.49$, $k_{OCC} = 1.0$

(c) M0, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.3$, $k_{OCC} = 0.65$

(d) M0, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.3$, $k_{OCC} = 0.65$

Figure 5.36.: Final radar generated occupancy grids for the campus and parking lot scenario using model 0. Figures (a), (c): Campus scenario (Ground truth: Figure 5.11). Figures (b), (d): Parking lot scenario (Ground truth: Figure 5.12a).



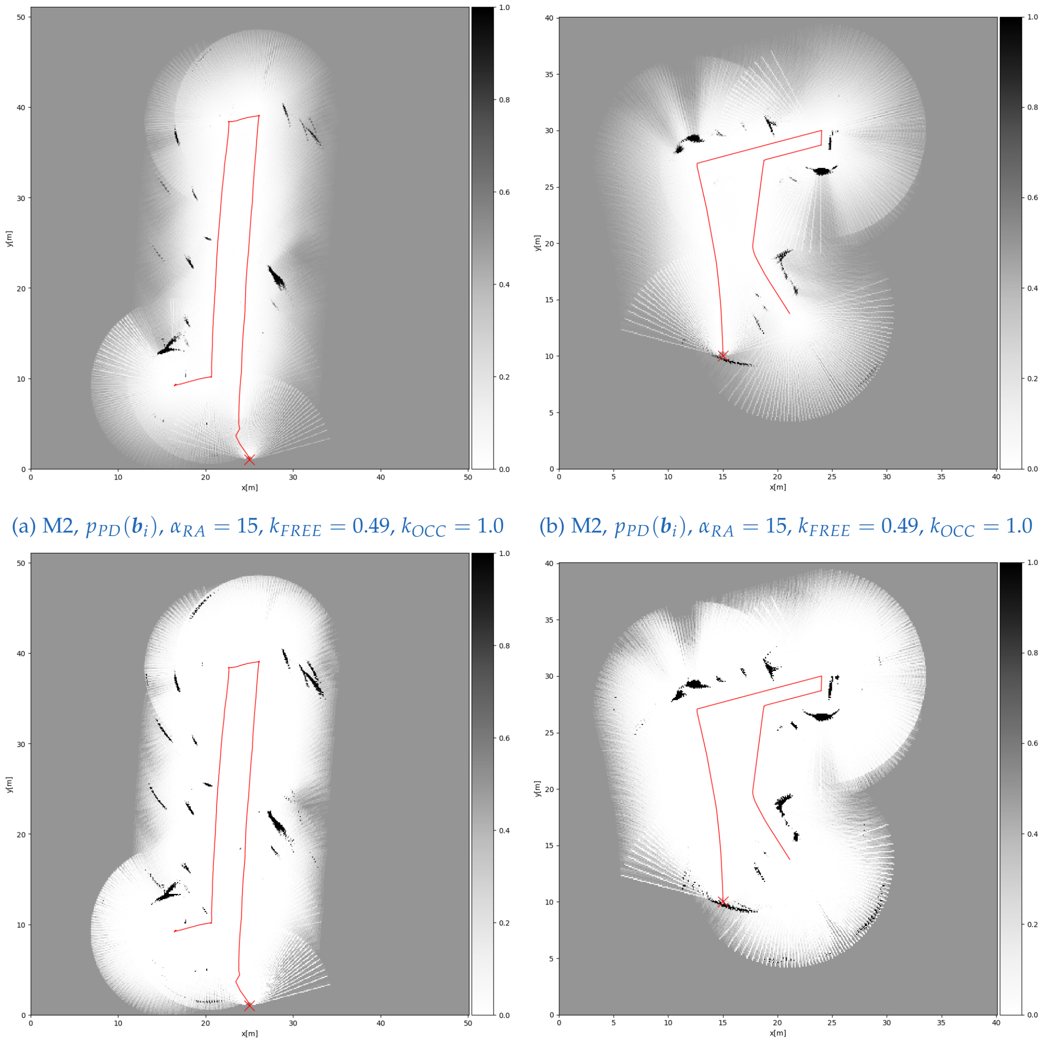
(a) M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

(b) M1, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1.0$

(c) M1, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.25$, $k_{OCC} = 0.8$

(d) M1, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 25$, $k_{FREE} = 0.25$, $k_{OCC} = 0.8$

Figure 5.37.: Final radar generated occupancy grids for the campus and parking lot scenario using model 1. Figures (a), (c): Campus scenario (Ground truth: Figure 5.11). Figures (b), (d): Parking lot scenario (Ground truth: Figure 5.12a).



(a) M2, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.49$, $k_{OCC} = 1.0$

(b) M2, $p_{PD}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.49$, $k_{OCC} = 1.0$

(c) M2, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1$

(d) M2, $p_{SNR}(\boldsymbol{b}_i)$, $\alpha_{RA} = 15$, $k_{FREE} = 0.45$, $k_{OCC} = 1$

Figure 5.38.: Final radar generated occupancy grids for the campus and parking lot scenario using model 2. Figures (a), (c): Campus scenario (Ground truth: Figure 5.11). Figures (b), (d): Parking lot scenario (Ground truth: Figure 5.12a).

5.7. Discussion

The results presented in Section 5.6.3 show that the investigated methods pose a considerable way to create occupancy grids using radar sensors. Naturally they cannot compete in terms of map quality with maps generated using lidars for now, but have other advantages e.g. being able to look beyond certain objects.

The results suggest that $p_{PD}(\boldsymbol{b}_i)$ is a better choice for transforming the SNR into a probability of occupancy than $p_{SNR}(\boldsymbol{b}_i)$. On reason for this is the high sensitivity of the results on the parameters. $p_{PD}(\boldsymbol{b}_i)$ definitely shows more predictable and stable results in that regard. However the discrepancy that increasing α_{RA} decreases the probabilities of occupancy needs to be further investigated. Ideally a mathematically correct model for assigning probabilities to SNR values for the occupancy grid framework is derived.

From the three different modeling approaches model 0 and in some cases model 1 show the best results. However they still have the problem of over- (model 0) or under-estimating (model 1) free space. This should have been solved with model 2, but as the results show this modeling approach needs further refinement. The most important component to improve the map quality is to more closely investigate the function $f_{FREE}(\phi, k_{FREE})$ (Equation 4.10). The results suggest that it should move faster towards 0.5 as the angle moves toward ±90°.

The biggest problem of the introduced approaches is the over-estimation of objects. As the static environment should be mapped, just the utilization of the velocities in the vicinity of the vehicles velocity for the range angle map could improve the quality of the grid maps. This would remove the fragments due to multi-path reflections for the most part, but also just consider the static environment - if the vehicle is moving. However those reflections would not be removed in static scenarios since they appear with the same velocity (0) as the original objects. Additionally the sensitivity for objects that are not directly in front of the sensor could decrease as the sensor measures a radial velocity which is a function of the angle. Of course a sensor with a higher angular resolution would also improve the over-estimation of objects. Finally setting α_{RA} to higher values is also an option. This however would make a more intense investigation of the parameter space of α_{RA}, k_{FREE} and k_{OCC} necessary. A deeper investigation of this space might make sense anyway, especially for model 0 and model 2 using $p_{PD}(\boldsymbol{b}_i)$. The results for model 0, $k_{FREE} = 0.5$ and $\alpha_{RA} = 50$ (e.g. Figure 5.15d) show that even higher values for α_{RA} would make sense as without any compensation by free space the objects are still over-estimated. A value of $0.49 < k_{FREE} < 0.5$ would be necessary. Also for model 2 a value of $k_{FREE} > 0.49$ would make sense (see e.g. Figure 5.38b). In some cases using values for $k_{OCC} > 1$ could also improve the results, however this comes with difficulties in terms of the laws for the probability representations that would need to be addressed.

6. Conclusion and Future Work

In this thesis inverse sensor models for creating occupancy grids using a radar sensor have been developed. Other than the commonly used approach the basis for this models is not the object list but a geometric representation derived from the raw radar data. It represents the sensor data in a form that can be interpreted more intuitively into a probability of occupancy for the gridmap creation. For this purpose two different ways of transforming a SNR value into a probability of occupancy have been derived. This two transformations have been used on three different inverse sensor models. The first model (model 0) directly transformed all SNR values into a probability of occupancy. This model has no explicit free space modeling. The second one (model 1) just considered this probabilities for detections. Free space was only mapped in front of them. The third one (model 2) extended this model by also considering the sensor's beam-pattern for a more accurate free space mapping.

To evaluate the models, dynamic measurements using a mobile robot platform were conducted. This platform was equipped with sensors that provided data for creating occupancy grids using radar. Additionally it was equipped with a laser-scanner to create reference occupancy grids and a camera. This reference gridmaps and the camera images have been used to qualitatively assess the the radar-generated gridmaps. Two different types of measurement scenarios have been conducted. The first one have been regulated measurements with a previously defined number an placement of objects. They have been used to examine the properties of the radar sensor. The second type of scenarios have been real-world scenarios that should test the performance of the developed models on setups that are interesting for future applications. Based on two of them a parametrization of the models has been done. For a formalization of the map quality criteria have been derived and applied to the final gridmaps. Effects that led to a reduced occupancy grid quality have been discussed based on these criteria.

The introduced range angle map has proven to be a considerable alternative to the commonly used object list as a basis for the occupancy grid mapping. For transforming the SNR into a probability of occupancy the approach using the detection probability should be preferred over the purely SNR based method. It proved more stable in it's parameters and achieved a better separation of noise and detections. A general statement which sensor model performed best cannot be made at this point since their performance was strongly scenario dependent. The results of model 2 however fell short of the expectations. It should have been the best performing option as it used to most extensive model of the sensor properties.

A first step for further improvement would therefore be to examine model 2 more closely. Especially the weighting function for free space $f_{FREE}(\phi, k_{FREE})$ plays an important role in this regard. Also using a higher value for α_{RA} is advised. This however requires a deeper investigation the parameter space for k_{FREE} and k_{OCC}.

The biggest challenge of this approach is the over-estimation of objects. A first step to address it would be to only utilize velocities in the vicinity of the vehicles velocity for creating the range angle map. Means for achieving a higher angular resolution would also be highly beneficial for avoiding this over-estimation. One method to do this would be to use more sophisticated angular estimation methods (e.g. Multiple Signal Classification). Another one to use a sensor that can provide a higher angular resolution. A topic for further research is also to adapt the configuration of the sensor according to the situation (e.g. speed of the vehicle) dynamically.

Appendix

Appendix A.

Categorization Tables

Symbol	Meaning
+	Criteria occurs very dominant
o	Criteria can be observed
-	Criteria has a minor influence

Table A.1.: Criteria rating (severeness of occurrence)

The three different levels are shown and explained in table A.1. No rating for a scenario means that this specific criteria does not occur. Table A.2 describes the abbreviations for the column titles. They represent the criteria given in Section 5.5.

Symbol	Meaning
OO	Over-Estimated Occupancy/Objects
UO	Under-Estimated Occupancy/Objects
OF	Over-Estimated Free Space
UF	Under-Estimated Free Space
N	Noise

Table A.2.: Criteria abbreviations

ID	Scenario	OO	UO	OF	UF	N
1	Metal pole 0m	+				-
2	Metal pole 2m	+				-
3	Metal pole 4m	o				
4	Plastic pole 0m	-				
5	Plastic pole 2m	o				
6	Plastic pole 4m		+			
7	Wood pole 0m	-				
8	Wood pole 2m		+			
9	Wood pole 4m		+			
10	Metal box 0m	+				+
11	Metal box 2m	o				o
12	Metal box 4m		+			o
13	Plastic box 0m	o				o
14	Plastic box 2m	-				o
15	Plastic box 4m		+			o
16	Wood box 0m	+				o
17	Wood box 2m					+
18	Wood box 4m		+			+
19	Metal box/Plastic box 2m	+				+
20	Metal box/Plastic box 4m	+	+			+
21	Metal box/Wood box 2m	+	+			+
22	Metal box/Wood box 4m	+	+			+
23	Plastic box/Metal box 2m	+				-
24	Plastic box/Metal box 4m	+	-			-
25	Plastic box/Wood box 2m	+				
26	Plastic box/Wood box 4m	+				
27	Wood box/Metal box 2m	+				
28	Wood box/Metal box 4m	+	o			
29	Wood box/Plastic box 2m	+				
30	Wood box/Plastic box 4m	+	+			
31	Street	-	o	+		-
32	Parking lot	-	o	+		-
33	Single Car	-	o	+		-
34	Campus	o	-			o

Table A.3.: Assessment of model 0 using $p_{PD}(\boldsymbol{b}_i)$.

ID	Scenario	OO	UO	OF	UF	N
35	Metal pole 0m	+			+	-
36	Metal pole 2m	+			+	-
37	Metal pole 4m		o		+	
38	Plastic pole 0m	-			+	
39	Plastic pole 2m	-			+	
40	Plastic pole 4m		+		+	
41	Wood pole 0m	-			+	
42	Wood pole 2m		+		+	
43	Wood pole 4m		+		+	
44	Metal box 0m	+			+	o
45	Metal box 2m	o			+	o
46	Metal box 4m		+		+	o
47	Plastic box 0m	-			+	o
48	Plastic box 2m		o		+	o
49	Plastic box 4m		+		+	o
50	Wood box 0m	+			+	o
51	Wood box 2m		+		+	o
52	Wood box 4m		+		+	o
53	Metal box/Plastic box 2m	+			+	o
54	Metal box/Plastic box 4m	+			+	o
55	Metal box/Wood box 2m	+			-	o
56	Metal box/Wood box 4m	+	+		-	o
57	Plastic box/Metal box 2m	+			+	-
58	Plastic box/Metal box 4m	+			+	
59	Plastic box/Wood box 2m	+			+	
60	Plastic box/Wood box 4m	+			+	-
61	Wood box/Metal box 2m	+			+	
62	Wood box/Metal box 4m	+	+		+	
63	Wood box/Plastic box 2m	+			+	
64	Wood box/Plastic box 4m	+	+		+	
65	Street	-	o		+	
66	Parking lot	-	-		o	-
67	Single Car	o	o	-	+	
68	Campus	+	o		+	-

Table A.4.: Assessment of model 1 using $p_{PD}(\boldsymbol{b}_i)$.

ID	Scenario	OO	UO	OF	UF	N
69	Metal pole 0m	+				-
70	Metal pole 2m	0				
71	Metal pole 4m		0			
72	Plastic pole 0m	-				
73	Plastic pole 2m		-			
74	Plastic pole 4m		+			
75	Wood pole 0m	-				
76	Wood pole 2m		+			
77	Wood pole 4m		+			
78	Metal box 0m	+				0
79	Metal box 2m	-				0
80	Metal box 4m		+			0
81	Plastic box 0m	-				-
82	Plastic box 2m		0			-
83	Plastic box 4m		+			-
84	Wood box 0m	+				0
85	Wood box 2m		0			0
86	Wood box 4m		+			0
87	Metal box/Plastic box 2m	+				0
88	Metal box/Plastic box 4m	+				0
89	Metal box/Wood box 2m	+				0
90	Metal box/Wood box 4m	+	+			0
91	Plastic box/Metal box 2m	+				-
92	Plastic box/Metal box 4m	+				
93	Plastic box/Wood box 2m	+				
94	Plastic box/Wood box 4m	+	-			-
95	Wood box/Metal box 2m	+				
96	Wood box/Metal box 4m	+	+			
97	Wood box/Plastic box 2m	+				
98	Wood box/Plastic box 4m	+	+			
99	Street	-	0	0		-
100	Parking lot		0	0		-
101	Single Car		0	+		-
102	Campus	-	+			-

Table A.5.: Assessment of model 2 using $p_{PD}(\boldsymbol{b}_i)$.

ID	Scenario	OO	UO	OF	UF	N
103	Metal pole 0m	0	0			+
104	Metal pole 2m	+				+
105	Metal pole 4m	+				+
106	Plastic pole 0m		-			+
107	Plastic pole 2m	+				+
108	Plastic pole 4m	+				+
109	Wood pole 0m		+			+
110	Wood pole 2m		+			+
111	Wood pole 4m		+			+
112	Metal box 0m	+	0			+
113	Metal box 2m	-				+
114	Metal box 4m	-				+
115	Plastic box 0m	-	-			+
116	Plastic box 2m	-				+
117	Plastic box 4m	-				+
118	Wood box 0m	0				+
119	Wood box 2m		-			+
120	Wood box 4m		+			+
121	Metal box/Plastic box 2m	0	+			+
122	Metal box/Plastic box 4m	0	+			+
123	Metal box/Wood box 2m	+	+			+
124	Metal box/Wood box 4m	+	+			+
125	Plastic box/Metal box 2m	+				+
126	Plastic box/Metal box 4m	+				+
127	Plastic box/Wood box 2m	+				+
128	Plastic box/Wood box 4m	+	-			+
129	Wood box/Metal box 2m	+	+			+
130	Wood box/Metal box 4m	+	+			+
131	Wood box/Plastic box 2m	+	+			+
132	Wood box/Plastic box 4m	+	+			+
133	Street	0	0	+		+
134	Parking lot	0	0	+		+
135	Single Car	0	0	+		+
136	Campus	0	-			+

Table A.6.: Assessment of model 0 using $p_{SNR}(\boldsymbol{b}_i)$.

ID	Scenario	OO	UO	OF	UF	N
137	Metal pole 0m	+	-		+	-
138	Metal pole 2m	+			+	
139	Metal pole 4m	0			+	
140	Plastic pole 0m	0			+	
141	Plastic pole 2m	0			+	
142	Plastic pole 4m		+		+	
143	Wood pole 0m	0			+	
144	Wood pole 2m		+			
145	Wood pole 4m		+			
146	Metal box 0m	+			+	0
147	Metal box 2m	+			+	0
148	Metal box 4m		+		+	0
149	Plastic box 0m				+	-
150	Plastic box 2m	0			+	-
151	Plastic box 4m		+		+	-
152	Wood box 0m	+	-		+	0
153	Wood box 2m		+		+	0
154	Wood box 4m		+		+	0
155	Metal box/Plastic box 2m	+	+		+	0
156	Metal box/Plastic box 4m	+	+		+	0
157	Metal box/Wood box 2m	+			0	0
158	Metal box/Wood box 4m	+	+		0	0
159	Plastic box/Metal box 2m	+	+		+	0
160	Plastic box/Metal box 4m	+			+	-
161	Plastic box/Wood box 2m	+	-		+	-
162	Plastic box/Wood box 4m	+			+	-
163	Wood box/Metal box 2m	+	-		+	-
164	Wood box/Metal box 4m	+	+		+	-
165	Wood box/Plastic box 2m	+			+	-
166	Wood box/Plastic box 4m	+	+		+	-
167	Street	-	0		+	-
168	Parking lot	-	0		+	-
169	Single Car	0	0		+	0
170	Campus	+	0		+	-

Table A.7.: Assessment of model 1 using $p_{SNR}(\boldsymbol{b}_i)$.

ID	Scenario	OO	UO	OF	UF	N
171	Metal pole 0m	+				
172	Metal pole 2m	+				-
173	Metal pole 4m	-				
174	Plastic pole 0m	-				
175	Plastic pole 2m	-				
176	Plastic pole 4m		+			
177	Wood pole 0m	-				
178	Wood pole 2m		+			
179	Wood pole 4m		+			
180	Metal box 0m	+				+
181	Metal box 2m	-				o
182	Metal box 4m		+			o
183	Plastic box 0m	-				o
184	Plastic box 2m					o
185	Plastic box 4m		+			o
186	Wood box 0m	+				o
187	Wood box 2m					o
188	Wood box 4m		+			+
189	Metal box/Plastic box 2m	+				-
190	Metal box/Plastic box 4m	+				+
191	Metal box/Wood box 2m	+				+
192	Metal box/Wood box 4m	+				+
193	Plastic box/Metal box 2m	+				o
194	Plastic box/Metal box 4m	+				-
195	Plastic box/Wood box 2m	+				-
196	Plastic box/Wood box 4m	+				
197	Wood box/Metal box 2m	+				
198	Wood box/Metal box 4m		+			
199	Wood box/Plastic box 2m	+				
200	Wood box/Plastic box 4m		+			
201	Street	-	o	+		-
202	Parking lot	-	o	+		-
203	Single Car		o	+		
204	Campus	+	+			-

Table A.8.: Assessment of model 2 using $p_{SNR}(\boldsymbol{b}_i)$.

Bibliography

[1] Elfes A. "Using Occupancy Grids for Mobile Robot Perception and Navigation." In: (1989) (cit. on pp. 5, 22).

[2] Foessel A. "Scene Modeling from Motion-Free Radar Sensing." PhD Thesis. The Robotics Institue, Carnegie Mellon University, 2002 (cit. on pp. 8, 9).

[3] Foessel A., Bares J., and Whittaker W. L. "Three-Dimensional Map Building with MMW Radar." In: (2001) (cit. on pp. 7, 37, 43).

[4] Bryan Clarke, Stewart Worrall, Graham Brooker, and Eduardo Nebot. "Sensor Modelling for Radar-Based Occupancy Mapping." In: (2012) (cit. on p. 8).

[5] Li M. *et al.* "High Resolution Radar-based Occupancy Grid Mapping and Free Space Detection." In: (2018) (cit. on p. 7).

[6] Werber K. *et al.* "Automotive Radar Gridmap Representations." In: (2015) (cit. on p. 7).

[7] Lyons R. G. *Understanding Digital Signal Processing 3rd Edition*. 2011. ISBN: 0137027419 (cit. on pp. 6, 10).

[8] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters." In: *Robotics, IEEE Transactions on* 23 (Mar. 2007), pp. 34–46. DOI: 10.1109/TRO.2006.889486 (cit. on p. 5).

[9] Krim H. and Viberg M. "Two decades of array signal processing research: the parametric approach." In: *IEEE Signal Processing Magazine* 13.4 (1996), pp. 67–94 (cit. on pp. 20, 21).

[10] Degerman J., Pernstål T., and Alenljung K. "3D occupancy grid mapping using statistical radar models." In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. 2016, pp. 902–908 (cit. on p. 7).

[11] Dickmann J., Appenrodt N., Klappstein J., Bloecher H., Muntzinger M., Sailer A., Hahn M., and Brenk C. “Making Bertha See Even More: Radar Contribution.” In: *Access, IEEE* 3 (Jan. 2015), pp. 1233–1247. DOI: 10.1109/ACCESS.2015.2454533 (cit. on pp. 1, 8).

[12] Hasch J., Topak E., Schnabel R., Zwick T., Weigel R., and Waldschmidt C. “Millimeter-Wave Technology for Automotive Radar Sensors in the 77 GHz Frequency Band.” In: *IEEE Transactions on Microwave Theory and Techniques - IEEE TRANS MICROWAVE THEORY* 60 (Mar. 2012), pp. 845–860. DOI: 10.1109/TMTT.2011.2178427 (cit. on pp. 1, 9).

[13] Mullane J., Jose E., Adams M., and Wijesoma W. “Including probabilistic target detection attributes into map representations.” In: *Robotics and Autonomous Systems* 55 (Jan. 2007), pp. 72–85. DOI: 10.1016/j.robot.2006.06.010 (cit. on p. 8).

[14] Stefan Kohlbrecher, Oskar Von Stryk, Johannes Meyer, and Uwe Klingauf. “A flexible and scalable SLAM system with full 3D motion estimation.” In: *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (Nov. 2011). DOI: 10.1109/SSRR.2011.6106777 (cit. on p. 5).

[15] Van Trees H. L. *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. 2002. ISBN: 9780471093909 (cit. on pp. 6, 20).

[16] Hammarsten M. and Runemalm V. “3D Localization and Mapping using automotive radar.” MsC Thesis. Department of Signals and systems, Chalmers University of Technology, 2016 (cit. on pp. 7, 31).

[17] Richards A. M. *Fundamentals of Radar Signal Processing 2nd edition*. 2014. ISBN: 9780071798334 (cit. on pp. 6, 18–20).

[18] Carlos Gálvez del Postigo Fernández. “Grid-Based Multi-Sensor Fusion for OnRoad Obstacle Detection: Application to Autonomous Driving.” MsC Thesis. KTH Royal Institute of Technology, 2015 (cit. on p. 6).

[19] Dube R., Hahn M., Schutz M., Dickmann J., and Gingras D. “Detection of parked vehicles from a radar based occupancy grid.” In: June 2014, pp. 1415–1420. ISBN: 978-1-4799-3638-0. DOI: 10.1109/IVS.2014.6856568 (cit. on p. 8).

[20] Prophet R., Stark H., Hoffmann M., Sturm C., and Vossiek M. "Adaptions for Automotive Radar Based Occupancy Gridmaps." In: *2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*. 2018, pp. 1–4 (cit. on p. 8).

[21] Weston R., Cen S., Newman P., and Posner I. *Probably Unknown: Deep Inverse Sensor Modelling In Radar*. Oct. 2018 (cit. on p. 8).

[22] Zou R. "Free Space Detection Based On Occupancy Gridmaps." MsC Thesis. Technical University Darmstadt, Computer Science Department, Intelligent and Autonomous Systems, 2012 (cit. on p. 8).

[23] Bartlett M. S. "Smoothing Periodograms from Time-Series with Continuous Spectra." In: *Nature* (May 1948), pp. 686–687. DOI: 10.1038/161686a0 (cit. on p. 21).

[24] Scheiblhofer S. "A short Introduction to FMCW / FSCW Radar Systems." 2011 (cit. on pp. 15, 17).

[25] Scheiblhofer S. "A short Introduction to Range-Doppler Processing." 2011 (cit. on p. 14).

[26] Thrun S., Burgard W., and Fox D. *Probabilistic Robotics*. 2005. ISBN: 9780262201629 (cit. on pp. 5, 23).

[27] Sebastian Thrun. "Learning Occupancy Grids With Forward Sensor Models." In: (Dec. 2002) (cit. on p. 23).

[28] Oppenheim A. V. and Schafer R. W. *Discrete - Time Signal Processing 3rd Edition*. 2010. ISBN: 978-0-13-198842-2 (cit. on pp. 6, 10).