



INSTITUTE FOR REAL-TIME COMPUTER SYSTEMS
TECHNISCHE UNIVERSITÄT MÜNCHEN
PROFESSOR SAMARJIT CHAKRABORTY



Radar Reprojection Mapping Improves Obstacle Avoidance in Mobile Robots with an Unsteered Radar Sensor

Laurenz Altenmüller

Master's Thesis

Radar Reprojection Mapping Improves Obstacle Avoidance in Mobile Robots with an Unsteered Radar Sensor

Master's Thesis

Supervised by the Institute for Real-Time Computer Systems
Technische Universität München
Prof. Dr. sc. Samarjit Chakraborty

Executed at BSH Home Appliances Corporation
Bosch Research and Technology Center
Palo Alto, California, USA

Advisor: Michael Balszun, M.Sc.

Author: Laurenz Altenmüller
Pommernstr. 9
80809 München

Submitted in September 2017

Abstract

Mobile indoor robots mostly rely on lidar and vision sensors to remotely detect obstacles in their path. These sensors have trouble detecting some common real-world obstacles like transparent surfaces and chair legs. Recent affordable near-range miniature radar sensors enable new solutions. This thesis explores a simple solution for a radar sensor being moving through a static environment without the need for beamforming or a mechanically scanning radar. The idea is put to test in experiments with an Omnidaradar FMCW radar mounted on a Kobuki robot platform. A comparison against lidar and RGBD sensors shows how promising the setup is for navigation in unstructured environments.

TODO German abstract

Acknowledgements

TODO Vielen Dank ...

München, im Monat Jahr

Contents

Acronyms	ix
Nomenclature	xi
1 Introduction	1
2 Theoretical Background	3
2.1 Traditional Obstacle Sensors	3
2.2 Radar Sensing	4
2.2.1 Doppler effect	5
2.2.2 Types of Radar	6
2.2.3 Frequency-modulated continuous wave (FMCW) radar	6
2.2.4 Direction of Arrival	12
2.2.5 Frequencies	13
2.3 Overview of Radar Research	15
2.4 Existing radar-based solutions for map building	16
2.4.1 SAR	16
2.4.2 Scanning radar	19
2.4.3 Radar slam	19
3 Novel Approach: Reprojection Mapping	23
3.1 Idea	23
3.2 Geometry for the Side-Facing 2D Case	23
3.3 Geometry for the General 2D Case	24
3.4 Geometry for the General 3D Case	26
3.5 Geometry for the General 3D Case with Two Sensors	27
3.6 Doppler speed estimation	28
3.7 Map building	28
3.8 Limitations	28
4 Implementation	31
4.1 Hardware Platform	31
4.1.1 Kobuki	31
4.1.2 ROS integration	31
4.1.3 Odroid	31
4.1.4 Lidar Slam	32
4.1.5 Radar sensor	32

Contents

4.1.6	Optimal chirp time configuration	42
4.2	Software Implementation Overview	44
4.3	Driver	44
4.3.1	Omniradar ROS driver	44
4.3.2	ROS node	46
4.3.3	Matlab interface	48
4.4	Data Preprocessing	49
4.4.1	Rosbag to Matlab	49
4.4.2	Raw Data Smoothing	50
4.5	Doppler Estimation with the Peak Gradient Algorithm	52
4.5.1	Inter-scan vs Intra-scan Doppler estimation	54
4.5.2	Subsample peak interpolation	55
4.5.3	Peak matching	55
4.5.4	Transmit crosstalk suppression	56
4.5.5	Minimum Peak height	56
4.5.6	Peaks overlaps at crossing target arcs	58
4.5.7	Output	58
4.5.8	Limitations	59
4.6	DOA Estimation	60
4.7	Reprojection Mapping	62
4.7.1	Orientation parameters	62
4.7.2	Projection direction	63
4.7.3	Sample splitting	64
4.7.4	Range compensation	65
4.7.5	Angle sensitivity compensation	68
4.7.6	Angle compensation window	68
5	Results and Evaluation	73
5.1	Evaluation	73
5.2	Results	74
6	Discussion and Comparison	79
6.1	Discussion	79
6.1.1	Reflection directionality	79
6.1.2	Material-dependent echo strength	79
6.1.3	Doppler vs Direction of Arrival data quality	79
6.1.4	Multipath effects	80
6.1.5	Object penetration	80
6.1.6	Negative obstacles	80
6.1.7	Cable detection	83
6.1.8	Minimum distance	83
6.2	Comparison with other mapping techniques	83
6.2.1	SAR	84
6.2.2	RGBD	85

6.2.3 Lidar	85
7 Conclusion	87
7.1 Future Work	87
7.1.1 Dynamic target rejection	87
7.1.2 Online mapping	87
7.1.3 ROS nodelet	88
7.1.4 Auto-thresholding	88
7.1.5 Realtime	88
7.1.6 Interference Investigation	88
7.1.7 Ultrasound	89
7.2 Conclusion	90
A libomniradar installation	91
A.1 VCO tuning curve import	91
B Usage	93
C List of Scans	95
List of Figures	99
List of Tables	103
Bibliography	105

Acronyms

AESA	Active Electronically Scanned Array
CW	Continuous wave
EM	Electromagnetic
FPGA	Field Programmable Gate Array
FMCW	Frequency modulated continuous wave
IR	Infrared
ISM	Industrial, Scientific, Medical
MCU	Microcontroller unit
MIMO	Multiple In, Multiple Out
RAII	Ressource Allocation Is Initialization
RCS	Institute for Real-time Computer Systems
	Radar cross section
RDK	Radar Development Kit
RMS	Root mean square
SAR	Synthetic Aperture Radar
SNR	Signal to Noise Ratio
TOF	Time of Flight
UWB	Ultra Wide Band
VCO	Voltage Controlled Oscillator

Nomenclature

A_r	Effective antenna aperture	m^2
c	Speed of light	m/s
d	Antenna separation	m
dR	Range resolution	m
f_a	Angle compensation factor	
f_r	Range compensation factor	
G_r	Receive antenna gain	
G_t	Transmit antenna gain	
n_{chirp}	Number of consecutive FMCW chirps	
P_r	Received target power	W
P_t	Transmission power	W
R	Target range	m
S_i	Complex antenna signal	
t_{chirp}	Duration of FMCW chirp	s
t_{msg}	Time between two ROS radar messages	s
v_R	Radar or robot movement speed	m/s
v_D	Relative Doppler speed	m/m
$\Delta\phi$	Phase difference	rad
λ	Wavelength	m
σ	Radar cross section	m^2
τ	Time of flight / Time delay	s

1 Introduction

Mobile indoor robots have become a common sight. Over 15 million units of iRobot's Roomba vacuum robots have been sold since their introduction in 2002 ¹⁾. Applications are not limited to vacuuming: there are service robots, entertainment robots, mapping robots, security robots, and many others. Mobile indoor robots in commercial spaces are upheld as "the next big thing"²⁾.

Robotic navigation has also come a long way. The first vacuum robots were merely equipped with a bumper sensor that let them traverse a room in a random fashion like a TV's DVD logo animation screensaver. In 2010, Neato Robotics managed to develop and integrate a low-cost lidar ³⁾ that first enabled path planning for their vacuum robot. Some robots like LG's Hom-bot⁴⁾ later used ceiling cameras with visual slam (simultaneous localization and mapping) for optical odometry information, and it was in 2014 that iRobot brought a vacuum robot to the market that used visual slam to create a map of its environment ⁵⁾.

There are still stories about vacuum robots failing hilariously in their navigation capabilities, including one woman's hair being eaten as she slept⁶⁾, and of course countless examples of the joke "What is worse than finding dog poop on the carpet? Your robot vacuum finding dog poop on the carpet"⁷⁾. Other obstacles that are really difficult to see with the sensors of today's vacuum robots are very thin objects (such as thin chair legs) in the case of visual navigation, and transparent surfaces like windows and glass walls with both visual and laser sensors. While theoretically it would be possible to detect mirrors just the way humans do it, by observing the geometry visible during movement and inferring the mirrors position and orientation, no home robot has demonstrated that capability so far.

Of course, home robot owners want their robot to have better navigational skills. A robot is more efficient if it can plan a path with knowledge of all obstacles on it. That translates to quicker cleaning. Additionally it is safer, because the robot won't break things and

¹⁾ <http://www.irobot.com/About-iRobot/Company-Information/History.aspx>

²⁾ <http://spectrum.ieee.org/automaton/robotics/robotics-hardware/indoor-robots-for-commercial-spaces>

³⁾ <https://www.sparkfun.com/news/490>

⁴⁾ <https://www.youtube.com/watch?v=UANWyiDf3hA>

⁵⁾ <https://www.technologyreview.com/s/541326/the-roomba-now-sees-and-maps-a-home/>

⁶⁾ <https://www.theguardian.com/world/2015/feb/09/south-korean-womans-hair-eaten-by-robot-vacuum-cleaner-as-she-slept>

⁷⁾ <http://www.boredpanda.com/robot-vacuum-cleaner-spreads-dog-shit-everywhere/>

1 Introduction

won't get hurt if it does not bump into potentially dangerous things.

Thanks to recent efforts in miniaturization and falling costs, there is a new option in the search for better navigation and obstacle sensors. Usually associated with ship and aircraft localization, radar technology made its way to the automotive industry in the form of weather-proof range sensors. The newest sensors are even quite small (some sensors come with antennas integrated in silicon) and cheap, with prices projected to fall below 1€ [61]. Today's embedded processors are also powerful enough to handle any kind of radar signal processing. Combined with the easy licensing in the Industrial, Scientific, Medical (ISM) frequency bands, it seems very worthwhile to investigate how the use of radar will improve obstacle sensing in mobile home robots.

There have been successful attempts at using scanning radars to create better maps of a robot's environment. In consumer products like vacuum cleaning robots, mechanical parts like a spinning radar turret pose a design challenge: Can a moving mechanical part be designed to be maintenance-free and precise while still coming at a cost low enough to reach an overall price that is attractive for the prospective customer? This thesis contributes a novel radar mapping method that pushes a new corner in the design space of such systems. Reprojection mapping is based only on the ratio between a target's radial speed and the sensor's speed of motion, which makes mapping with scanless range sensors possible. Real-world experiments show that in a static environment previously very challenging obstacles like chair legs and glass walls can now be mapped.

To understand why and how radar should be integrated in next-generation home robots the next chapter 2 will give a brief overview of conventional sensing techniques as well as the theoretical background of radar sensing, which lays the foundation of why radar reprojection can work. In chapter 3, reprojection mapping, the novel technique for mapping with unsteered range sensors, is proposed. Chapter 4 then details a proof-of-concept implementation of the method. Results of experiments with this method are listed in chapter 5 and discussed and compared in chapter 6. Finally, chapter 7 concludes the thesis with an outlook and places reprojection mapping within the bigger picture of obstacle sensing.

2 Theoretical Background

A variety of sensors enable robots to sense their environment. Most robots monitor proprioceptive (i.e. self-sensing) information like odometry from inertial measurement units (IMU) and wheel encoders. This section focuses on how a robot can avoid or detect collisions with exteroceptive (i.e. environment-sensing) sensors.

2.1 Traditional Obstacle Sensors

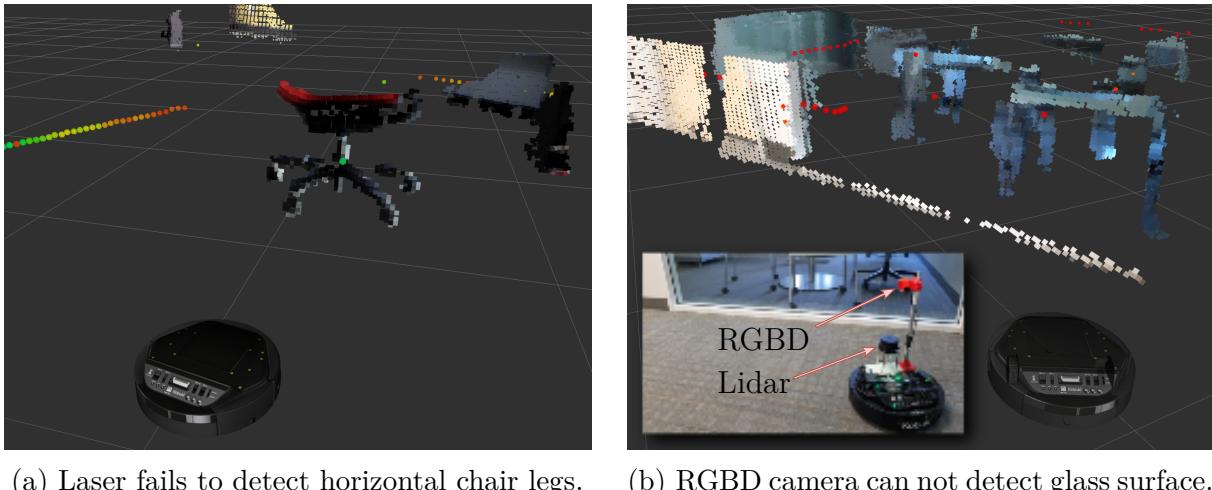
The fallback option of obstacle sensing is almost always a bumper sensor. If the robot runs into an obstacle in its path, a bumper switch will be pressed and the robot's navigation system knows that it can not traverse further in this particular direction. The concept can be extended to measure sudden acceleration that indicates a collision or even monitor motor current to find that the robot is stuck against an obstacle, but the idea is the same: With this kind of sensor, the robot can only detect that it already had a collision. The detection can be shifted to a slightly earlier point in time with whisker-like antennae or capacitive sensors [67]. A special case is negative obstacle (e.g. cliff) detection. Here, infrared (IR) range sensors are usually employed on the underside of the robot.

Intelligent navigation and path planning however can only be achieved with ranging sensors. Classic range sensors are IR-, ultrasound (US)-, or laser-based. They measure the distance to the closest target in one direction. Because obstacles can appear in any direction, these sensors need to be scanned. Scanning means that the bearing of active sensing direction is changed over time to achieve range scans that are multiplexed over a field of view (FOV). Usually this is done mechanically, with a sensor turret spinning on a servo motor. The best known example of this is the classic lidar sensor, which spins a laser range finder around.

Vision sensors are scannerless and include regular cameras as monocular vision sensors and stereo-camera setups to record depth information. Structured light sensors such as the first-generation Microsoft Kinect sensor record depth disparity from triangulation, via correlation between a known and perceived projected light pattern. Time-of-flight (TOF) cameras like the second-generation Microsoft Kinect illuminate the scene with amplitude-modulated near-IR light and calculate depth from the phase shift between the transmitted and received signal [69].

There is also a scannerless version of lidar. Full-field lidar, also known as flash lidar [17], uses TOF measurements of omnidirectional light pulses to capture a 2D scene.

2 Theoretical Background



(a) Laser fails to detect horizontal chair legs. (b) RGBD camera can not detect glass surface.

Figure 2.1: Screenshots showing traditional sensor shortcomings.

To map out the environment and localize itself, the robot will usually employ a version of a slam algorithm [72] — After years of being a research field, slam is now finally showing up in products.

Figure 2.1a shows a situation where a robot’s laser scanner only detects a single point (green) on the pillar of an office chair. The resulting gridmap will feature a single occupied cell at this location, but the horizontal chair legs are completely undetected. In practice, the robot often bumps into the “invisible” chair legs and sometimes gets stuck while navigating in this environment. These horizontal chair legs can be detected with a depth camera, which is the source of the chair’s 3D image and the 3D occupancy grid in figure 2.1a. The vision/depth sensor is however also not perfect: A decent amount of processing power is necessary to run a visual slam; but the bigger problem is that some obstacles are still not visible. In figure 2.1b, chairs behind the window and a part of the window frame are recognized, but the glass wall is literally completely transparent.

2.2 Radar Sensing

Radar sensing is based on the transmission and reflection of electromagnetic signals. A transmitting antenna radiates EM energy, some of which is scattered off reflective objects, called targets, and intercepted by a receiving antenna. This signal is amplified and checked for time delay, frequency shift, phase shift and amplitude attenuation with respect to the transmitted signal. This allows to capture certain target properties like range, radial velocity, size, shape and, among others, even surface smoothness and orientation [20].

The return signal's echo power is described for the interference-free case in vacuum by the radar equation,

$$P_r = \frac{P_t G_t}{4\pi R^2} \cdot \frac{\sigma}{4\pi R^2} \cdot A_r \quad (2.1)$$

where

P_r	received target echo power
P_t	transmission power
G_t	transmit antenna gain
R	range of target (distance)
σ	radar cross section (RCS) of target
A_r	effective area of receiving antenna.

Of the three factors in the equation, the first factor represents the power density at the radar-illuminated target's distance. The second factor represents how much of the radar energy is scattered back by the target. The third factor denotes how much of the echo power is collected by the receiving antenna. [20] Conventional radar only becomes useful with directive antennas. The antenna gain G_t is defined as the ratio of increased power in a particular direction compared with that from an isotropic antenna [34]. Antenna theory shows [59] the relation of receive antenna gain G_r with radiation wavelength λ :

$$G_r = \frac{4\pi A_r}{\lambda^2} \quad (2.2)$$

With constant antenna loss factor $L > 1$, substituting equation 2.2 into equation 2.1 yields the classical radar equation

$$P_r = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 R^4 L} \propto \frac{\sigma}{R^4} \quad (2.3)$$

In practice, the actual received power is lower than predicted by this equation due to many factors including interference and atmospheric conditions. σ is also not constant but varies with viewing angle and material properties of the target. [34]

2.2.1 Doppler effect

Many radar systems measure radial velocity with the Doppler frequency shift. Austrian physicist Christian Doppler described the kinematic effect in 1842. It describes the change of wavelength caused by the motion. A common example is the change of pitch that can be heard when a race car or ambulance passes the observer. The Doppler frequency shift f_D is

$$f_D = 2 \frac{v_r}{\lambda} = 2 \frac{v \cos(\theta)}{\lambda} \quad (2.4)$$

where v_r is the radial velocity component of the target, which travels at a speed v at angle θ between the target's direction and the radar beam with wavelength λ [20]. The factor 2 is caused by the Doppler shift being applied twice; once for the incident wave, and once for the reflected wave. In effect this means that a radar sending out an EM wave with a

2 Theoretical Background

frequency of exactly 60 GHz towards a target moving at a relative speed of $v_r = 1 \text{ cm/s}$ towards the radar will receive back an echo with a frequency of 60 000 000 004 Hz because of the frequency shift of $f_D = 4 \text{ Hz}$.

2.2.2 Types of Radar

Continuous Wave radar

Continuous wave (CW) radar is the earliest radar system. It uses a continuous transmission at a fixed frequency. Thanks to antenna directionality it can find a target's azimuth in radio direction finding. A target's velocity information can be extracted from the frequency shift due to the Doppler effect.

Pulse radar

Pulse radars send series of short bursts. The time delay τ between transmission and reception of a pulse is called time of flight (TOF). Together with speed of light c it yields a target's range R :

$$R = \frac{c\tau}{2} \quad (2.5)$$

The range resolution ΔR is given by

$$\Delta R = \frac{c\tau_m}{2} \quad (2.6)$$

with τ_m the pulse high time of the burst modulation (see figure 2.2). To achieve high range resolution pulses must be very short, which requires very high peak transmission power to still produce a detectable echo signal. Pulse compression radars send a longer pulse with an internal modulation, which combines the higher transmission energy of longer pulses with the resolution of short pulses. Velocity is again known from frequency shift.

2.2.3 Frequency-modulated continuous wave (FMCW) radar

FMCW radars use a frequency modulation to measure range and speed at the same time. The transmitted modulation is compared to the modulation in the received signal to detect signal delay and frequency shift. Applications in robotics use this kind of radar the most, for reasons of lower transmission power and high-range resolution [34].

An FMCW radar's modulation is called a frequency sweep or chirp and is usually triangular with a linearly increasing and decreasing frequency. Most sensors use a voltage controlled oscillator (VCO) to generate the modulation waveform. VCOs do not have a linear transfer function, so in order to obtain a linear sweep, the input to the VCO must be pre-distorted with the inverse of the VCO's nonlinear transfer function. Instead of

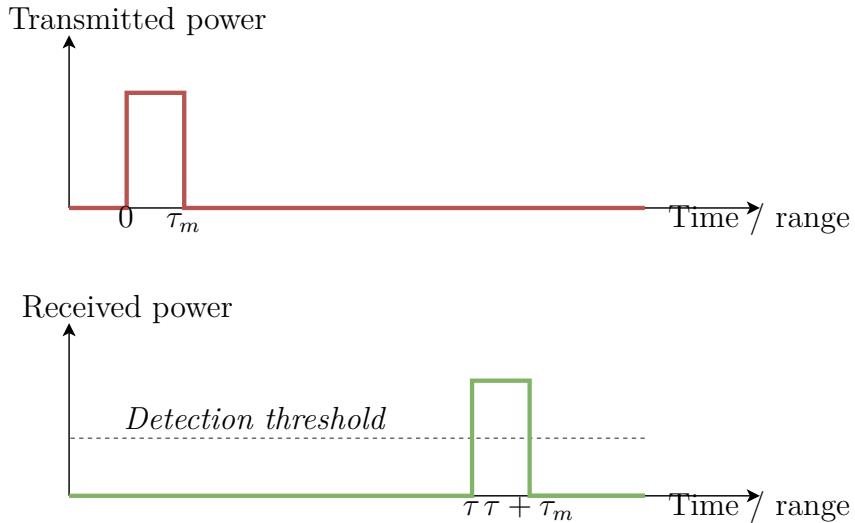


Figure 2.2: Pulse radars measure the time between transmission and reception of a short EM burst. Adapted from [34] p. 52.

a VCO, direct digital synthesizers together with phase-locked loops (PLL) can be used. They generate better (more linear) sweeps at the price of increased design complexity and cost [74].

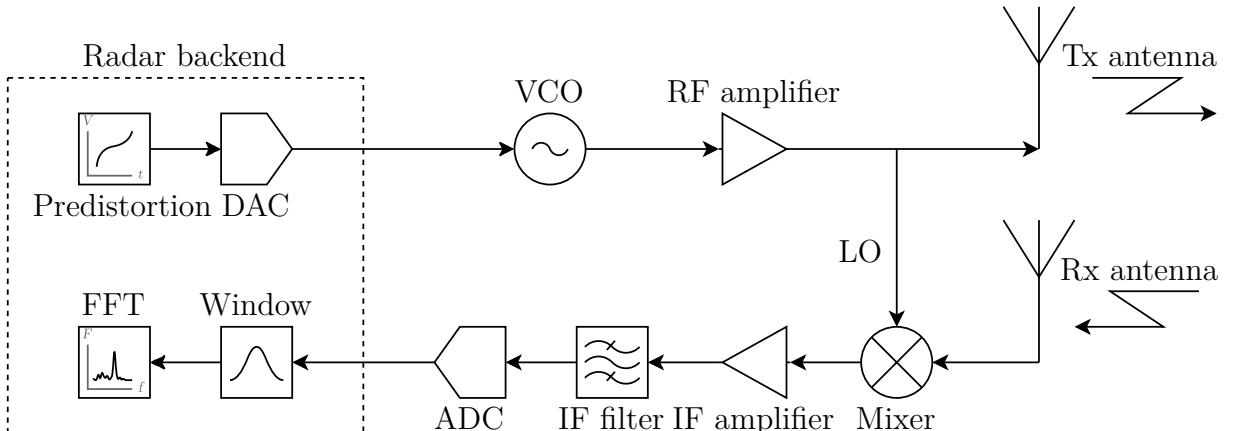


Figure 2.3: Simplified FMCW architecture. Adapted from [57].

After the VCO's signal is amplified and transmitted, it reflects at visible targets and is received as echo in the same frequency band.

Figure 2.4's top subplot shows the transmitted frequency sweep from f_0 to $f_0 + \Delta f$ over a sweep length of T_d of a triangular modulation. The middle plot also shows the received signal as caused by a single stationary ideal reflector. Time of flight causes a delay τ in the received signal. To understand where the beat signal comes from, we focus on the rising part of the triangle modulation, the upsweep. Using the superheterodyne principle the received signal v_{Rx} and a portion of the transmitted signal v_{Tx} (called the local oscillator (LO)), are frequency mixed in an analog multiplier to get the intermediate frequency (IF)

2 Theoretical Background

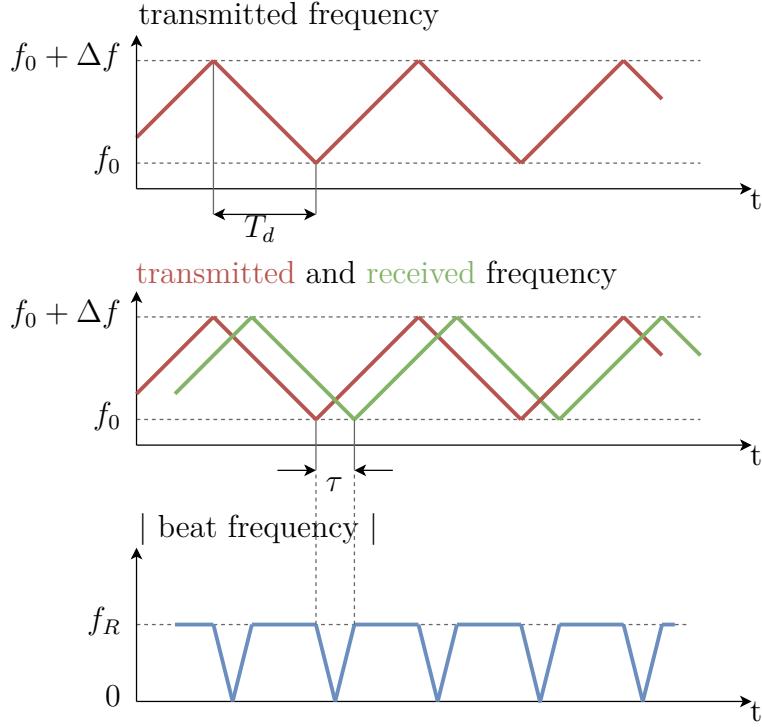


Figure 2.4: FMCW radars detect targets in the beat frequency, which is a frequency mix of the transmitted and received modulation. Adapted from [34] p. 57.

v_{Mixer} . The IF contains a target's beat frequency, which is proportional to the target's range. With the transmitted signal v_{Tx} and the received signal v_{Rx} as a function of time t ,

$$v_{Tx}(t) = A_{Tx} \cos(\omega_{Tx}(t) t) \quad (2.7)$$

$$v_{Rx}(t) = A_{Rx} \cos(\omega_{Tx}(t - \tau) t) \quad (2.8)$$

where

$$\omega_{Tx}(t) = \underbrace{\omega_c}_{\text{Carrier frequency}} + \underbrace{\pi \frac{\Delta f}{T_d} t}_{\text{Upsweep modulation}}$$

is the (angular) frequency of the transmitted signal. The signal v_{Mixer} behind the frequency mixer can be calculated as the mix (i.e. multiplication) of $v_{Tx}(t)$ and $v_{Rx}(t)$ from equations 2.7 and 2.8:

$$\begin{aligned} v_{Mixer}(t) &= v_{Tx}(t) v_{Rx}(t) \\ &= A_{Tx} A_{Rx} \cos(t\omega_{Tx}(t)) t \cos(\omega_{Tx}(t - \tau) t) \end{aligned} \quad (2.9)$$

With the trigonometric identity

$$\cos A \cdot \cos B = \frac{1}{2} (\cos(A + B) + \cos(A - B))$$

equation 2.9's v_{Mixer} can be rewritten as

$$v_{Mixer}(t - \tau) = \frac{A_{Tx}A_{Rx}}{2}(B_1 + B_2)$$

where

$$\begin{aligned} B_1 &= \cos \left[2\omega_{Tx}(t - \tau)t - \omega_{Tx}(\tau)\tau \right] \\ &= \cos \left[2 \left(\omega_c + \pi \frac{\Delta f}{T_d}(t - \tau) \right) t - \left(\omega_c - \pi \frac{\Delta f}{T_d}\tau \right) \tau \right] \end{aligned} \quad (2.10)$$

$$\begin{aligned} B_2 &= \cos \left[2 \left(\pi \frac{\Delta f}{T_d}t \right) \tau - \omega_{Tx}(\tau)\tau \right] \\ &= \cos \left[2\pi \left(\frac{\Delta f}{T_d}\tau \right) t - \left(\omega_c + \pi \frac{\Delta f}{T_d}\tau \right) \tau \right] \end{aligned} \quad (2.11)$$

Note that B_1 consists of high angular frequencies around the carrier frequency, from $f_0 = \frac{\omega_c}{2\pi}$ to $f_0 + \Delta f$. B_2 is a lower frequency (theoretically up to $2\pi\Delta f$ at $\tau = T_d$, but much lower in practice, as echos from targets this far away have very low intensity A_{Rx} and can't be detected) signal containing the beat frequency. The output of the low-pass filter intrinsic in the mixer stage will thus only consist of the beat frequency (plus noise of similar frequencies) from equation 2.11:

$$v_{beat}(t, \tau) = \frac{ATx A_{Rx}}{2} \cos \left[2\pi \left(\frac{\Delta f}{T_d}\tau \right) t - \left(\omega_c + \pi \frac{\Delta f}{T_d}\tau \right) \tau \right] \quad (2.12)$$

The term $\frac{\Delta f}{T_d}\tau$ in equation 2.12's $v_{beat}(t, \tau)$ is known as the beat frequency f_b . For stationary targets, the range-specific frequency $f_R = f_b$. Knowing that the delay time τ depends on the speed of light c and the range R , the relationship between a target's f_R and range R can be given as

$$f_R = \frac{2R}{c} \frac{\Delta f}{T_d} \iff R = \frac{cT_d}{2\Delta f} f_R \quad (2.13)$$

This also gives the range resolution dR ,

$$dR = \frac{c}{2\Delta f} \quad (2.14)$$

A moving target however will introduce a Doppler shift f_D in the received signal v_{Rx} , which will shift the target's beat frequency f_b away from its range-specific frequency f_R . The direction of the frequency shift depends on the modulation: An up-sweep will have a corresponding $f_{b,up}$, while the down-sweep will have $f_{b,down}$

$$f_{b,up} = f_R - f_D \quad (2.15)$$

$$f_{b,down} = f_R + f_D \quad (2.16)$$

2 Theoretical Background

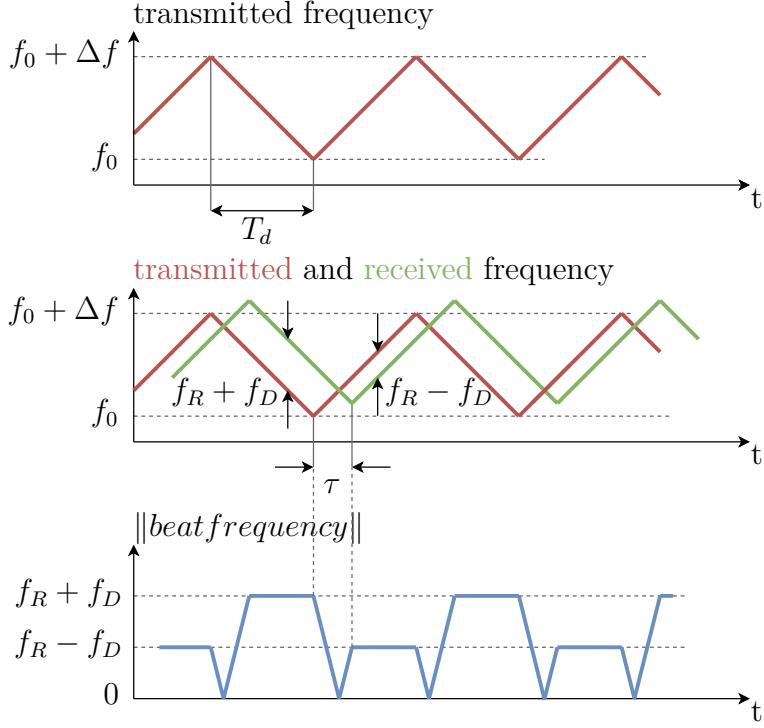


Figure 2.5: Target motion introduces a Doppler shift in the received signal, which changes the beat frequency during up- and down-sweeps. Adapted from [34] p. 57.

The range-specific frequency and Doppler frequency can be extracted from the two beat frequencies of equations 2.15 and 2.16 by averaging them:

$$f_R = \frac{f_{b,down} + f_{b,up}}{2} \quad (2.17)$$

$$f_D = \frac{f_{b,down} - f_{b,up}}{2} \quad (2.18)$$

The benefit of the triangular sweep becomes clear here: with a sawtooth waveform, only $f_{b,up}$ can be determined. A stationary target and a moving target a range and Doppler speed corresponding to the same resulting frequencies would not be distinguishable.

Of course, more than one target can be visible at a time. If multiple echos are received, as in figure 2.6, the intermediate frequency will contain multiple frequencies. The beat signal will have more than one dominant frequency in its spectrum, with each one corresponding to a different target.

Finally, with the help of equation 2.13, this frequency spectrum can be interpreted as range profile, with every target's dominant frequency corresponding to the target's range.

Figure 2.7 shows a real-world example of how the beat signal f_b (figure 2.7a) and its frequency spectrum (figure 2.7b) look like. At $t = T_d = 32$ ms, a jump in the beat frequency is caused by the modulation change from upsweep to downsweep. In the frequency spectrum, three stationary targets are visible at ca. 3 kHz, 6 kHz, and 9 kHz. In this example,

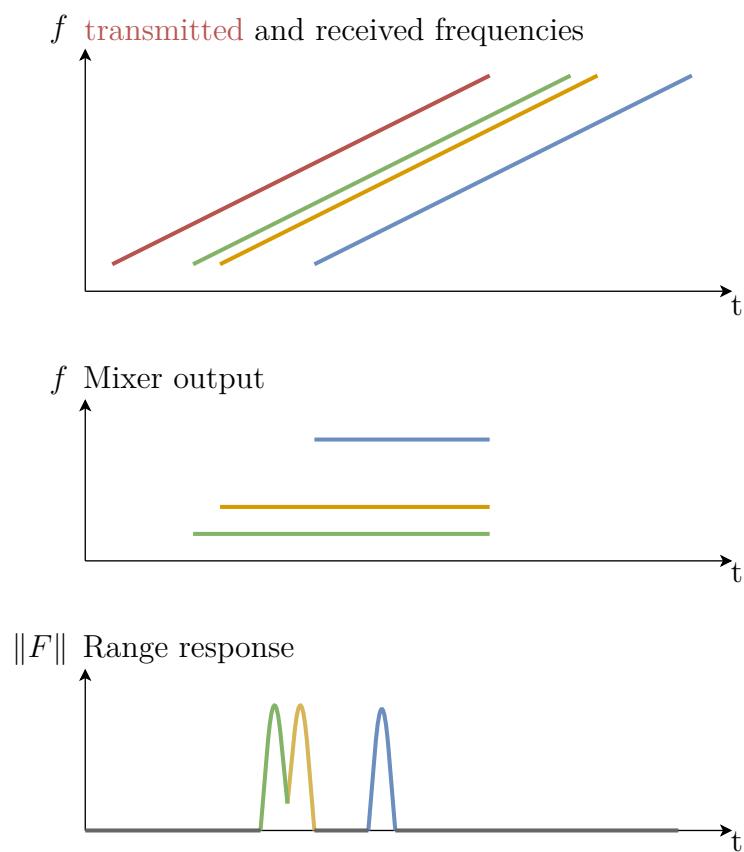


Figure 2.6: By reflecting a portion of the transmission signal back to the radar, each target (green, yellow and blue lines) contributes a frequency corresponding to its range in the beat spectrum / range response.

2 Theoretical Background

$T_d = 32 \text{ ms}$ and $\Delta f = 7 \text{ GHz}$, so the targets were at ranges of 0.5 m, 1 m, and 1.5 m. The Fourier transform in figure 2.7b shows the frequency spectrum's magnitude $\|S\|$.

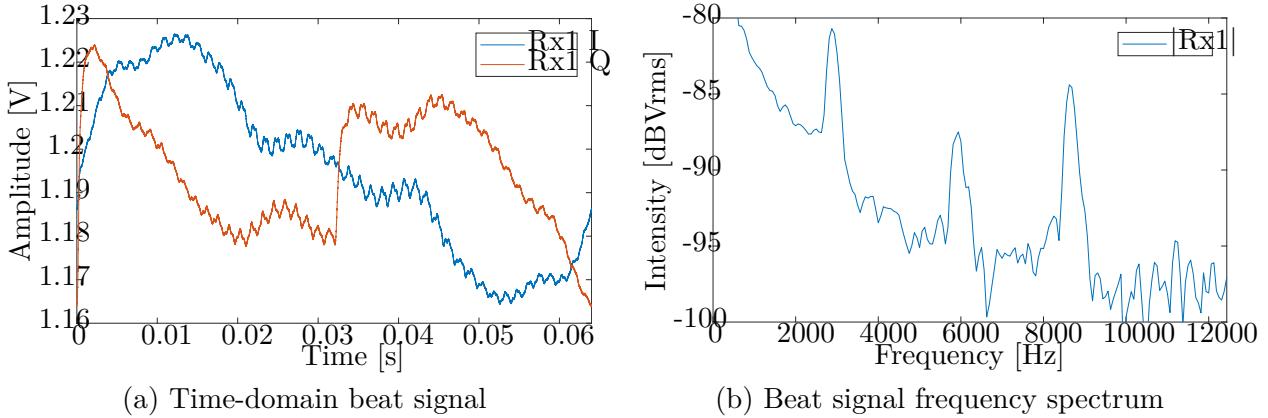


Figure 2.7: Beat signal and spectrum / range profile in a real-world measurement.

2.2.4 Direction of Arrival

A radar sensor with two or more receiving antennas which are separated by not more than half a wavelength can measure the Direction of arrival (DOA) of one or multiple targets. Because the echo from a target has to travel a slightly longer distance to antennas further away, a phase difference between the different antenna signals is measurable.

The basis for this is that not only the magnitude $\|S\|$ of a signal is received, but that also the in-phase component S_I and quadrature component S_Q of the analytic signal are measured in separate channels in order to retain phase information. This is the reason for separate $RX\ 1\ I$ and $RX\ 1\ Q$ signals in figure 2.7a. Figure 2.8a shows how the two components make up the complex signal. The physical reason for the two parts existing is the oscillation of the radar wave's EM energy between electric and magnetic fields. The frequency f of the wave determines the speed of the oscillation and the phase ϕ indicates at which point in this oscillations cycle the wave currently is. As the propagation speed of the wave is limited to the speed of light c , two spatially separated antennas receive the same signal at different oscillation phases. This difference is the phase difference $\Delta\phi$. Other than the spatial antenna separation d , $\Delta\phi$ only depends the wave's frequency f (or wavelength λ) and the direction of arrival θ (see figure 2.8b).

Viewing a complex analytic received signal S at time t ,

$$S(t) = S_I(t) + j S_Q(t) \quad (2.19)$$

as

$$S(t) = A_{Rx} \exp(j\phi)$$

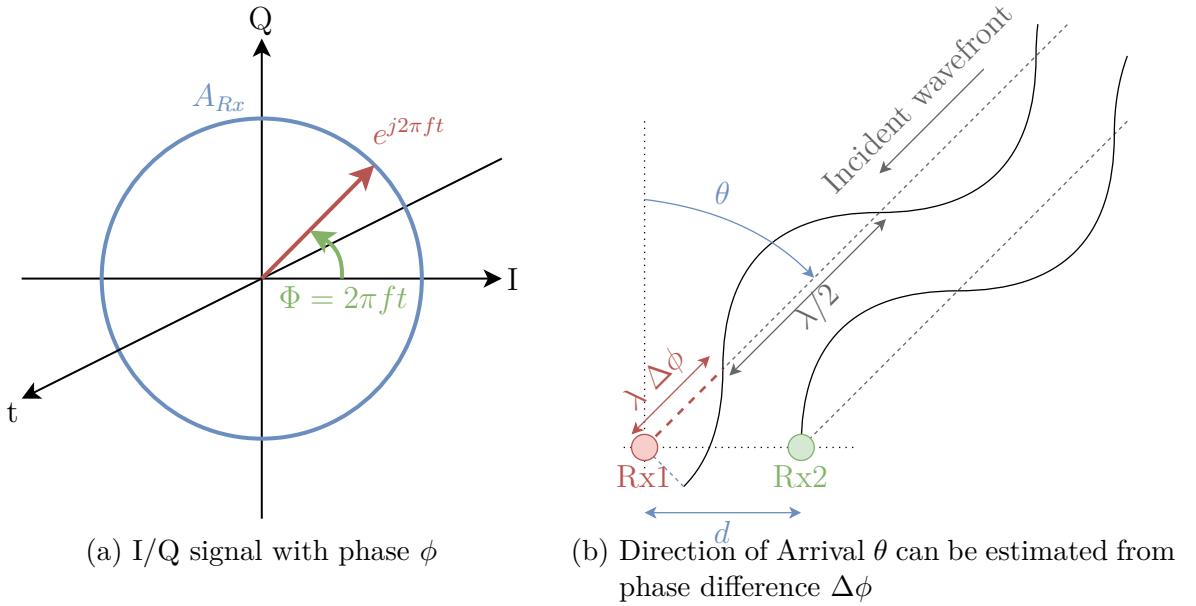


Figure 2.8: Phase difference and I/Q signal

explains why the phase difference $\Delta\phi$ can be calculated from the complex angle difference of the complex analytic antenna signals S_{Rx1} and S_{Rx2} at antennas $Rx1$ and $Rx2$, respectively:

$$\Delta\phi = \angle S_{Rx1} - \angle S_{Rx2} \quad (2.20)$$

With equation 2.20's $\Delta\phi$ direction of arrival θ can be estimated [57] as

$$\theta = \sin^{-1} \left(\frac{\lambda}{2\pi d} \Delta\phi \right) \quad (2.21)$$

DOA estimation has been thoroughly studied and many algorithms have been designed and evaluated, including subspace-based decorrelation methods like MUSIC and ESPRIT [2, 3, 56], and beamforming techniques like Bartlett and Capon [1, 44] and others like the method of minimum phase error proposed by Cho *et al.* in [91]. The presented approach is a simplified two-antenna version of [91]. The implementation that will be detailed in chapter 4 does not require a very accurate θ but relies mostly on $\text{sgn}(\theta)$. Results in section 5.2 show however that θ is a reasonable estimation that holds well for strong signals.

2.2.5 Frequencies

Even though radar applications exist for many frequencies, only a few of them are OK to use for radiolocation and in home robots. The “Industrial, Scientific, Medical” (ISM) bands allow the unlicensed use of some frequencies for radiolocation, including center frequencies of 24.125 GHz, 61.25 GHz, 122.5 GHz and 245 GHz. Applications must, however, accept harmful interference [92].

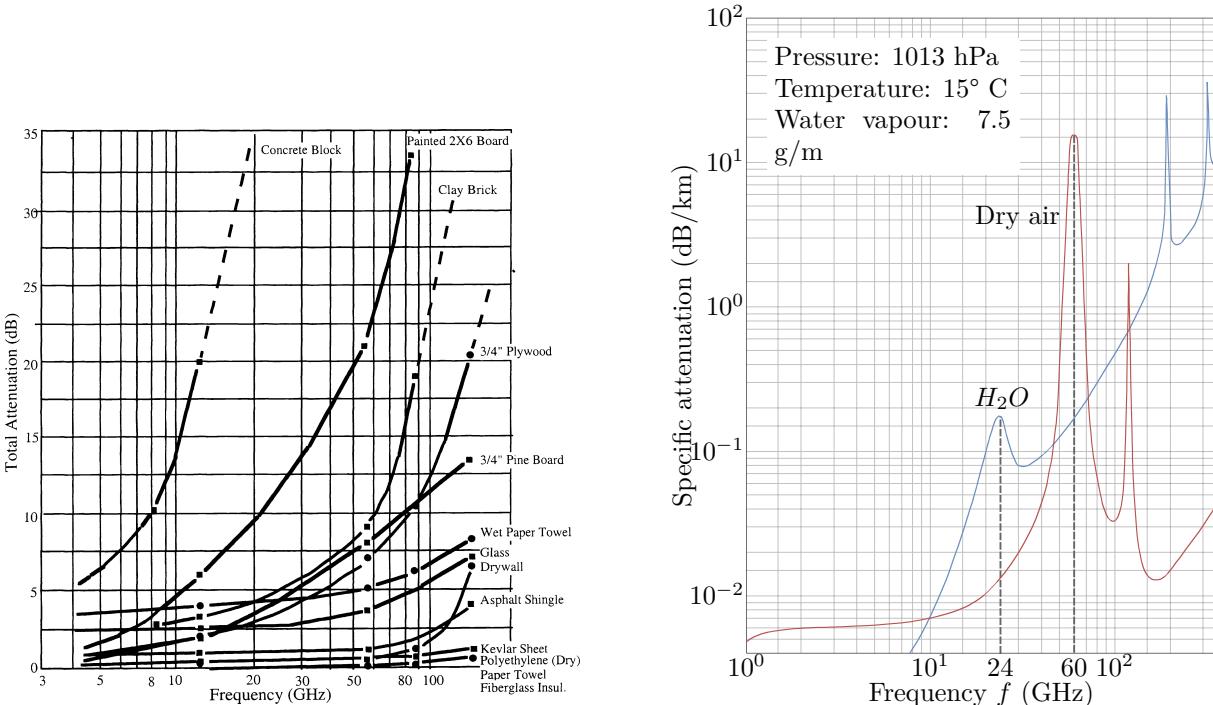
2 Theoretical Background

The 77 GHz band is “restricted to vehicle-mounted field disturbance sensors used as vehicle radar systems.” (FCC Part 15 §15.253(c)). The European Telecommunications Standards Institute (ETSI) defines it as “Automatic Cruise Control ‘long-range radar’ operating at 77 GHz. This enables a vehicle to maintain a cruising distance from a vehicle in front.” (EN 301 091). The German Bundesnetzagentur also declares it “Kraftfahrzeug-Kurzstreckenradar” (Vfg 66 / 2014).

A 24 GHz center frequency is a safe bet. There are many radar systems available and it being an ISM band makes licensing much easier. The drawback is the very limited maximum bandwidth of 250 MHz in this band.

Some newer radars use the 60 GHz ISM band. It allows a rather wide bandwidth of up to 9 GHz in some regions (see figure 2.10). According to equation 2.14, this gives a very good range resolution in the order of a few cm. At these high frequencies, RF energy attenuation in material increases noticeably [6]. The effect is that 60 GHz waves are limited to short ranges of a few m and don’t usually penetrate walls.

Atmospheric attenuation also limits long-range applications. At short ranges it should however not present a problem. This is why these high frequencies are often used in indoor applications with transmission paths of not more than a few m.



- (a) Total attenuation of RF energy when transmitted through various materials as a function of frequency. Source: [6]

- (b) Specific RF attenuation due to atmospheric gases. Source: [5]

Figure 2.9: RF attenuation in different media.

The downside is that there are some other technologies using the same frequency bands,

most notably 802.11ad a.k.a. WiGig [39]. The WiGig frequency allocations in figure 2.10 show in which regions the 60 GHz band is available (also for radar).

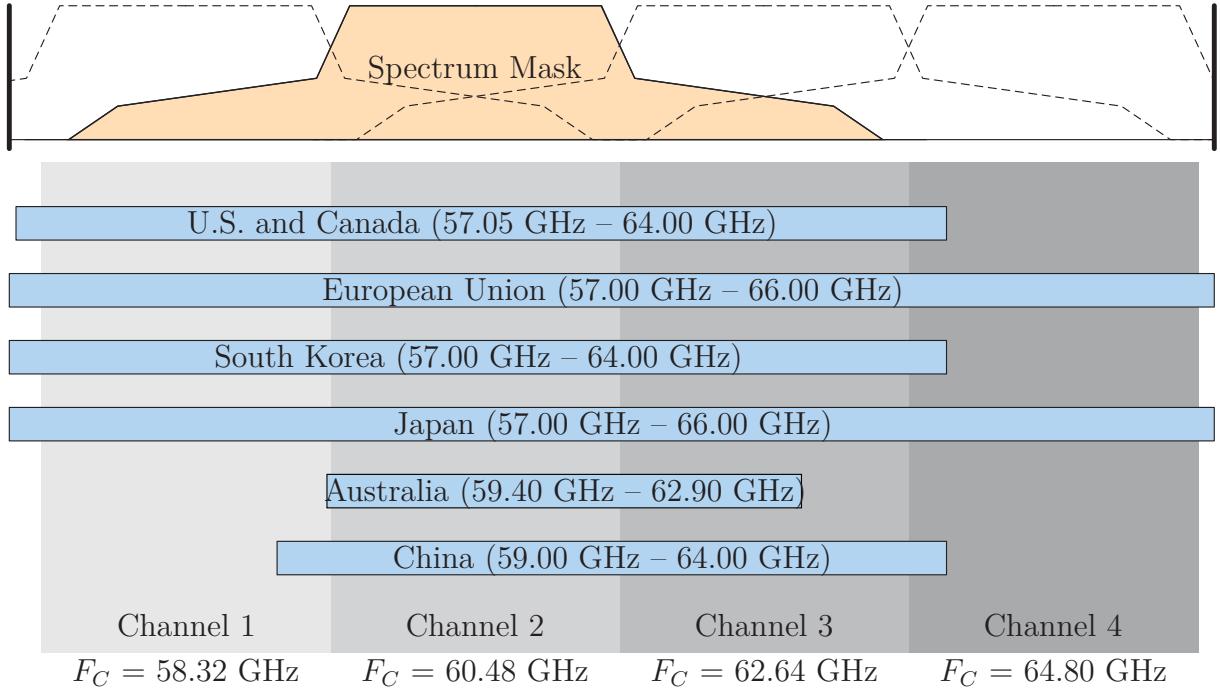


Figure 2.10: WiGig Channel Plan and Frequency Allocations by Region. Source: [39]

2.3 Overview of Radar Research

Radar has been used and researched since the 1940s. While it was historically only used to detect aircraft and ships, it is an active research domain in many fields today. Identification and localization of vessels is still an important application in both civil and military sectors. There is a great amount of research going into synthetic aperture radar, in terrestrial imaging, and general and concealed imaging [8, 9, 14, 22, 25, 46, 88]. Another area of research is radar antenna technology and quasi-optics, which aims to find design improvements and more adapted antennas for the manifold applications [15, 16, 48, 54, 70, 75, 80, 87]. Radar is used in human presence detection and monitoring, including heartbeat detection [32, 33, 51, 65, 68, 74, 86, 89, 95]. A new and very promising discipline is radar-based gesture recognition, which enables innovative human machine interaction applications [66, 78, 79]. Indoor communication and localization with radar beacons is another interesting and upcoming technology [29, 36, 71, 87, 89, 90]. The radar-related research area that is most relevant for mobile robots is radar-based slam [10–13, 18, 19, 23, 24, 26, 30, 34, 35, 38, 40, 45, 52, 76, 81, 83, 84, 93].

2.4 Existing radar-based solutions for map building

2.4.1 SAR

In 1950 Doppler frequency analysis was found to improve image resolution of side-looking radar, which led to the development of the synthetic aperture radar (SAR) technique. SAR uses azimuth (along-track) motion to synthesize an aperture that is longer than the physical size of the radar antenna [22]. The three major configurations are stripmap SAR, scan SAR, and spotlight SAR. Current SAR systems can operate in either mode by dividing their planar antenna into sub-apertures, whose phase and amplitude are controlled by the individual few-hundred transmit/receive modules [46].

Radar echo data is sampled in both fast-time and slow-time, with fast-time meaning the range scan dimension (fast, because the EM waves travel at very high speed, c) and slow time denoting the azimuth or along-track dimension (slow, because movement velocity will be $\ll c$). This raw data does not give any useful information and needs to be signal-processed first. Because SAR systems typically use pulse-compressed radars, each range line needs to be convoluted with the complex conjugate of the transmitted chirp's spectrum to obtain the range-compressed data¹⁾. In a second step, azimuth compression takes place by convolving the signal in slow-time with the complex conjugate of the expected azimuth-response from a target.

An elemental scatterer at range $R(t)$ will return an echo $s_a(t)$ over time t :

$$s_a(t) = P_r \sqrt{\sigma} \exp(j\varphi_s) \exp\left(j \underbrace{\frac{-4\pi}{\lambda} R(t)}_{\text{az. phase var. } \omega_D}\right) \quad (2.22)$$

where P_r is the echo power of the received target, accounting for dependencies like transmit power and path loss, σ is the target's RCS, imaginary unit j , φ_s the scattering phase, and $\frac{4\pi}{\lambda} R(t)$ the azimuth phase variation due to changing distance [9]. The target's range $R(t)$ is described by the range at closest approach R_0 and the radar's (constant) movement speed v_R :

$$R(t) = \sqrt{R_0^2 + (v_R t)^2} \approx R_0 + \frac{(v_R t)^2}{2R_0} \text{ for } \frac{v_R t}{R_0} \ll 1 \quad (2.23)$$

Substituting equation 2.23 into the azimuth phase in equation 2.22 and derivating with respect to time yields the azimuth frequency f_D

$$f_D = \underbrace{\frac{1}{2\pi}}_{\omega_D=2\pi f_D} \frac{\partial}{\partial t} \frac{-4\pi}{\lambda} \underbrace{\left(R_0 + \frac{(v_R t)^2}{2R_0}\right)}_{R(t)} = -\frac{2v_R^2}{\lambda R_0} t \quad (2.24)$$

¹⁾ For the FMCW system used in the later parts of this thesis this is not necessary - the FMCW beat frequency spectrum is equivalent to the range-compressed data

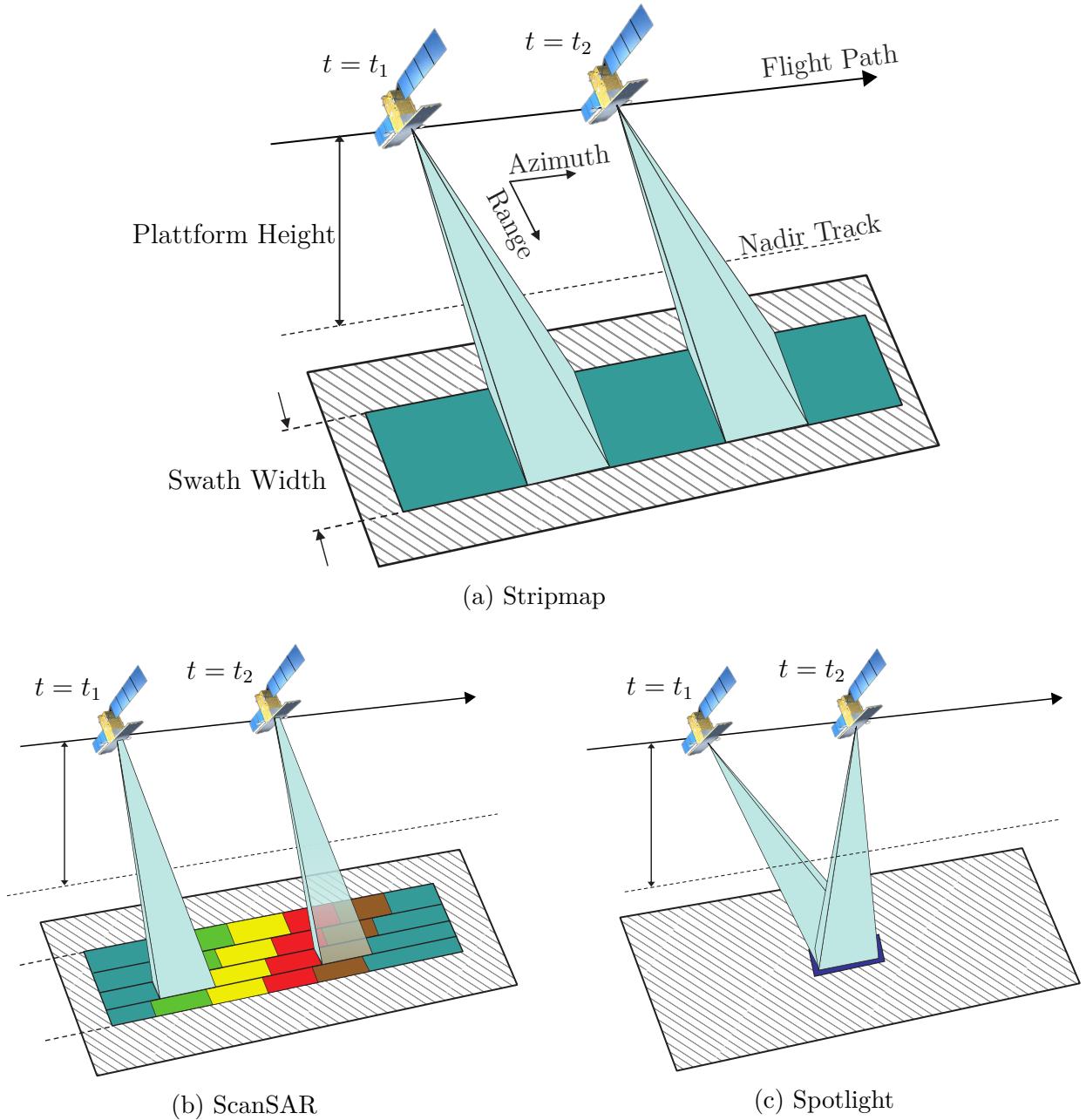


Figure 2.11: Illustration of different SAR operation modes which are used to increase the swath width (ScanSAR) or improve azimuth resolution (Spotlight) compared to Stripmap mode. Adapted from [46]

2 Theoretical Background

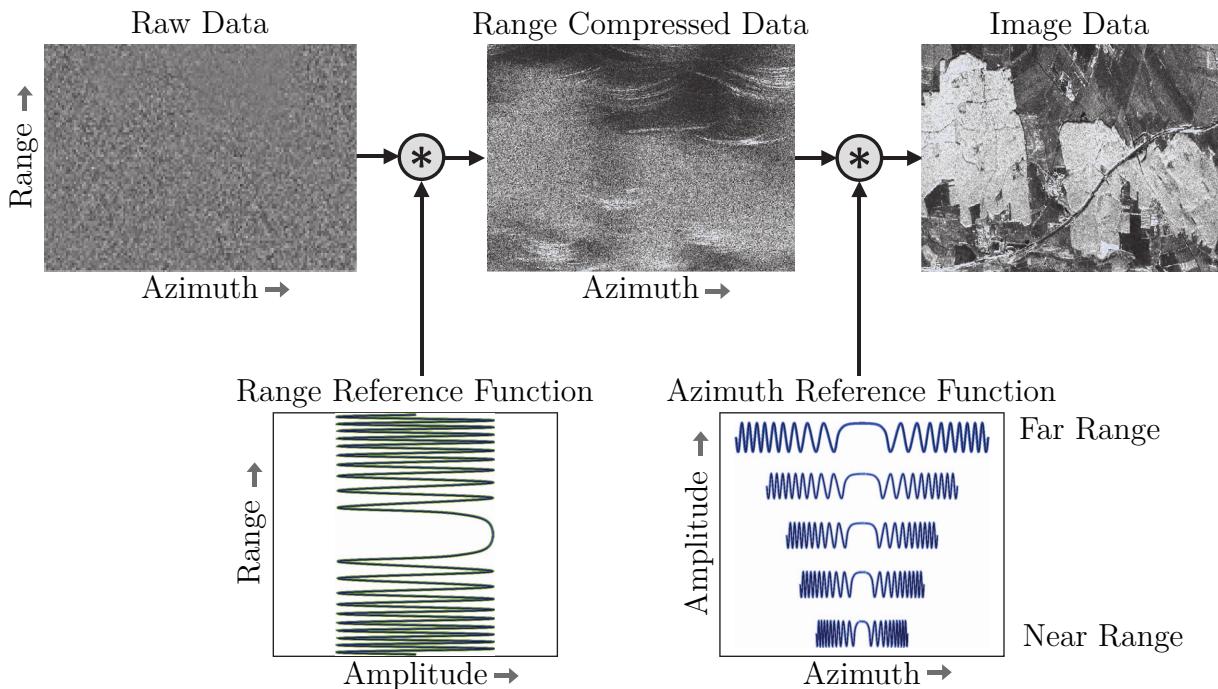


Figure 2.12: Summary of SAR processing steps where the range compressed data result from a convolution of the raw data with the range reference function. In a second step the azimuth compression is performed through a convolution with the azimuth reference function, which changes from near to far range.
Source: [46]

The azimuth frequency varies linearly with time t and is inversely proportional to the closest approach (slant range) R_0 , hence the azimuth reference function depends on geometry and is adapted to range. Because of the frequency-shifting effect it is analogous to and also called the Doppler frequency.

The most challenging aspect of SAR is the correction of range cell migration induced defocusing. Range cell migration is visible in figure [figure 2.12](#)'s curvature of range compressed data. It occurs when a point target's echo energy is distributed over several range cells, causing azimuth defocusing. This effect is range-variant, as the curvature depends on R_0 . Hence a non-stationary two-dimensional reference function is necessary. There are several approaches in tackling this, including the omega-k / wavenumber processor, range-Doppler, and chirp scaling algorithms [\[46\]](#).

As far as moved radar sensors go, SAR is a great technique for imaging a scene while it is passed on the side. If obstacle detection is to be performed, it is however necessary to know what lies ahead. Today, robots that use a radar sensor for this task therefore use a scanning radar.

2.4.2 Scanning radar

A radar system that changes the direction of its beam in a scanning pattern is called a scanning radar. This allows covering a larger overall search space by sampling a set of smaller regions, usually with a tighter beam shape for higher spatial accuracy. The scanning process can be implemented by mechanically rotating the radar's antenna, electronically through phased arrays (see [figure 2.13a](#)), or even both in active electronically scanned arrays (AESAs), which are used in fighter jets (see [figure 2.13b](#)). Some research robots use the K2PI radar sensor [\[18\]](#), which rotates a pencil-shape beam over 360° (see [figure 2.13c](#)).

2.4.3 Radar slam

Simultaneous localization and mapping algorithms are of course a very useful tool in the effort to create an accurate map of a robot's surroundings and have been referred to as the “holy grail” of autonomous robotics research [\[7\]](#).

Most newer slam navigation algorithms are derived from the classical Bayesian probabilistic recursion. They model the system by a probabilistic density function (PDF) on map and agent location. At the discrete time step t the joint PDF for location X_t and map M_t is based on the history of (noisy) control inputs U , state X (usually map location in terms of translation and rotation from known location X_0), and exteroceptive measurements of the environment Z [\[34\]](#):

$$P_{t|t}(X_t, M_t | Z_{1:t}, U_{1:t-1}, X_0) \quad (2.25)$$

2 Theoretical Background

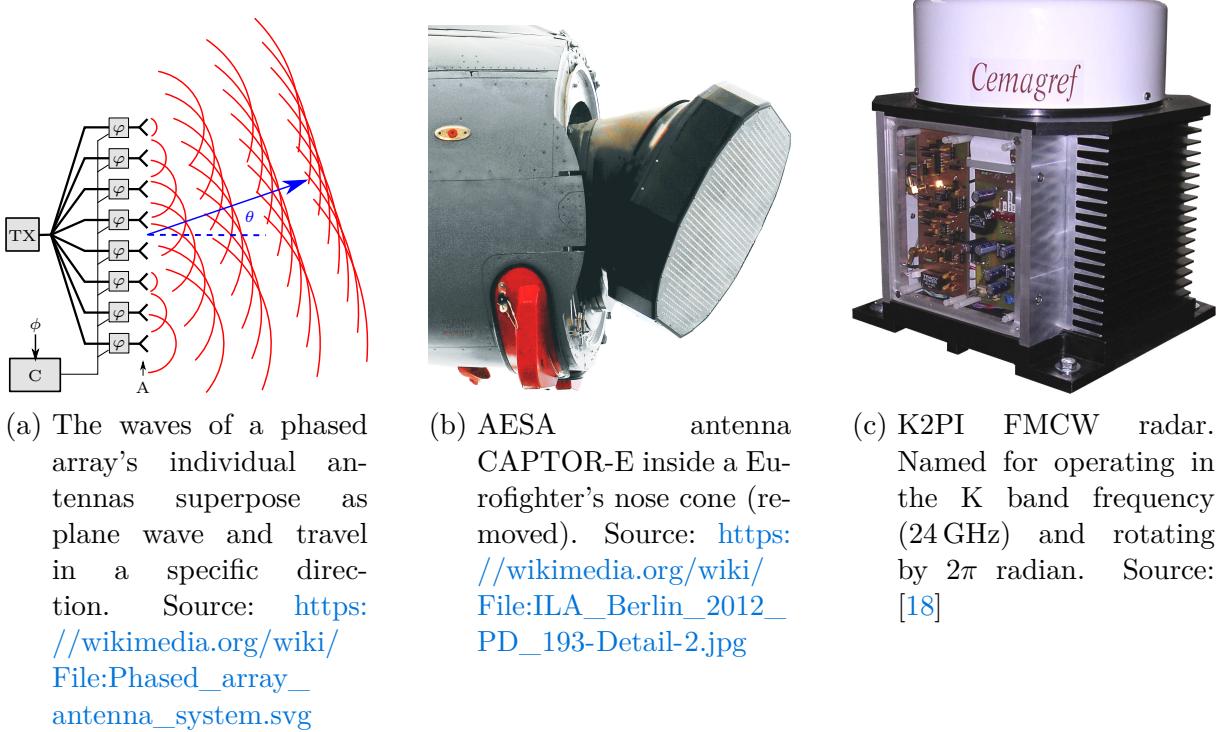


Figure 2.13: Scanning radars.

Recursive probabilistic solutions then first *predict* the joint state $P_{t|t-1}(X_t, M_t | Z_{1:t-1}, U_{1:t-1}, X_0)$ from control inputs and then *correct* the state with the observations, to get the posterior of equation 2.25.

Many current applications using slam acquire observation data with laser scanners or vision sensors (VSLAM), but this is not an inherent limitation of slam. In fact, any kind of sensor can be used as long as it delivers reproducible data where features can easily be re-detected from different locations.

For feature-based slam algorithms, representative landmarks are extracted from observation data. For multistatic radar sensors [19, 23, 35, 40] offer a solution that classifies walls, edges and corners from multipath signal propagation round trip times.

In [34], Adams, *et al.* explore robotic navigation and mapping with radar. Grid-based radar mapping is shown to have advantages over other slam sensors, because of its wider beam width and foliage penetration capability. Figure 2.14b shows superimposed radar and lidar scans of the scene in figure 2.14a. Radar tends to spatially blur scene information, but penetrates further than optical sensors (laser, vision) as it is less susceptible to occlusions.

Figure 2.15 shows occupancy gridmaps collected with a laser-based and a radar-based slam. The concrete walls of the moat in the upper left of the scene as well as foliage was captured much better in the radar gridmap.

2.4 Existing radar-based solutions for map building

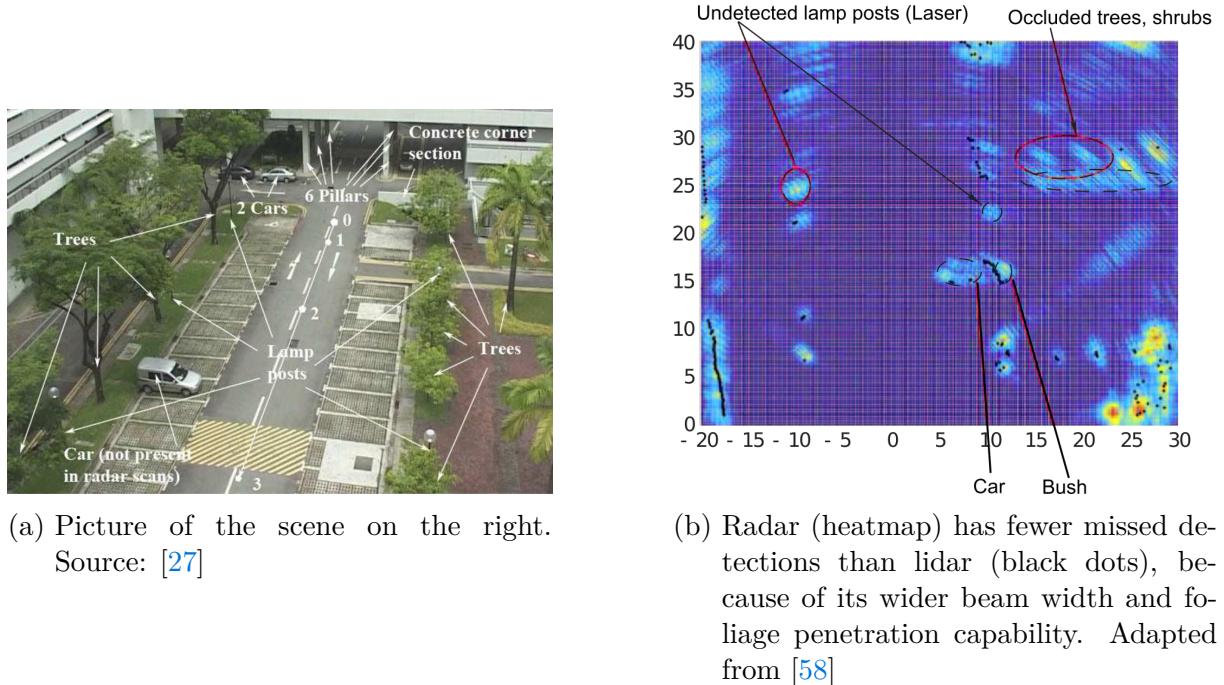


Figure 2.14: Superimposed radar and lidar plan position indicator display.

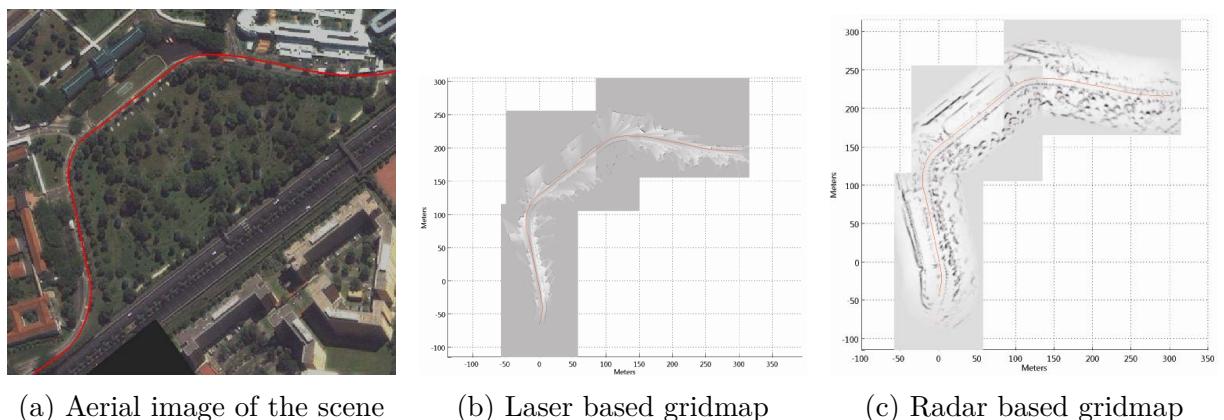


Figure 2.15: Comparison of radar and lidar occupancy gridmaps. Source: [58]

3 Novel Approach: Reprojection Mapping

3.1 Idea

A radar range reading reveals the distance to detected targets, but not what angle they are seen under. As explained in section 2.4.2, this can be solved by scanning the radar antenna to exploit beam directionality. There are however some drawbacks to scanned radars. The increased design complexity and cost of electronic scanning with phased arrays makes a sensor design more expensive. Passive phased arrays have a limited field of view. The motorized parts of mechanical scanning and active phased arrays not only increase cost compared to scanless antennas, but also introduce additional points of failure, higher maintenance requirements and increased power consumption.

This thesis proposes the reprojection method, with which a radar target's source can be determined without the need to scan the sensor. While the target distance is of course already available in the radar range data the target angle must be known in order to build or update a map of the environment. Under certain circumstances the target angle can be extracted from range readings with the reprojection method.

The method requires that the sensor moves at a known speed v_R through an otherwise static environment. Through the sensor's motion the distance to visible targets will then change over time, which causes a Doppler effect for the entire visible scene. Based on the deviation of a target's measured Doppler speed v_D from the known sensor movement speed v_R the magnitude of the angle between the movement vector and the line-of-sight to the same target can then be calculated.

3.2 Geometry for the Side-Facing 2D Case

Assume that the sensor's antenna is sensitive only to co-planar targets — this can be achieved with antennas that feature a fan-shaped beam pattern that has high sensitivity only around zero elevation. In the simplest case, the antenna is also sensitive only to targets on one side of the path of motion described by the sensor. Figure 3.1a depicts a robot (gray circle) with a radar sensor (triangle) moving at speed v_R . The three targets A, B, and C (colored circles) are detected. Targets A and C represent the edge cases: The

3 Novel Approach: Reprojection Mapping

radar is moving straight towards target A (i.e. $\alpha = 0$) and measures its Doppler speed as

$$v_D = v_R, \text{ for } \alpha = 0. \quad (3.1)$$

At the same time it just passes target C at the point of closest approach with $\alpha = \frac{\pi}{2}$. Target C's Doppler speed will be measured as

$$v_D = 0, \text{ for } \alpha = \frac{\pi}{2}. \quad (3.2)$$

Target B finally shows the relationship of v_D , v_R and α for all targets on one side of the radar's path, including the extremes in equations 3.1 and 3.2:

$$v_D = v_R \cos(\alpha), \text{ for } 0 \leq \alpha \leq \pi \quad (3.3)$$

This is of course also true for already passed targets, which will yield a negative Doppler speed $v_D < 0$. Simply rearranging equation 3.3 for α then gives the target's *reprojection angle*

$$\alpha = \arccos\left(\frac{v_D}{v_R}\right). \quad (3.4)$$

Note that equation 3.4 holds, because the scene must be static and hence $-v_R \leq v_D \leq v_R$.

3.3 Geometry for the General 2D Case

In the planar side-facing case, only the principal value of the inverse cosine in equation 3.4 is used. However, as soon as the radar antenna features angle sensitivities that allow the detection of targets on both sides of the motion path, there will be an angle ambiguity. This is the case for a sensor that is mounted forward-facing or that can detect targets in a field of view of over 180° . This case is practically very relevant as it is of high importance for an obstacle sensor to be sensitive to obstacles in its path rather than just objects located on the side.

Figure 3.1b visualizes this geometry: For target A the angle is still clear, because according to equation 3.1 $\alpha = 0$. Targets B and C however are located at angles α and $-\alpha$, which according to equation 3.3 causes the same Doppler speed. In this case, equation 3.4 will yield an ambiguous result.

It would be possible to resolve this ambiguity by tracking the targets while changing the robot's movement direction as shown in figure 3.2. If the robot's bearing is changed by β , the targets B and C appear at angles $\alpha + \beta$ and $\alpha - \beta$, respectively. The target whose Doppler speed changes to $v_D = v_R \cos(\alpha + \beta)$ is the one that the robot turned away from, and vice versa for the other target. The drawback is that newly detected targets can only be mapped after a change of bearing, which means that the robot has to drive in a zig-zag path to properly map its environment.

For multistatic radars there is however another solution to resolving the angle ambiguity: Using direction of arrival (DOA) information obtained from inter-antenna phase difference

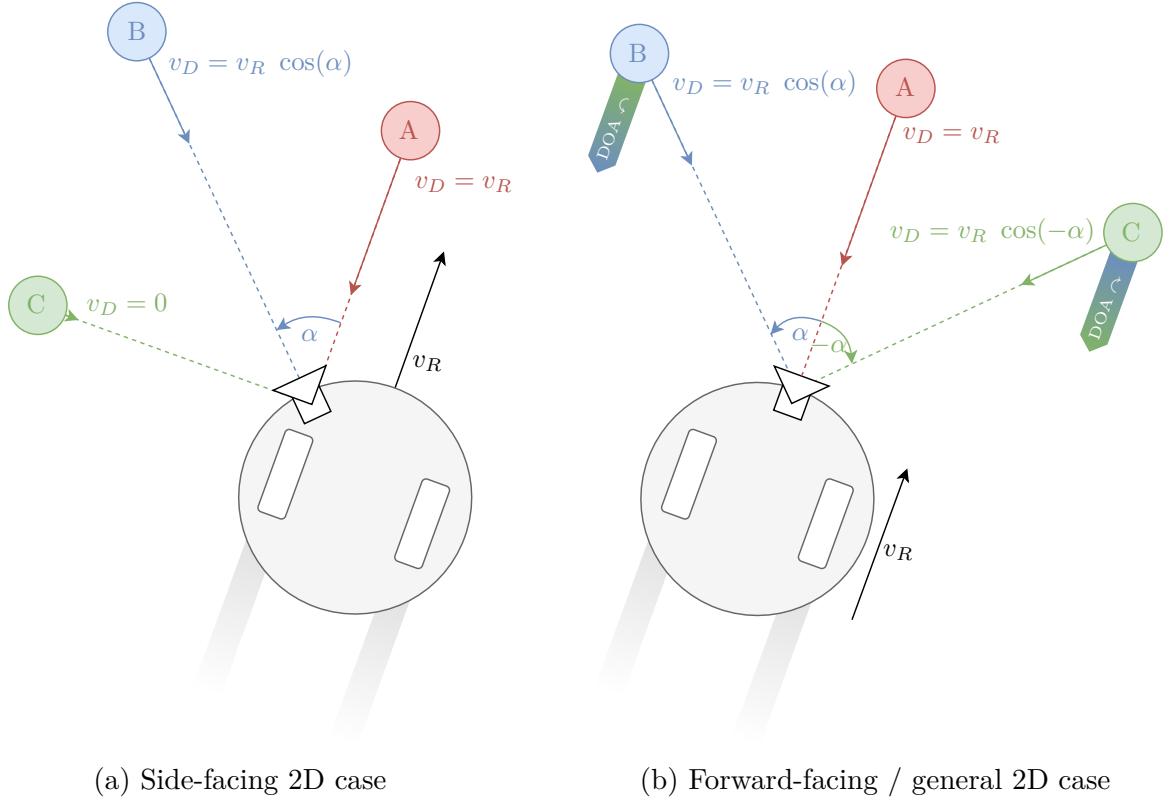


Figure 3.1: Reprojection geometries

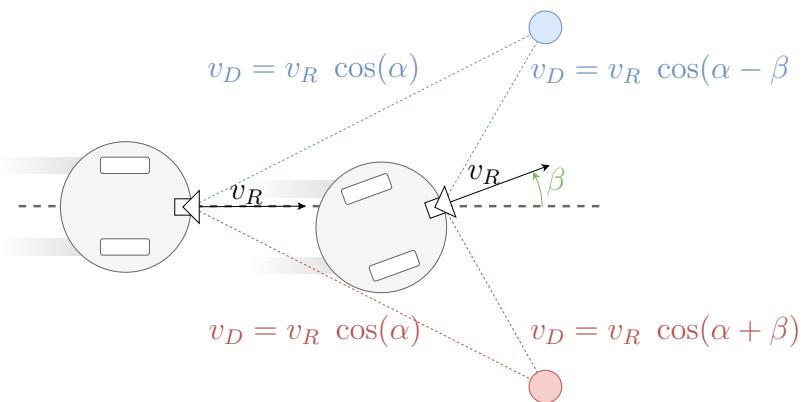


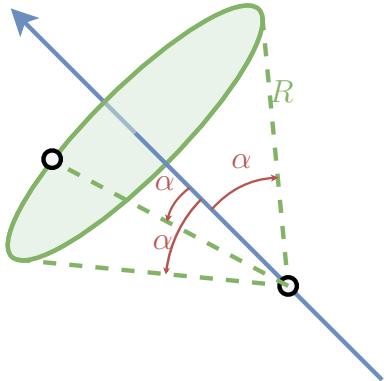
Figure 3.2: Ambiguity resolution through target tracking during turns

3 Novel Approach: Reprojection Mapping

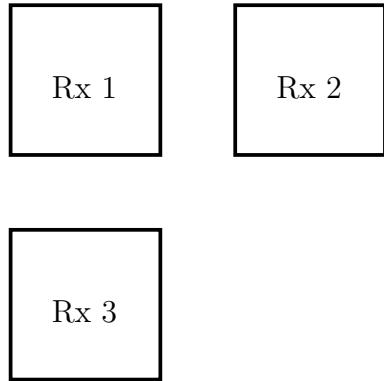
measurements (see section 2.2.4). Figure 3.1b shows that if the radar is mounted at a straight forward angle, a target's DOA indicates the side it is being passed on. If the radar is not facing straight ahead, the DOA value should be tracked so its gradient can be used instead — If a target's DOA is gradually shifting towards left (target B in figure 3.1b), it will pass the radar on the left side. On the other hand, if it is gradually shifting towards right, it will pass on the right side (target C in figure 3.1b).

3.4 Geometry for the General 3D Case

The geometry in section 3.3 can be further generalized in a three dimensional geometry. The 3D case is depicted in figure 3.3a. The model makes most sense for a robot that can traverse in all three dimensions, like a plane or drone. Hence the blue line in the figure can be called the sensor's flight path vector. Just like in the 2D case, when a static target is detected at range R , its radial speed v_D leads to a reprojection angle α . The difference is that the possible target location lies not on two points, but on the green circle with radius $R \sin(\alpha)$ whose origin lies $R \cos(\alpha)$ away on the flight path vector.



(a) A target detected at range R with a relative Doppler speed corresponding to re-projection angle α will be on a point on the circle (green) around the flight path (blue).



(b) Non-collinear triple receive antenna arrangement.

Figure 3.3: 3D geometry and ambiguity reduction through 2D-DOA.

This makes sense because it shows that the 2D case is a special case of the 3D case, where the possible target locations are reduced from the full circle to the intersection points of the circle and the floor plane on which the robot navigates.

The ambiguity could be reduced through a two-dimensional DOA estimation. This is possible if a sensor has three or more non-collinear receiving antennas. The simplest configuration would be the one shown in figure 3.3b, which allows horizontal DOA estimation from the phase difference between $Rx1$ and $Rx2$, and vertical DOA estimation between $Rx1$ and $Rx3$. However, DOA measurements can not resolve the target location single

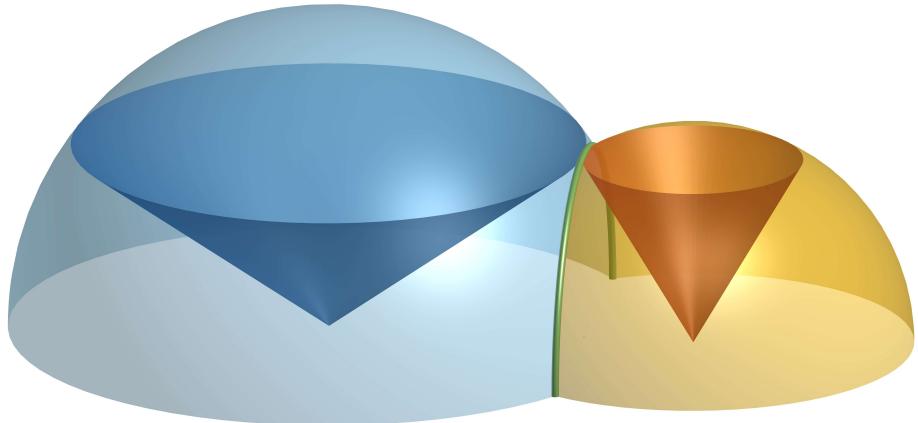


Figure 3.4: 3D rendering of a multi-sensor scenario: Two sensors are enough to single out a target’s location, while a trilateration approach would need a third sensor to narrow down the location from the intersection circle (green) of the trilateration spheres (orange and blue)

point like in the 2D case and it would be necessary to implement the target-tracking-during-change-of-motion-vector approach.

With this geometry, reprojection mapping could be used to build probabilistic 3D occupancy gridmaps like in [43], with vehicles moving in 3D space, like the TUM RCS’s Modular Airborne Real-Time Testbed (MART)¹⁾ that was also used in other research and publications [60].

3.5 Geometry for the General 3D Case with Two Sensors

The ambiguities in section 3.4 can also be eliminated by using more than one range sensor. In a 2D environment, this makes it possible to do regular trilateration, so reprojection mapping would not be necessary. In the 3D case however, two sensors would be enough for trilateration to pinpoint a target location — instead, the target would be localized on the intersection circle of two spheres around the sensors, each with a radius equal to the target range. With the addition of reprojection geometry, for each sensor a cone with an opening angle of twice the reprojection angle can be added. The cone’s base edge lies on the sphere. A target’s location is now known to be at the intersection of the two cones, which is a single point.

The method also helps when more than one target is visible and they need to be discerned from each other: while many trilateration spheres intersect in larger or smaller circles, the cone base edges touch (or, with added noise, are very close) only for the same target. 2D target location resolution is completely sufficient for mobile ground robots. Hence the

¹⁾ <https://www.rcs.ei.tum.de/forschung/mart/>

3 Novel Approach: Reprojection Mapping

rest of this thesis will focus on the side- and forward-facing 2D geometry described in section 3.3.

3.6 Doppler speed estimation

In order to estimate the reprojecion angle with equation 3.4, the target Doppler speed v_D must be measured. If the measurement is not precise, the reprojecion angle α will be imprecise and noisy. A map built with noisy observations will contain smeared-out targets or even false positive detections on the map.

With FMCW radar, a target's range and Doppler speed can be simultaneously registered. Especially at low speeds the speed resolution or accuracy may not be very high with this method. Doppler speed can instead be estimated from a target's change in range over time. The development of the peak gradient algorithm that accomplishes this with subsample resolution is detailed in section 4.5.

3.7 Map building

As soon as a target's range and angle are known, it can be mapped in relation to the radar's position. The radar position can be assumed at the map's origin for the first measurement; however the radar's translation and rotation relative to this start position needs to be known for subsequent measurements. This can easily be achieved with proprioceptive sensors such as inertial measurement units (IMUs) and encoder odometry. To compensate odometry drift in longer scans position corrections from absolute position sources should be integrated. Such a source is usually available from a magnetometer for heading, from global navigation satellite system (GNSS) sensors (e.g. GPS), MoCap camera systems or other ground truth systems [42]. It is also possible to run a slam algorithm or probabilistic localization like adaptive Monte Carlo localization (AMCL) for localization.

Building and updating of the map at a detected target location can either be in form of a binary target indicator, or with a probabilistic distribution. The latter can be taken from the target peak in the sensor's range profile.

The map can then be further processed with path planning algorithms such as Dijkstra, A*, D*, probabilistic road maps (PRM), and rapidly-exploring random trees (RRT) to allow intelligent path planning and obstacle avoidance for mobile robots [73].

3.8 Limitations

The reprojection method has some inherent limitations. The most obvious one is that since target angles are calculated from radial target speed, all objects must not have a

speed of their own, i.e. they must be static. This limitation however sounds more severe than it is in practice. On the one hand, robots often operate alone and without any other moving objects present, such as a vacuum robot cleaning the house while its owner is at work. On the other hand, if moving objects appear, such as a cat quickly escaping the vacuuming noise, they can be detected and tracked as soon as their radial speed exceeds the robot speed.

In 2D geometry, target echos must be limited to one plane and in the preceding explanation, this is assumed. If this is not the case, target elevation needs to be estimated in addition to target azimuth. With an antenna beam pattern that limits sensitivity to planar elevation, coplanarity of all targets can be assumed.

Another noteworthy issue is that real-world targets may not be adequately represented by a point target. However, especially if Doppler speed is estimated from peak gradient, any target visible in a range scan can only have one radial speed associated with it. If a long wall is visible over many range bins, the reprojection angle will be calculated from the speed of the peak point in the wall's echo. A map entry will be line-shaped with an angle corresponding to the point of strongest echo intensity, which is usually orthogonal to wall orientation, and with a length corresponding to detected peak width, which depends on the objects reflectivity at different incident angles. In effect, a wall will be built up on the map as the robot passes it, and the mapped wall thickness will be slightly influenced by how much of the wall reflects in a single range scan (usually a few centimeters, short solid red line in figure 3.5).

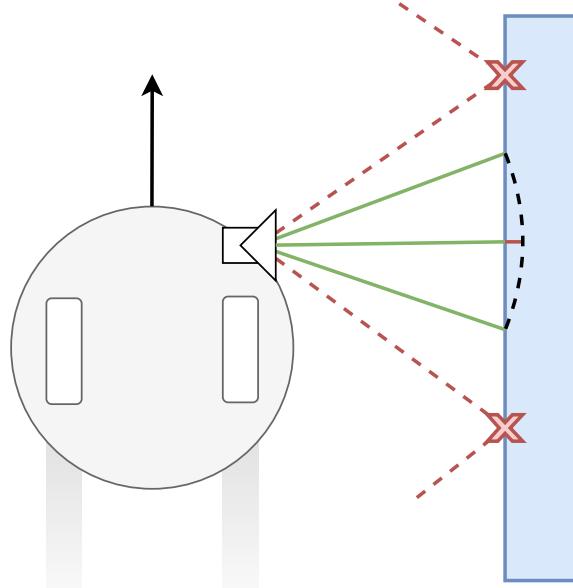


Figure 3.5: Walls reflect only at limited, material-dependent backscatter angles. They thus appear as point targets with a Doppler speed equal to the radar's motion component orthogonal to the wall.

Lastly, without direct Doppler measurement, only one target can be detected at any range

3 Novel Approach: Reprojection Mapping

bin. This limitation does not strongly affect practical operation, because situations where several targets are at the same range for more than a few seconds are very rare or can be easily avoided. For example, both walls of a corridor can be detected if the radar is not moving exactly in the middle of the corridor.

4 Implementation

The validity and performance of reprojection mapping is put to test with a proof of concept implementation in Matlab for a system based on an Omnidar RIC60-A mounted on a Kobuki robot platform.

4.1 Hardware Platform

4.1.1 Kobuki

The reprojection method requires that the radar is moved through the scene with a known position and velocity. The Kobuki robot platform (shown in figure 4.1a) was developed by Yujin Robot¹⁾, a Korean robotics and toy company. As base for the second Turtlebot, the reference design for the Robot Operating System (ROS), it not only provides a fairly good odometry system, but also allows to move the radar in a controlled way. Its shape and performance are very comparable to a real-world vacuum robot (it is actually based off Yujin's iClebo vacuum robot series), which makes the application relatively realistic.

4.1.2 ROS integration

The Robot Operating System (ROS)²⁾ is widely used in robotics research. It provides a framework of robotic hardware drivers, software libraries with state-of-the-art algorithms, and powerful developer tools for building robot applications. The Kobuki ROS package³⁾ is widely used and tested. With `rosbags`, the ROS system also provides a convenient way to record data for later playback and analysis.

4.1.3 Odroid

In the Kobuki platform, only an embedded microcontroller is integrated to drive the robot's hardware. ROS however needs an OS to run on and a computing platform with at least some performance. In this implementation, Hardkernel's Odroid XU4⁴⁾ (see

¹⁾ <http://kobuki.yujinrobot.com/>

²⁾ <http://www.ros.org/>

³⁾ <http://wiki.ros.org/kobuki>

⁴⁾ http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825

4 Implementation



(a) Yujin’s iClebo Kobuki robot platform.
Source: [41]



(b) Hardkernel Odroid XU4. Source:
https://wikimedia.org/wiki/File:Hardkernel_Odroid_XU4_Board.jpg

Figure 4.1: The supporting hardware

figure 4.1b) single-board computer provides plenty of power with its Samsung Exynos Cortex-A15/Cortex-A7 2 GHz Octa-core CPU paired with 2 GB ram. It runs Ubuntu Linux 14.04 with ROS Indigo Igloo.

4.1.4 Lidar Slam

A RoboPeak / Slamtec RPLidar A1⁵⁾ lidar sensor is mounted on the Kobuki base to enable laser-based slam. This is very useful to both provide odometry correction during reprojection mapping and to build a lidar slam occupancy gridmap that can then be compared with the radar reprojection map.

The slam algorithm used in the implementation is Google’s Cartographer [77] with the Cartographer ROS Integration⁶⁾. Cartographer ROS for TurtleBots⁷⁾ conveniently provides configuration files that work well with the Kobuki.

4.1.5 Radar sensor

One of the most important aspects of the implementation is the choice of radar sensor.

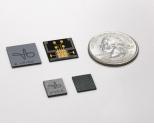
⁵⁾ <https://www.slamtec.com/en/Lidar/A1>

⁶⁾ https://github.com/googlecartographer/cartographer_ros

⁷⁾ https://github.com/googlecartographer/cartographer_turtlebot

Devkit list

There are quite a few short-range UWB FMCW radar modules available. They are all small enough to be integrated in a product of a size. Important features are high bandwidth to achieve high range resolution, good update rate, and easy configuration and integration of both hardware and software. Of course the sensor must also be available for sampling or purchase. The following table compares some promising solutions, roughly sorted by bandwidth. f_C and Δf denote the center frequency and bandwidth of the product, respectively. Note that the development kit (DK) price is usually much higher than the price of the radar chip sold for integration in products.

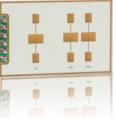
Product	Note	$f_C, \Delta f$	Antennas	DK Price	Picture
Omniradar RIC60A ⁸⁾	High bandwidth. Presentation at SoC 2015 [61]	60 GHz, 7 GHz	On-chip, 1 Tx, 2 Rx	\$4000	
Google / Infineon Soli ⁹⁾	Expected 2018. Sub-millimeter accuracy, running at over 10,000 frames per second [79]	60 GHz, 7 GHz	In-package, 2 Tx, 4 Rx	?	
Walabot Pro ¹⁰⁾	3D configuration. Slow update rate	6.8 GHz, 7 GHz	On-board, 9 Tx, 9 Rx	\$599	
Bosch Prototype	Two-port network analyzer used as core element in in-wall pipe detection	5.15 GHz, 6.7 GHz	External, 2 Tx/Rx	-	
Silicon Radar SiRad Simple ¹¹⁾	Has WiFi	122 GHz, 6.4 GHz	On-chip, 1 Tx, 1 Rx	?	
Anokiwave AWMF-0117 ¹²⁾		12.5 GHz, 4.5 GHz	On-chip, 1 Tx/Rx	?	

Continued on next page

4 Implementation

Product	Note	$f_C, \Delta f$	Antennas	DK Price	Picture
NXP Cocoon Radar ¹³⁾	Relatively small board. Presentation at FTF 2016 [82]	77 GHz, 4 GHz	On-board, 3 Tx, 4 Rx	?	
TimeDomain P440 ¹⁴⁾	Can operate as multistatic radar or UWB communication node	4 GHz, 1.7 GHz	External, 2 Tx/Rx	\$5000	
Novelda Xethru X4M03 ¹⁵⁾		8 GHz, 1.5 GHz	On-board, 1 Tx/Rx	\$399	
RFbeam MR2001_RD ¹⁶⁾		77 GHz, 1 GHz	On-board, 4 Tx, 6 Rx	?	
Inras 77 Ghz Radarbook ¹⁷⁾	Configurable FPGA processing chain	77 GHz, 1 GHz	On-board, 4 Tx, 8 Rx	\$7300	
Inras 24 Ghz Radarbook ¹⁸⁾		24 GHz, 250 MHz	On-board, 4 Tx, 4 Rx	\$7300	
Acconeer A1 ¹⁹⁾	Sub-mm accuracy	60 GHz, ?	On-chip, ?	?	
Infineon BGT24-RFB2412-EVAL ²⁰⁾		24 GHz, 250 MHz	On-board, 1 Tx, 2 Rx	\$1333	
IMST DK-sR-1200e ²¹⁾		24 GHz, 250 MHz	On-board, 1 Tx, 2 Rx	\$3333	

Continued on next page

Product	Note	$f_C, \Delta f$	Antennas	DK Price	Picture
InnoSenT IVS-565 ²²⁾		24 GHz, 250 MHz	On-board, 1 Tx, 2 Rx	?	
ST EVB-StradA431 ²³⁾	SMA connectors for internal signals	24 GHz, 250 MHz	External, 1 Tx, 3 Rx	?	
OmniPreSense OPS241-A ²⁴⁾	Arduino shield with BGT24LTR11	24 GHz, 80 MHz	On-board, 1 Tx/Rx	\$169	

The antenna solutions are diverse. Some products provide only one antenna for transceiving, while others have up to 18 multiple-in-multiple-out (MIMO) antennas. A sufficient number of transmit antennas enables electronic beamforming, while enough receiving antennas provide DOA estimation capabilities. Some products require external antennas (and hence some additional design efforts), others with antennas integrated in the copper layer or in ceramic packages on the printed circuit board (PCB). Very interesting are also the radar sensors with antennas integrated in the integrated circuit's (IC) package or even silicon die. They miniaturize the product and take a lot of RF design work off the integrating engineer.

⁸⁾ <https://www.omniradar.com/products/>

⁹⁾ <https://www.infineon.com/cms/en/product/promopages/soli/>

¹⁰⁾ <https://walabot.com/store/us/products/walabot-developer-pack.html>

¹¹⁾ http://www.siliconradar.de/evalkits_e.html

¹²⁾ <http://www.anokiwave.com/products/awmf-0117/index.html>

¹³⁾ Reuter2016

¹⁴⁾ <http://www.timedomain.com/products/pulson-440/>

¹⁵⁾ <https://www.xethru.com/xethru-development-platform.html>

¹⁶⁾ https://www.rfbeam.ch/files/products/26/downloads/ProductBrief_MR2001_RD.pdf

¹⁷⁾ <http://www.inras.at/en/products/radarbook.html>

¹⁸⁾ [dkradarbook](#)

¹⁹⁾ <http://www.acconeer.com/>

²⁰⁾ <https://www.infineon.com/dgdl/?fileId=5546d46259d9a4bf0159f9f1fa503f1d>

²¹⁾ <http://webshop.imst.de/dk-sr-1200e-development-platform-for-24-ghz-fmcw-radar-application.html>

²²⁾ http://www.innosent.de/fileadmin/media/dokumente/DATASHEETS_2016/Datenblatt_IVS-565.pdf

²³⁾ http://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/automotive-ic-eval-boards/evb-strada431.html

²⁴⁾ <https://www.omnipresense.com/product/ops241-a/>

4 Implementation

In the given time frame, the Walabot Pro, Omnidar RIC60-A and a proprietary Bosch prototype were available to be tested.

Bosch Radar

The prototype provided by Bosch Power Tools contains an RF core module which works as two-port network analyzer and is used in applications such as wallscanners for in-wall pipe detection. Getting it working in a Linux environment presented some challenges — After its Matlab driver was patched for cross-platform compatibility, it turned out that the on-board MCU's firmware had an incompatible protocol format. A newer version of the prototype that was later provided did work well under Windows, but by that time the decision to focus on Omnidar's chip was made and the newer driver was not cross-platform patched.

Vayyar Walabot

Vayyar²⁵⁾ is an Israeli startup²⁶⁾ that was founded in 2011. Coming from a medical background, they moved away from use cases such as breast cancer detection towards general 3D imaging in the consumer market with their Walabot sensor. Vayyar's Walabot Pro sensor uses an 18-antenna MIMO array for 3D radar imaging. Vayyar is very quiet about the technology and algorithms used in their product and even the nature of the data that the sensor sends.

In their Python API documentation²⁷⁾ they showcase the available sensor operation modes: 3D imaging, 2D imaging, object tracking, pipe detection and raw data (see figure 4.2).

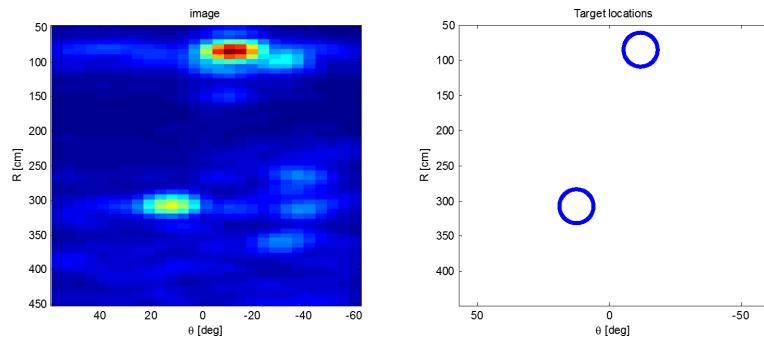


Figure 4.2: Vayyar's Walabot Pro sensor is claimed to have Target localization and tracking. Source: https://api.walabot.com/_features.html#_examples

The catch is that it is almost impossible to do imaging without background subtraction, which they do in all their examples. Granted, this works well in scenarios where the sensor

²⁵⁾ <https://www.vayyar.com/>

²⁶⁾ <https://www.crunchbase.com/organization/vayyar>

²⁷⁾ <https://api.walabot.com>

is at a fixed position or if the region of interest is very small, like in the pipe detection scenario. However, In the case of a robot-mounted sensor this does not provide great data quality, because not only a small part, but the whole scene changes when the robot moves.

At the time of writing, the only interesting published project using the Walabot is a fall detection scenario by Haider and Shaker [94], in which people can be localized at intersections of vertically and horizontally oriented heatmaps.

Static range test Figure 4.3a shows the signal from two Walabot antenna pairs as it records the scene in image figure 4.3b with metal can stacks at 0.5 m, 1 m, and 1.5 m. The signal is very stable over time and shows next to no noise. Unfortunately however it doesn't seem like the radar sensor can detect the metal cans very well. The higher frequency signal in figure 4.3a is the "raw signal" as reported by the Walabot sensor. It is however hardly believable that the signal was measured like this, as its base frequency is around 7 GHz. It was not possible to get any useful information from Vayyar's technical support regarding this. A more interesting data source is the envelope signal, which can easily be constructed from the analytic signal, using the `scipy.signal` package's Hilbert transform.

Another problem with the data is that the peaks of the envelope jitter in range. This can be fixed by combination with another oddity: The last 180 samples rise very strongly in magnitude. If they are cut and prepended to the first sample, they match up perfectly. Peaks can then be detected in the signal (represented by the dots figure 4.3a) and the range set to zero at the first peak. This eliminates the range jitter completely. The reason is that the first peak is caused by transmit antenna crosstalk and can thus be used as a timing reference point.

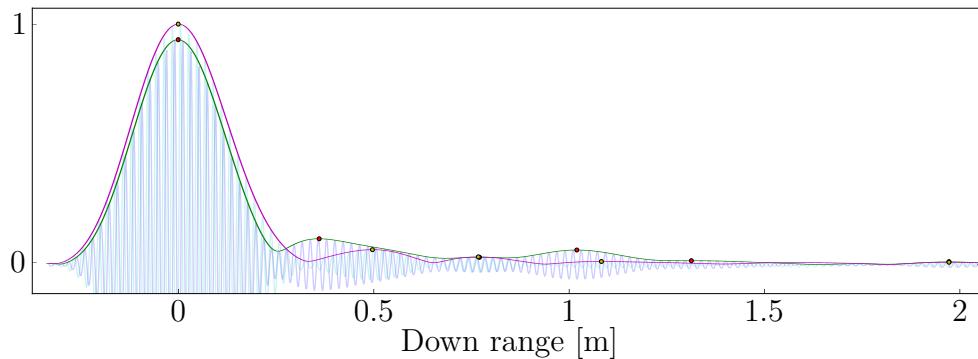
Dynamic range test Waving hands in front of the sensor did give a change in signal, but it was difficult to interpret the data conclusively. To objectively test the sensor's response, an aluminum plate that gave a strong echo signal was taped to the Kobuki robot. The robot was then driven with a constant speed away from the sensor and then towards the sensor as pictured in figure 4.4a.

The sensor was sampled at a constant frequency in raw data mode. The analytic signal magnitude of the range scans is stacked at the right end of the matrix displayed in figure 4.4b.

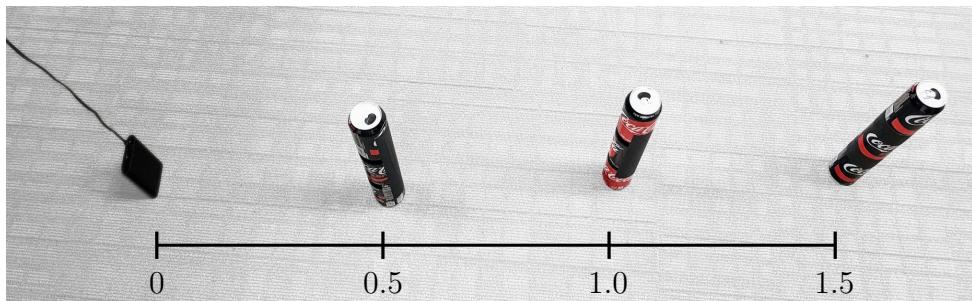
The figure shows that the Walabot has problems with what looks like standing waves. The great amount of background signal is also visible. Because of its static nature, this can easily removed for a fixed radar. As some Walabot reviewers have noticed [85] this background signal changes heavily and seemingly random when the sensor is moved. This makes the signal processing very difficult.

Walabot advertises object detection capabilities. The catch with this mode is however

4 Implementation



(a) Normalized echo intensity of range profile measurement with two antenna pairs. After the transmit peak at $t = 0$, three peaks corresponding to the three can targets are vaguely perceptible.

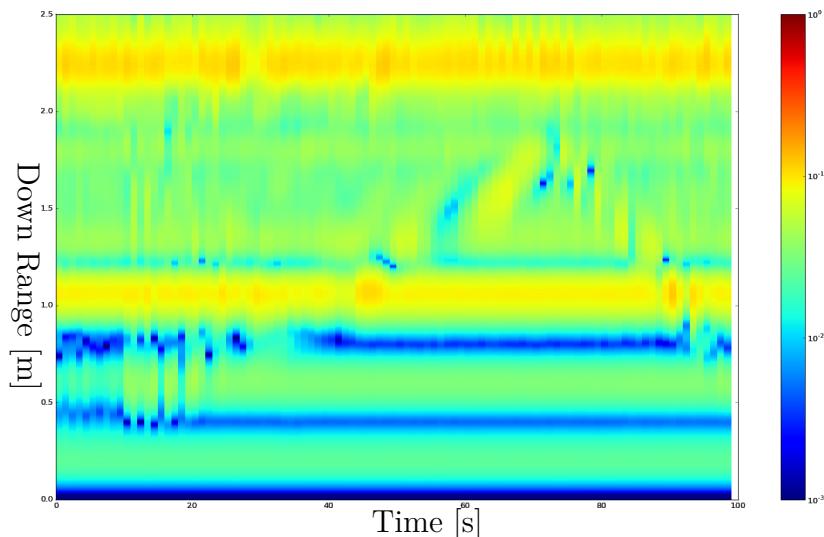


(b) Setup with three towers of cans in front of the Walabot sensor (scale: [m])

Figure 4.3: Static range test with the Walabot sensor



(a) Setup with a reflective metal plate mounted on the robot



(b) Range profile (y -axis) over time (x -axis) while the robot was moved at constant speed away and towards the Walabot sensor

Figure 4.4: Dynamic range test with the Walabot sensor

that the number of objects to be detected must be configured as a fixed parameter first.

Omniradar RIC60-A

With its good range resolution and a good idea of its capabilities from [74], this sensor promised good results. It will provide the basis for the proof of concept implementation, so it receives a more detailed look.

Founded in 2010, Omnidaradar²⁸⁾ is a Dutch startup²⁹⁾ that claims to be the first to integrate a complete 60 GHz radar including antennas and analog to digital conversion in one chip.

With the RIC60-A they offer a Radar Development Kit (RDK) that gives 7 GHz of bandwidth on two receiving antennas. An Altera Cyclone IV FPGA handles the signal acquisition and communication. Figure 4.5 shows the radar IC and how the three antennas are integrated in silicon.

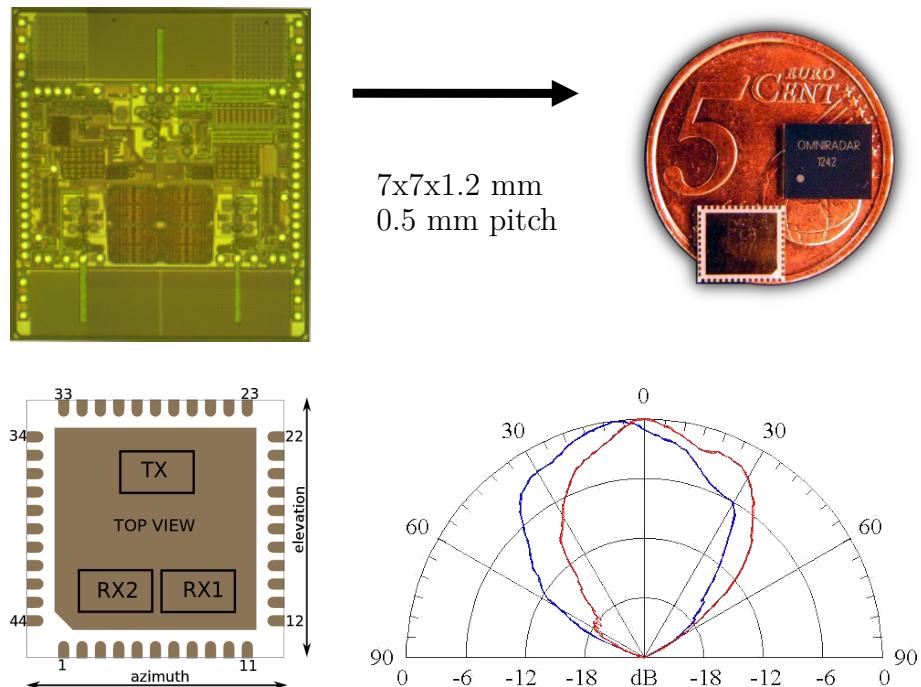


Figure 4.5: Decapped Omnidaradar IC (top left); five cent coin as size reference (top right); schematic with antenna locations (bottom left); and antenna directivity pattern of Omnidaradar's RIC60-A (bottom right). Adapted from [61] p.9

The radar sensor's beam is fan shaped, which means it is fairly sensitive over a wide angle in azimuth direction, but relatively focused in elevation. This makes it a very good candidate for the radar reprojection, as targets can be seen from the robot in a wide field of view, but floor and ceiling reflections are kept at a minimum. Of course the sensor can

²⁸⁾ <https://www.omnidaradar.com/>

²⁹⁾ <https://www.crunchbase.com/organization/omnidaradar>

4 Implementation

also be rotated. Omnidaradar also supplies a horn-like extension for the sensor board, which forms the radar sensitivity into a pencil-shaped beam that is very focused at a narrow field of view.

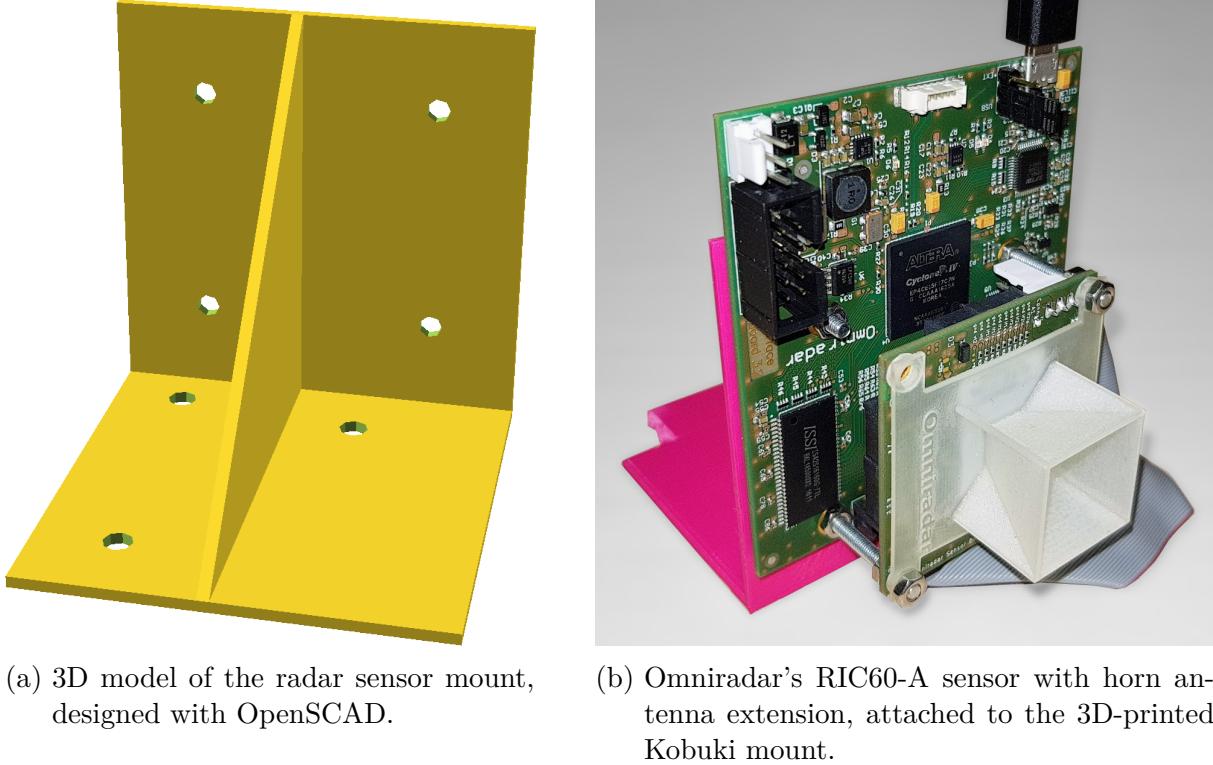


Figure 4.6: Radar sensor on mount and model of the mount.

Radar mount A 3D-printed part makes sure that the radar sensor is firmly mounted on the robot as it explores its environment. The part was designed in OpenSCAD³⁰⁾ (rendering in section 4.1.5) and printed on a Dremel 3D printer³¹⁾. The bottom mount hole positions were extracted from the mechanical drawings of the Kobuki Base [37]; the side holes from the Altium layer document of the version 3.2 Omnidaradar interface board [55].

When rotated to face to the left side of the robot, the RPLidar mount was in the way, so small parts of the print are clipped off in section 4.1.5.

Two screws hold the Omnidaradar interface board to the mount. The other two mounting holes in the interface board hold the Omnidaradar sensor board. The horn extension can be affixed to these screws as well. If a different sensor orientation is necessary, the sensor board can be rotated by 90°, thanks to the symmetric layout.

³⁰⁾ <http://www.openscad.org/>

³¹⁾ <https://3dprinter.dremel.com/>

Doppler sensitivity The Omnidaradar FMCW radar is not sensitive enough to use the Doppler speed directly. The following example illustrates this.

The RIC60A has a sensitivity of 400 Hz/(m/s). A target with a Doppler speed of 0.02 m/s (A low speed at which the Kobuki robot still moves continuously and without jerking) will cause a frequency spike with a shift of 8 Hz in the FMCW beat frequency.

The speed resolution capability is inversely proportional to the measurement or acquisition time. A 10 ms long acquisition gives a 100 Hz frequency resolution, or a speed resolution of 0.9 km/h (or 0.25 m/s).

The sampling frequency,

$$F_s = 25 \text{ MHz}$$

and RIC60A Doppler sensitivity,

$$S_D = 400 \text{ Hz}/(\text{m/s})$$

are constant values of the Omnidaradar RDK. Given a chirp duration of

$$T_{chirp} = 2.5 \text{ ms}$$

there are

$$N_s = \lfloor F_s T_{chirp} \rfloor = 62\,500 \text{ samples}$$

available for every sweep, and hence

$$N_r = \left\lfloor \frac{N_s}{2} \right\rfloor = 31\,250 \text{ samples}$$

per up/downsweep. With FFT frequency bin width

$$dF = \frac{F_s}{N_r - 1} = 800 \text{ Hz}$$

this yields a Doppler speed resolution of

$$\frac{dF}{S_D} = 2 \text{ m/s}$$

Even with subsample peak interpolation the accuracy will not be very good and targets will be reprojected at imprecise angles.

It would be possible to use higher precision equipment. But another solution is to track the movement of target peaks in the range profile, using the peak gradient algorithm described in section 4.5.

4.1.6 Optimal chirp time configuration

The chirp duration T_{chirp} is configurable and has an effect on how the raw range profile data will look like. Next to SNR of raw data, an important measure is chirp efficiency, which influences effectiveness of raw data smoothing (because more data allows better smoothing) and hence SNR of processed data. The chirp efficiency η is defined as the percentage of time spent on actual measurement:

$$\eta = \frac{n_{chirp} t_{chirp}}{t_{msg}} \quad (4.1)$$

with number of consecutive sweeps n_{chirp} , chirp length t_{chirp} , and the time between the start of one respective measurement and the next, t_{msg} . Figure 4.7 shows how chirp efficiency increases with chirp time.

Very short duration (< 2 ms) Will have very low SNR and incur considerable processing overhead and hence chirp efficiency. It is hard to detect targets in this mode.

Short duration (2 ms to 5 ms) Will have acceptable SNR, and are more efficient with respect to overhead.

Medium duration (5 ms to 15 ms) Empirically found to be the optimal range.

Long duration (15 ms to 20 ms) Good SNR and not a lot of overhead. However, some radar signal packets are glitched and unusable. This seems to be a bug in the RIC60-A/FPGA communication that occurs over a certain amount of bandwidth. The probability of glitches occurring increases with the amount of requested data.

Very long duration (> 20 ms) These large amounts of data require the Omnidaradar driver to be patched on Linux so as not to freeze when chirps longer than 20 ms are requested. Even with the patched driver the RDK's FPGA firmware is not very reliable at sending large volumes of data at once and corrupts packet headers or aborts transmissions intermittently. At higher robot speeds, target peaks also get blurred over several range bins as they move in range over time. Less intense target echos are more difficult to detect then.

Figure 4.8 shows that the chirp length has an effect on accuracy and resolution. Short chirp times lead to a higher noise floor, but exhibit less variance, i.e. a narrower band of frequencies. Longer measurement times have a lower noise floor but have more outliers. Attaching the horn extension generally yields a higher SNR because the radar antenna beam shape is focused on a smaller field of view.

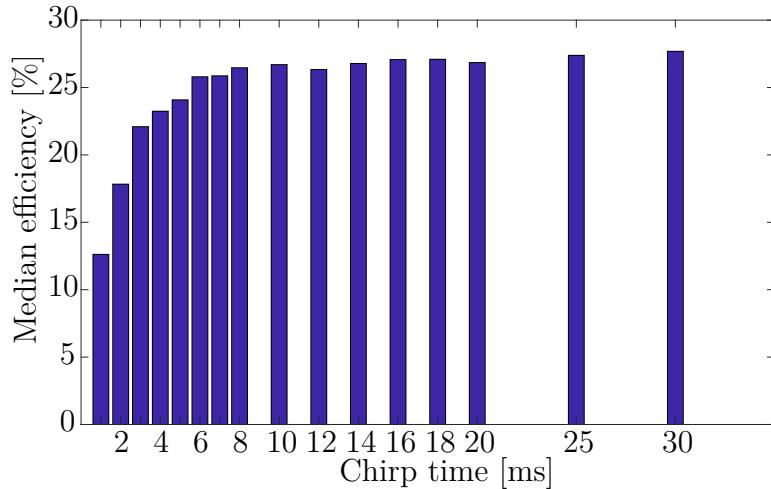


Figure 4.7: Chirp efficiency η (measurement time per time spent) for various chirp lengths.
For this data, $n_{chirp} = 2$.

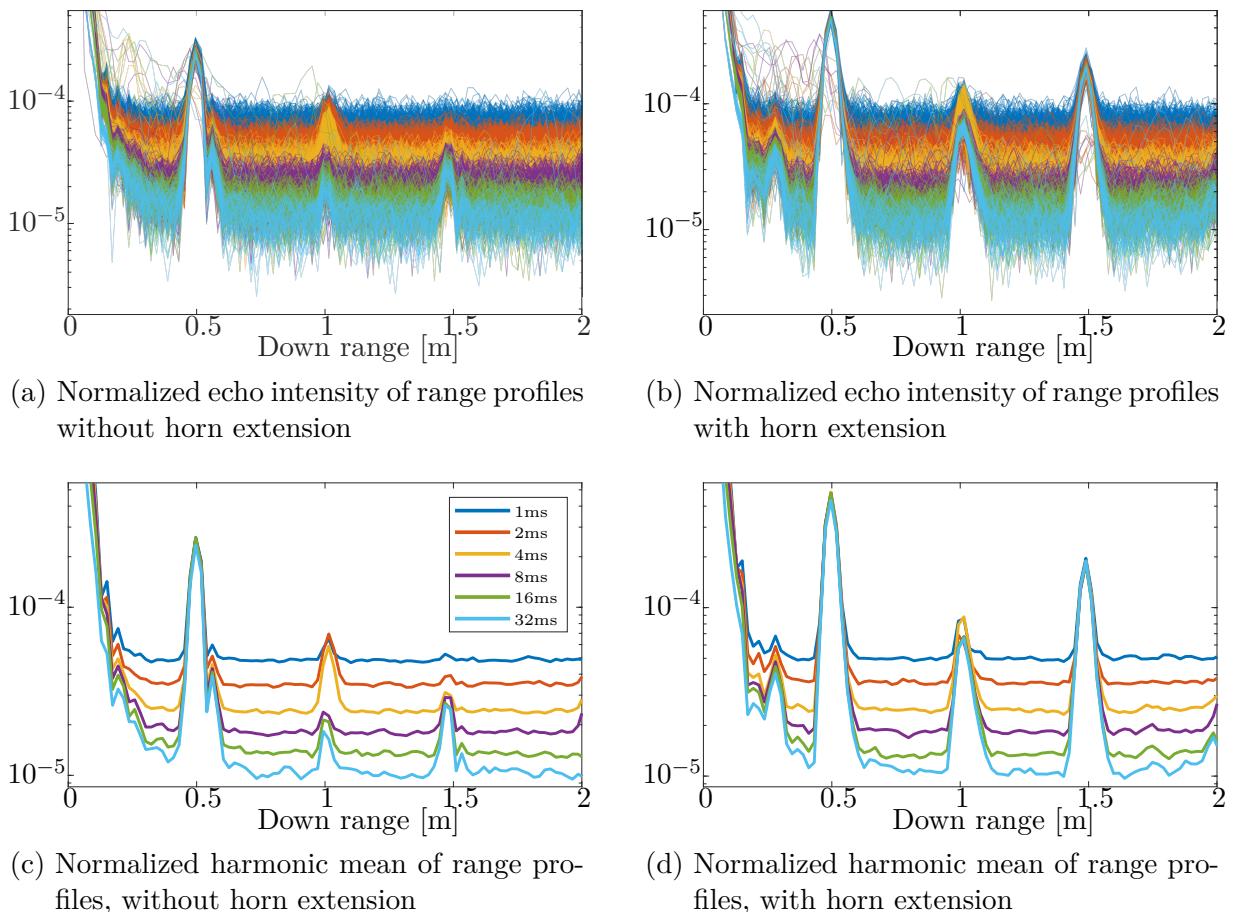


Figure 4.8: Effect of chirp length on SNR; with and without Horn extension. The range profiles depict the same scene and are recorded over 10 s each.

4.2 Software Implementation Overview

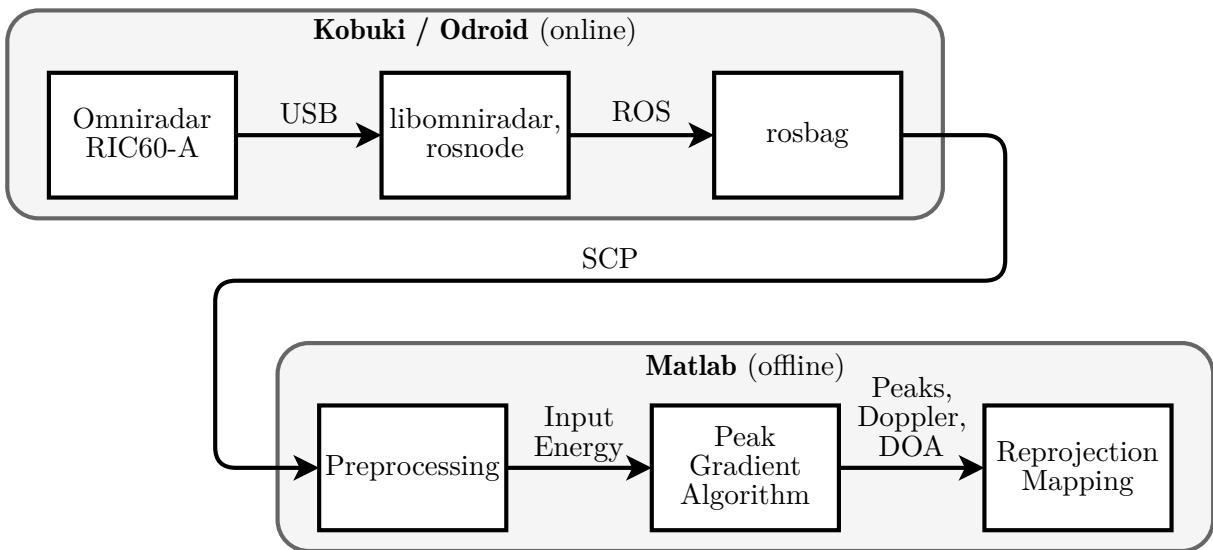


Figure 4.9: Schematic overview of the information flow from the hardware to the reprojection map

Figure 4.9 shows the flow of information in the presented hardware/software implementation. The top part in the diagram represents the online part, which comprises the components that are necessary to perform a radar scan: Radar data is gathered from the hardware setup described in section 4.1 while the `/omniradar_node` rosnode (section 4.3.2) uses `libomniradar` (section 4.3.1) to read in the raw data from the radar sensor. The rosmessages transmitted from the node are then stored in a `rosbag`. This `rosbag` is later copied over the network to a machine running the offline parts of the implementation. In Matlab, the data is preprocessed (section 4.4), Doppler speeds are estimated (section 4.5) and finally the radar reprojection map is constructed (section 4.7).

4.3 Driver

4.3.1 Omniradar ROS driver

The Omniradar RIC60A Radar Development Kit comes with a precompiled Matlab MEX driver library, which works well in a Windows OS on x86-based computers with a Matlab installation. An early goal was however to have the robot carry the radar module around wireless which would only have been possible with a complete Windows laptop mounted on the Kobuki platform. However, Omniradar was kind enough to provide the driver sources under an NDA agreement. This allowed recompiling the MEX driver for Linux systems. There were still some issues with FTDI's D2XX serial communication library that had to be fixed for Linux systems. One challenge with the D2XX driver was that

Ubuntu automatically loads the regular FTDI serial IO driver, `ftdi_sio`. This was solved by unbinding the driver using a set of udev rules³²⁾. Another issue was that due to a bug in the D2XX implementation, the Omnidar driver would freeze when more than 2 MB of data (equivalent to a 20 ms FMCW chirp) were requested. After a lot of debugging, this could be solved by never requesting a bigger amount of data than was already available in the D2XX buffer.

C++ bindings and library

Since Matlab does not run on Arm arch processors such as the Odroid on the Kobuki robot, a new set of platform-independent C++ bindings was added to the driver. The C++ bindings serve the same purpose as the Matlab bindings.

To include the library, some files need to be installed or pointed to by the binary that needs to use it. They are available in precompiled form from <https://github.com/lalten/libomnidar>.

File	Default install destination	Purpose
<code>omnidar.h</code>	<code>/usr/local/include</code>	Library header file
<code>libomnidar.so</code>	<code>/usr/local/lib</code>	Dynamically linked shared object
<code>51-omnidar.rules</code> , <code>52-omnidar.rules</code>	<code>/etc/udev/rules.d</code>	Udev rules to unbind <code>ftdi_sio</code>

Table 4.3: File installation destinations of libomnidar

Appendix A contains information on how to build, install and integrate the library. It offers the same functions as Omnidar's Matlab driver, with two differences. The optional device index number is 0-based instead of 1-based and the `AcquireEcho` functions return (a shared pointer to) packed data instead of unpacked data for performance reasons. With the

```
static std::shared_ptr< std::vector< std::vector< uint8_t > > >
    demultiplex(std::vector<uint32_t> &packed_data);
```

function, the library offers an easy way to demultiplex the packed data array into a (shared pointer to a) vector of four vectors - one for each echo signal of the I/Q channel of left and right receiving antennas.

Useage is simple as the driver follows the RAII principle. Allocation of an object of type `Omnidar` causes the library to fully initialize the RIC60A RDK. After that, the configuration string and the VCO tuning curve should be set. The VCO tuning curve is responsible for predistorting the (non-linear) VCO in a way that yields a linear frequency sweep. Omnidar provides Matlab scripts to measure the sensor's VCO tuning curve.

³²⁾ <https://stackoverflow.com/questions/44529376>

4 Implementation

Appendix A.1 contains information on how to convert the Matlab script results into something that can be used in a C++ application.

`libomniradar` enables the development of a C++ ROS node that handles communication with the radar module on the Linux/Arm based Odroid. The files comprising `libomniradar` are publicly available for integration in projects using the RIC60 sensor at <https://github.com/lalten/libomniradar>.

4.3.2 ROS node

A new `omniradar` ROS package was developed to support the use of the Omnidar sensor within the ROS environment. A set of `roslaunch` launchfiles comes with the package that support the startup of the Kobuki robot in teleoperation mode, optionally together with lidar-based Cartographer slam, AMCL localization and an Astra Orbbee RGBD camera. A simple version to start the only the node to get radar data would be:

```
<launch>
  <node pkg= "omniradar" type= "omniradar_node" name="omniradar_node"
    output="screen">
    <param name="n_sweeps" value="1" />
    <param name="t_sweep"  value="5" />
  </node>
</launch>
```

The node sends out the ROS topic `/omniradar_node/radar_raw` of the custom `RadarEcho` type, which is defined in `RadarEcho.msg` as

```
Header header
string ric_config
uint8 n_sweeps
float64 t_sweep
uint32[] packed_echo
```

Early versions of the driver unpacked the radar echo bitstream inside the node and were able to use standard ROS message types like the `std_msgs/ByteMultiArray` message. However, sending out the packed bitstream proved to be much more efficient in terms of chirp efficiency η . The custom message type also allows to send out the configuration (RIC configuration string, number and length of FMCW sweeps) that was used to attain the message's echo. The message's timestamp and sequential ID is contained in the regular `std_msgs/Header`. Figure 4.10 shows the `/radar_raw` rostopic in a node graph of the rosnode.

While parameters from the ROS parameter server (as configured in the launch) are respected, the node also offers a dynamic reconfigure server to change RIC configuration string, number of sweeps and length of sweep on the fly without the need to restart the node (see figure 4.11).

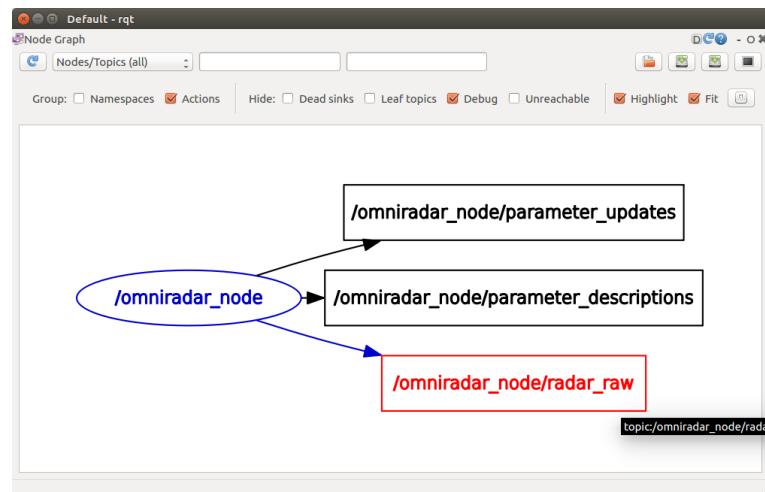


Figure 4.10: Omniradar rosnode node graph with leaf topics

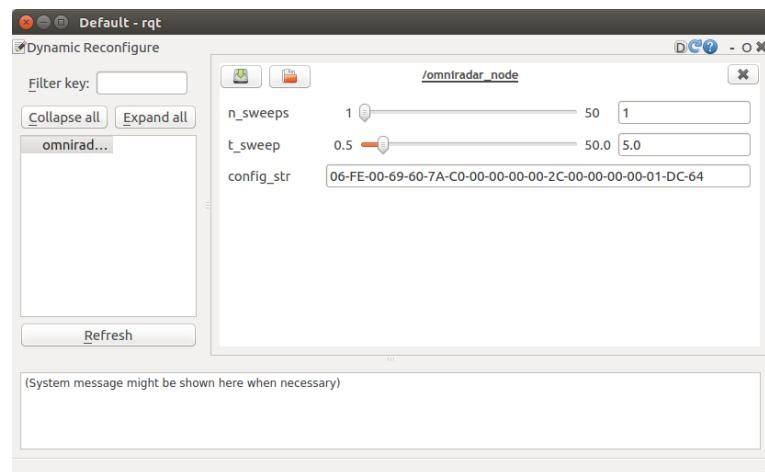


Figure 4.11: Dynamic reconfigure options for the Omniradar rosnode

4 Implementation

The core of the node is a while loop that continuously triggers radar echo acquisition and copies the result into a new `omniradar::RadarEcho` message. The radar sensor update rate could be increased by offloading the message assembly and data copying into a C++11 lambda thread that is immediately detached.

```
std::lock_guard<std::mutex> lock_rdk(mtx_rdk);
auto t_echo = ros::Time::now();
auto p_echo = rdk->AcquireEcho(msg.n_sweeps);

std::thread t (
    [] () {
        std::lock_guard<std::mutex> lock_msg(mtx_msg);
        msg.header.stamp = t_echo;
        msg.packed_echo.resize(p_echo->size());
        std::copy(p_echo->begin(), p_echo->end(), msg.packed_echo.begin());
        pub.publish(msg);
        msg.header.seq++;
    }
);
t.detach();
```

The multithreading approach lets the node use between 100 % and 140 % CPU on the 2 GHz octa-core Odroid (observed with `htop`).

Like `libomniradar`, the rospackage providing the `omniradar` rosnode is available at <https://github.com/lalten/omniradar>.

4.3.3 Matlab interface

Matlab (release R2017a) was chosen as implementation platform and language because it allows quick prototyping, provides relatively easy visualization, and, with the Robotics Toolbox, supports many ROS features.

It is necessary to install custom message support with the `roboticsAddon`³³⁾ and to generate³⁴⁾ some files to read in rosbags with the custom `RadarEcho` type messages.

³³⁾ <https://www.mathworks.com/help/robotics/ref/roboticsaddons.html>

³⁴⁾ <https://www.mathworks.com/help/robotics/ug/create-custom-messages-from-ros-package.html>

4.4 Data Preprocessing

4.4.1 Rosbag to Matlab

Before a target's radar echo can make its way to the map, it passes several stages. In this proof of concept implementation, the map building is done offline, so all data coming out of the omniradar ROS node is recorded in a `rosbag` for later analysis. This is a good solution because it (1) allows working even without access to hardware, (2) allows quick iteration and prototyping with scripted languages such as Matlab, which are not capable of receiving at real-time speeds, and (3) allows replaying any situation in which an algorithm fails, so that it can be tested and improved until the situation can be handled.

The rosbags can be replayed with `rosbag play` and received in a subscribing `rosnode` in Matlab. However the replay speed needed to be slowed down by a factor of ten, or Matlab will not be able to keep up and will lose some messages. This also the reason why a live system where rosmessages are sent via network to a computer running Matlab is not feasible. A better solution is to read the rosbag files directly into Matlab, using its `readMessages(bag,rows)`³⁵⁾. The system quickly runs into memory problems when reading even a moderately sized bag (e.g. 400 MB) at once, because the Java heap space is limited by default. It is not only safer (no `java.lang.OutOfMemoryError: GC overhead limit exceeded` crashes) to read in smaller chunks, but also allows to display a `waitbar` that shows that the program didn't freeze, but just takes a while to read in the bag. The described reading in is handled in the `radar_bag2array` function:

```
function [ radar_data ] = radar_bag2array( bag_filename )
```

The messages inside the read bags need to be processed first. If a `/tf` topic exists in the bag, `compensate_map_odom_tf(bag_tf)` extracts the `map-odom` transform as specified in REP 105³⁶⁾. This is the information from the Cartographer slam that corrects odometry offsets and drifts.

Odometry messages from the `/odom` topic are read in next. Some messages have unchanged values with timestamps that lie very close to the previous message. These are removed, because they cause problems in the later interpolation stages.

Next, the actual radar echo messages are read from the bag's `/omniradar_node/radar_raw` topic. The messages still contain the packed radar data. The function

```
function [ ch1I, ch1Q, ch2I, ch2Q ] = radar_unpack_data( packed_echo )
```

takes care of the unpacking by reordering bit-interleaved 32 bit words into the eight sigma delta ADC samples for four channels (Rx1/2, I/Q). This function is called very often, and the bit operations take a very long time in Matlab code. Performance was greatly

³⁵⁾ <https://www.mathworks.com/help/robotics/ref/readmessages.html>

³⁶⁾ <http://www.ros.org/reps/rep-0105.html>

4 Implementation

increased after using Matlab Coder to generate C/C++ code embedded in a Matlab-callable MEX file. Once this is done, the range profile is extracted from the time-domain beat signal with the `radar_echo2range` function,

```
function [down_range, ranges_1_up, ranges_2_up, ranges_1_dn, ranges_2_dn]
    = radar_echo2range( [ ch1I, ch1Q, ch2I, ch2Q ] , config )
```

which first corrects ADC gain using the formula $echo_V = 4.8 \text{ V} \cdot echo_{Raw} - 1.2 \text{ V}$ and subsequently applies the Fourier transform on the Blackman-windowed time domain data.

The global (i.e. relative to map origin) radar position is then calculated from the mapodom transform together with the odometry messages and a constant offset representing the translation from odometry node to radar sensor. This position vector is then linearly interpolated at the timestamps of the radar messages.

Lastly, the `radar_data` matrix is assembled as a vector of structs containing the basic information of each individual range scan, including sweep time, position, cross range mileage. This information is saved into `.mat` files for quick reloading.

4.4.2 Raw Data Smoothing

A single range reading by itself will usually be relatively noisy. One solution to getting cleaner range data with higher SNR is oversampling. It is possible to use a moving average over a certain accumulation distance to achieve this. However, the number of raw samples is quite high and processing each sample takes a considerable amount of time (around the same time span as recorded data duration). It is better to make use of binning, with bins the width of the accumulation distance. All samples in one bin are averaged to represent that bin's value. This greatly improves processing time (to less than a second for some minutes of recorded data).

There are several ways to implement the averaging, and they can be quantitatively compared. All of them improve the SNR, because the noise is statistically mostly uncorrelated to the signal. While peak intensities don't vary much (because there is less noise in the signal), some averaging methods are better than others at decreasing the signal level in troughs. The difference can be expressed by the root means square (RMS) of the difference to the best signal, which is acquired with the harmonic mean. Table table 4.4 compares the three Pythagorean means (arithmetic, geometric, and harmonic), a trimmed mean and the median with this metric. The means and RMS are defined as follows.

If the arithmetic mean is defined as

$$\text{mean}(x_{i=1,2,\dots,n}) = \frac{1}{n} \sum_{i=1}^n x_i = \frac{x_1 + x_2 + \dots + x_n}{n} \quad (4.2)$$

then the RMS of a signal x_i is

$$\text{RMS}(x_{i=1,2,\dots,n}) = \sqrt{\text{mean}(x_i^2)} \quad (4.3)$$

4.4 Data Preprocessing

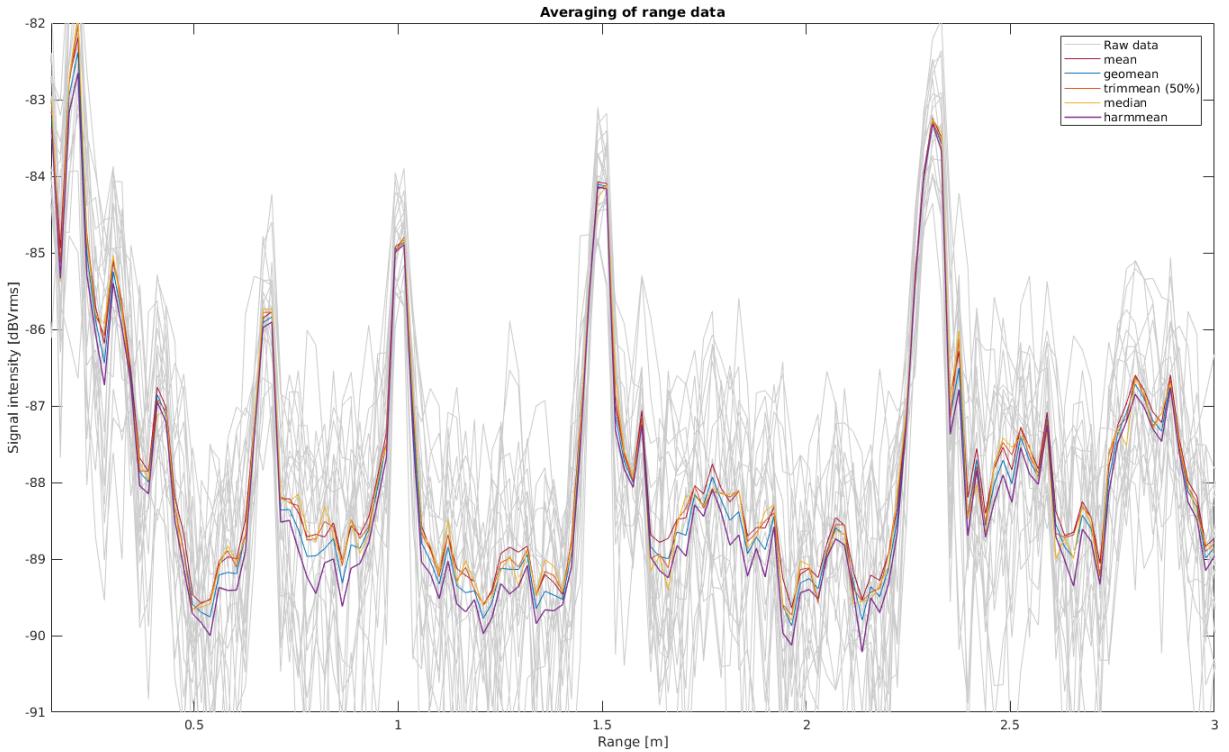


Figure 4.12: Range profile quality comparison with different averaging functions. Higher peak-to-trough ratio is better.

The harmonic mean is defined as

$$\text{hmean}(x_{i=1,2,\dots,n}) = \frac{1}{\text{mean}(x_i^{-1})} = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}} = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}} \quad (4.4)$$

The geometric mean is defined as

$$\text{gmean}(x_{i=1,2,\dots,n}) = \left(\prod_{i=1}^n x_i \right)^{\frac{1}{n}} = \sqrt[n]{x_1 x_2 \cdots x_n}. \quad (4.5)$$

In a set of n data points, the 50 % trimmed mean is defined as the arithmetic mean of the middle $n/2$ data points, tossing out the $n/4$ highest and lowest points. The median is defined as the value that lies equidistant between the lower and the upper half of sample values.

The last row of table 4.4 denotes the (arithmetic) mean of the value for the raw data $s_{d_{down},d_{cross}}$ of each of the 20 scans,

$$\text{mean}_{d_{cross}} \left[\text{RMS} \left(s_{d_{down},d_{cross}} - \text{hmean}_{d_{down}}(s_{d_{down},d_{cross}}) \right) \right] \quad (4.6)$$

The high value confirms that all the mean methods produce a better signal than the untreated signal. The value also gives an idea of how much better than the raw data each of the smoothed signal's versions is.

4 Implementation

Figure 4.12 shows in gray the raw data of the first 20 range scan lines from the “Mancave” dataset and compares the different presented methods of averaging over the 20 scans.

Signal	RMS of difference to harmonic mean
Harmonic mean	n/a
Geometric mean	0.144
Arithmetic mean	0.275
50 % Trim mean	0.284
Median	0.318
Raw data, mean	1.064

Table 4.4: Comparison of mean types by RMS relativ to harmonic mean

Due to the good signal quality, the implementation uses the harmonic mean to average the bins. Weighting the average with triangular or Gauss-shaped weight distribution did not noticeably improve data quality for any of the averaging methods.

Note that the range signal is not the only signal that needs to be averaged in a range bin. All other parameters that are part of the range scan need to be averaged as well. These parameters are mileage at scan time, robot position and orientation, and robot speed. Sweep time and down range bins don't change.

4.5 Doppler Estimation with the Peak Gradient Algorithm

The peak gradient algorithm is a way to find Doppler speeds from consecutive range profiles.

In figure 4.13a the scans at radar mileage 0.355 m and 0.375 m of a scene (“Basement”) are overlaid. At this small cross range difference the range profiles of the two scans are relatively similar. However, as visible in figure 4.13b some peaks from target echoes are shifted, as the distance to the targets changes with the radar moving through the scene. The rate at which the distance to a target changes is its relative speed to the radar, the Doppler speed.

Usually, speed is measured in distance per time. In this case, it actually makes sense to ignore the scan's time stamps and look at the cross range (driven mileage) instead. With the Doppler speed as change of down range (distance to target) per cross range driven, calculations become time independent and hence radar movement speed independent.

A target's distance from the radar is assumed to be at the range bin of the corresponding peak in a scan's range profile. When a target's distance changes the peak will shift, too. This is visible in figure 4.13b and we can read the Doppler speed from the figure. The peak around 0.67 m moves $d_{down,1} = 0.0216$ m closer in range, while the peak around

4.5 Doppler Estimation with the Peak Gradient Algorithm

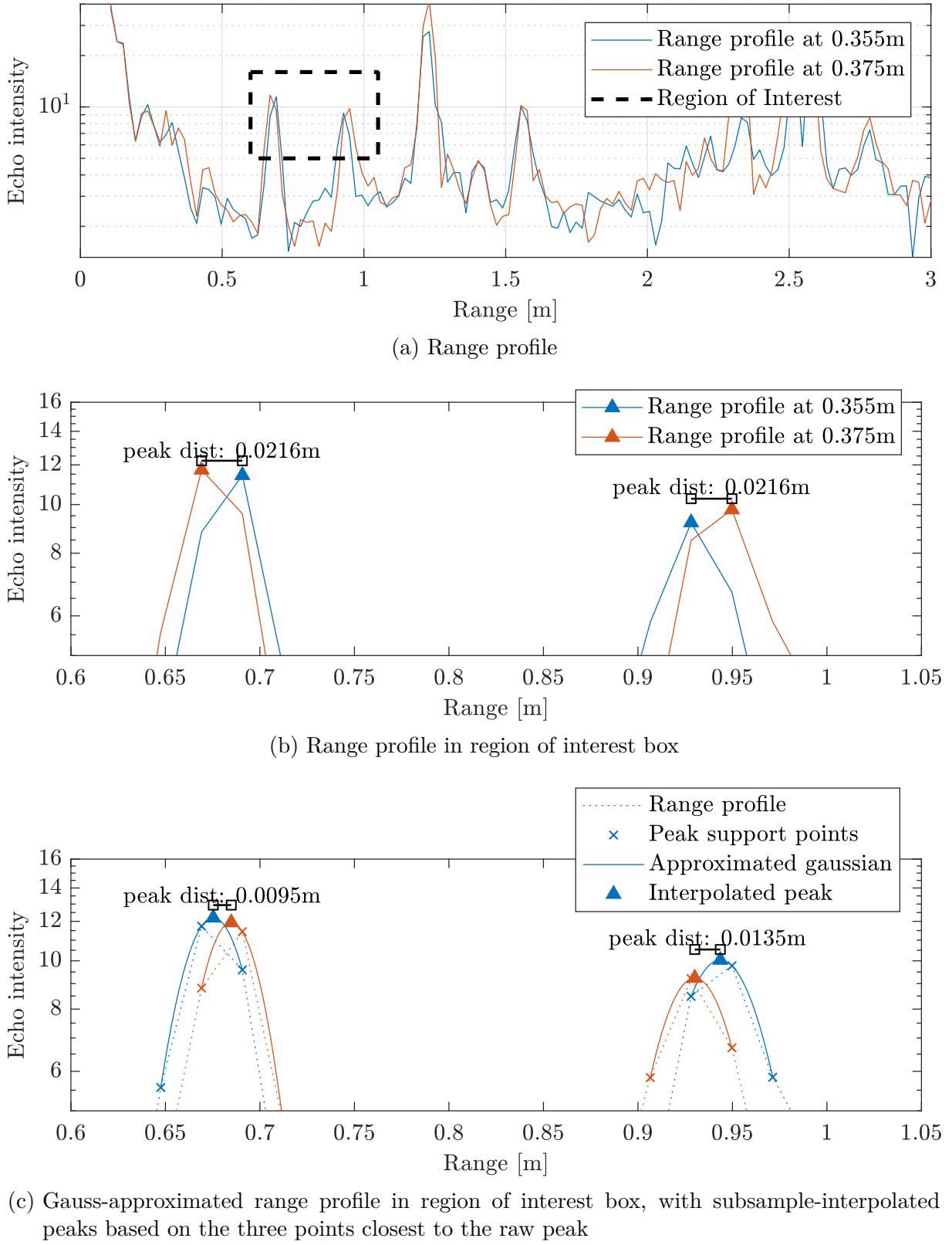


Figure 4.13: Peak detection and subsample peak interpolation to find peak distance between two range profiles

4 Implementation

0.95 m moves $d_{down,2} = -0.0216$ m closer (i.e., away). Combined with the change in cross range, $d_{cross} = 0.3752$ m – 0.3555 m = 0.0197 m, we can calculate Doppler speeds of $v_{D,1} = \frac{d_{down,1}}{d_{cross}} = 109.55\%$ (of radar movement speed) and $v_{D,2} = \frac{d_{down,2}}{d_{cross}} = -109.55\%$. If the speeds are 100 % and -100 % it would mean that the targets are directly ahead and directly behind the moving radar. Speeds over 100 % are impossible in a static environment where all relative target motion is caused by the radar movement. The targets are therefore either dynamic and moving by themselves, or the peak locations that determined those too-high speeds were not exact. Since we know that there were no dynamic moving objects in the controlled environment of the scan, the latter must be the case.

This effect of imprecise target peak localization and Doppler speed estimation could be overcome by averaging noisy data so that the average peak distance is close to the actual change in target range. A lot more scans are necessary for that though, and scan oversampling needs to be drastically reduced. This would lead to lower SNR, which means that some peaks with lower echo intensity could not be detected.

With higher downrange resolution, peaks could be localized more precisely. However, the down range resolution is limited by the available bandwidth of the radar sensor. In the Omnidaradar RIC60A, up to 7 GHz are available, which is already extremely high. Its range resolution $dR = \frac{c}{2BW}$ is roughly 2.1 cm. With this method, dR is of course the smallest measurable change of target range.

The localization of peaks is however not limited by range resolution, but by range accuracy, which mainly depends on SNR. It is much better than range resolution with $\sigma_R = \frac{dR}{\sqrt{SNR}}$. This can be utilized with subsample peak interpolation.

4.5.1 Inter-scan vs Intra-scan Doppler estimation

For correct Doppler estimation it is important to have the exact timing, or for relative Doppler speed, exact cross range mileage of the range scans whose peaks are compared in the peak gradient algorithm.

The Omnidaradar sensor can send multiple consecutive sweeps without any delay between them. The timing will then be very exact, because the precise length of one sweep is known. However, the number of sweeps in such a set of sweeps is limited (transmission of high data volumes will often fail), so they can't be smoothed through oversampling and will be noisy. Smaller peaks will then not be detected reliably. Another problem with this approach is that it will give target Doppler speeds in terms of change of down range over time, but not the robot speed-invariant relative Doppler speed in terms of change of down range over change of cross range.

Inter-scan comparison gives better Doppler estimation, because the data can be smoothed through oversampling first. Consecutive sweeps can still be used: They need to be separated into individual scans with timestamps adjusted to $t_{msg} + i \cdot t_{sweep}$ (with message

timestamp t_{msg} , consecutive sweep index i and sweep duration t_{sweep}) and cross range mileage interpolated at that timestamp.

4.5.2 Subsample peak interpolation

In subsample peak interpolation a curve is fitted on several supporting points in the coarse-resolution data. In the case of a single, non-overlapping radar echo peak, a Gaussian pulse of the form

$$g_i(x) = a_i e^{-b_i(x-c_i)^2}$$

is a good approximation. In figure figure 4.13c, the data point of the respective peak as well as its left and right neighbors are fitted with a Gaussian. The fit parameters a , b and c are calculated using Travis Wiens's crit_interp_g function³⁷⁾.

As evident by visual inspection, the intensity and location of the fitted function's maximum are much closer to the real value.

With the same procedure as explained above, we measure peak distance shifts of $d_{down,1} = 9.510$ mm and $d_{down,2} = -13.52$ mm in figure figure 4.13c and can hence estimate Doppler speeds of $v_{D,1} = 48.26\%$ and $v_{D,2} = -68.63\%$ of radar movement speed. These values are a much more plausible estimation and generally work very well when used to calculate the reprojection angle in the reprojection method.

4.5.3 Peak matching

Visually it seems clear which peaks belong to each other in consecutive range scan lines. But for the algorithm this poses a challenge. Range scan lines usually don't have the same number of peaks because new targets can appear, old targets can disappear, noise can temporarily mask out targets peaks and target arcs cross each other. However if the cross range difference, i.e. the driven distance between scans, is not too high, a single target will not change very much in both range and echo intensity. This allows the detection of peak matches, between which the Doppler speed will be calculated. The parameters for peak matching tolerance are therefore allowed intensity change `ValueSearchArea` and allowed range change `LocSearchArea`, respectively:

$$\text{ValueSearchArea} = \frac{\delta I}{\Delta d_{cross}} \quad (4.7)$$

and

$$\text{LocSearchArea} = \frac{\Delta d_{down}}{\Delta d_{cross}} = v_{D,\max} \stackrel{\text{eq. 3.3}}{=} v_R \cos(\alpha) \quad (4.8)$$

with intensity change factor δI and down range deviation Δd_{down} per added cross range mileage Δd_{cross} .

³⁷⁾ <https://www.mathworks.com/matlabcentral/fileexchange/24465>

4 Implementation

$\Delta d_{down}/\Delta d_{cross}$ is more easily expressed as Doppler speed v_D in percent of robot speed v_R . It makes sense to allow LocSearchAreas greater than 1 v_R to deal with noise and jitter – otherwise a target peak closing at roughly v_R would not always be detected and smoothing over cross-range would be worse. The detected $v_D > v_R$ of course need to be clipped to v_R or ignored in reprojection (which is what the current implementation does). Good values are determined to be a ValueSearchArea of 5 cm^{-1} and a LocSearchArea of $2\text{ }v_R$.

4.5.4 Transmit crosstalk suppression

At low range, spurious peaks occur. The first one is caused by transmit antenna crosstalk and is visible as very high intensity echo around $d_{down} = 0$. After the transmit antenna crosstalk spike there is another peak around $d_{down} = 0.25\text{ m}$ which is consistently visible. It can be explained with static objects sitting close to the radar, i.e. robot parts and the floor below the robot.

These spurious peaks create two problems: (1) automatic color scaling or height scaling respectively in plots is more difficult, and (2) high intensity false positives would be visible next to the robot path in the final map.

Hence these spurious peaks must be ignored during peak detection. This can be achieved in two ways. The first is to simply replace all data values under a down range limit d_{mute} (usually $d_{mute} = 0.30\text{ m}$) with NaN values. For the second way, one effect of range compensation is exploited. As shown in figure 4.14, the spike that previously had its maximum in the first range bin, at $d_{down} = 0$, now has its maximum in a later range bin. The reason is that the range compensation factor at $d_{down} = 0$ is $r(0) = 0$ but $r(d_{down} > 0) > 0$. As the transmit peak now has a maximum with neighbors lower than its maximum, `findpeaks` can find it. Color scaling can then be made to work correctly by clipping all values higher than the *second highest* peak. The Peak Gradient Algorithm also has an optional parameter `SkipFirstPeak` which, when set to *true*, ignores the first peak in each range scan line. This can help to ignore these echoes.

4.5.5 Minimum Peak height

Without countermeasures a lot of noise with low height and low full-width-at-half-maximum will clutter the reprojection map. The easiest way to prevent this is to apply a threshold on the detected peak height that rejects all peaks that do not reach a certain minimum height. It's not easy to specify such a threshold that works for all scans, because the noise floor level strongly depends on the chirp time. Figure 4.8 shows that this persists even when the echo intensity is normalized to intensity 1 at the first range bin before transmit cross talk suppression (see section 4.5.4).

4.5 Doppler Estimation with the Peak Gradient Algorithm

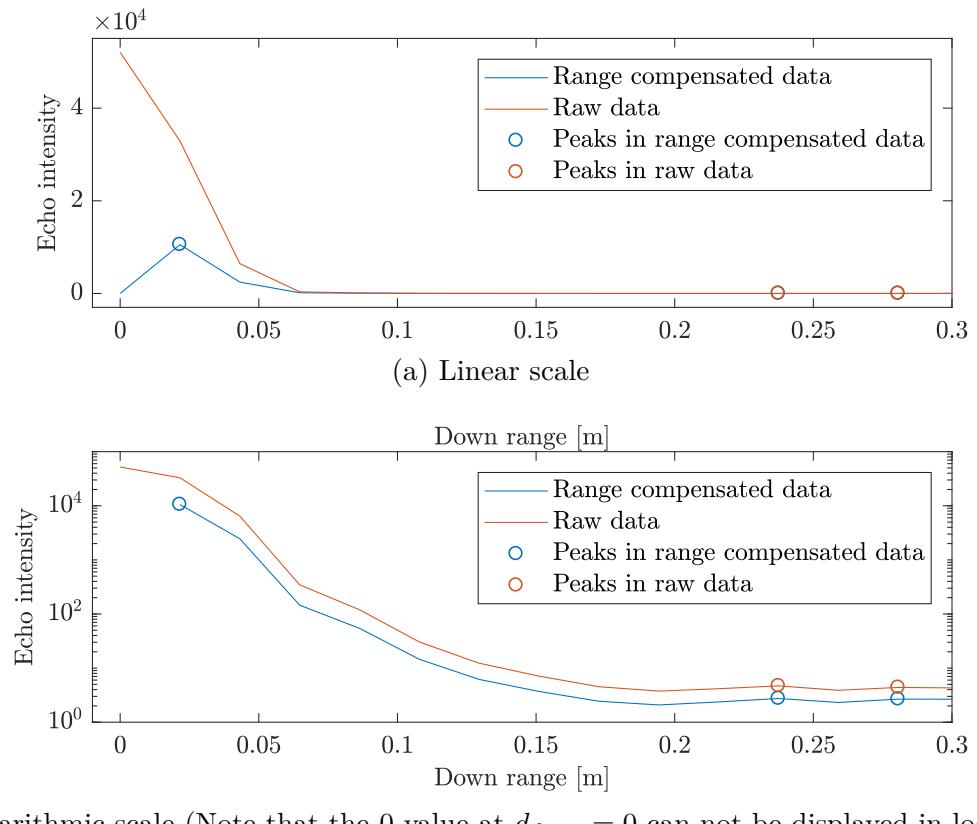


Figure 4.14: Attenuating effect of range compensation on transmit peak in a range profile

4.5.6 Peaks overlaps at crossing target arcs

Often there appears a situation where the range cell migration arcs of two targets cross each other. In this case, their peaks can almost always be still detected separately. However peak-associated data like DOA and Doppler speed will superpositioned. Using these in a regular fashion would yield wrong results. An approach that proved to work well in practice is to sort all peaks in a range scan line by ascending value (peak height), then assigning the average over the FWHM of the feature in question to all output down range cells within half the FWHM of the respective peak. Higher peaks are hence treated as more important and overwrite the values of lower peaks. Figure 4.15 is an example where the overlapping effect is visible.

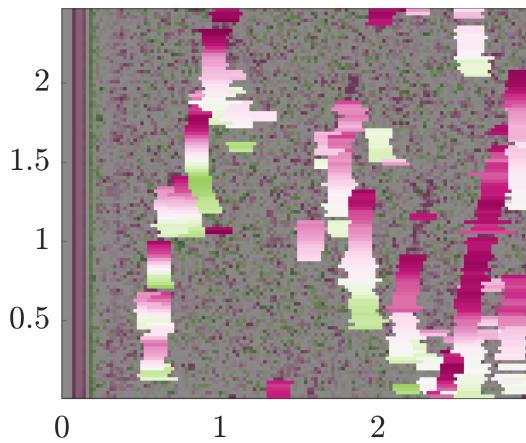


Figure 4.15: DOA of Attic scan. The DOA of higher peak values is applied over the FWHM, superseding the values of lower peak values.

TODO annotate in figure where the overlap is

4.5.7 Output

In this implementation the resulting size of the reprojection map is known in advance so the map does not need dynamic resizing and can be allocated ahead of time. To make sure everything will fit, the initial size is chosen as `round[max(radar_position) - min(radar_position) + 2 * radar_max_range + 2 * 0.2 m]` in both X and Y dimension.

The `world_res_m` parameter defines the resolution of the gridmap to be created. Larger values (e.g. 10 cm) result in coarse homogeneous map, while small values (e.g. 1 cm) lead to unmapped area between reprojection lines. The latter is because noise and jitter in Doppler speed estimation cause imprecision in reprojection angles. As explained in section 4.7.3, the reprojected down range cells of the range scan lines fall into a maximum of four output pixels. With a limited number of reprojections that have slight variations in reprojection angle, this can result in gaps between the output cells. The optimal

`world_res_m` is hence somewhere between these extremes, a good value is slightly larger than the down range resolution dR , which is around 2.1 cm in this implementation.

By saving the output after each processed range scan line and reassembling it, a video of the map buildup can be generated. Otherwise, the map can be trimmed of its empty cells.

For some scans, the `map` topic is available. In this case, the radar map can be overlaid over the lidar slam gridmap. To achieve this in Matlab, the lidar map is plotted as `surface` of zero height and an RGB texture that is generated from the values in the gridmap. The radar map is then overlaid as separate surface with an alpha channel of zero opacity for empty reprojection map cells, 70 % opacity for reprojection cells with a normalized value $\geq 70\%$, and linearly mapped opacity between 0 % to 70 % normalized value.

4.5.8 Limitations

Next to the inherent limitations of reprojection mapping (see section 3.8) there are some limitations that arise from the presented implementation.

Problems with imperfect fit function for subsampling

The subsample peak interpolation in section 4.5.2 works with a Gaussian fit function. This is a good but not perfect approximation of the real peak shape. The effect only becomes noticeable when the cross-range distance Δd_{cross} between (smoothed) range scan lines becomes very small. Interpolated peaks at the center and edges of the down range bins are correctly placed because both the fit function and the true peak shape are symmetric around the true peak. In between those however, peaks are interpolated slightly off because the fit function does not perfectly reflect the true shape of the curve. When examining a peak migrating at a constant speed through the range profile, its subsampled peaks will then lie on an S-shaped path between every (correctly interpolated) cell edge and center. In effect, the Doppler speed estimated with the peak gradient algorithm (see section 4.5) will show oscillations around the true Doppler speed. Since wrongly estimated Doppler speeds cause wrong reprojection angles, this causes a radial smearing of targets in the reprojection map.

The effect becomes less noticeable with greater cross-range distances and completely disappears at $\Delta d_{cross} \geq \Delta d_{down}$. Other than increased performance, that is also the reason for choosing a binning approach over a running average for raw data smoothing (see section 4.4.2).

4 Implementation

Problems with close targets

When Doppler speed is measured directly using FMCW, there will be several Doppler peaks, each representing a different target at the same range but with individual relative speeds. With the Peak Gradient Algorithm however, multiple targets at the same range are difficult to separate. In some cases this is only a temporary problem and is resolved by the radar moving a little farther so the ranges are separated by more than the range resolution. Sometimes however some peaks come from point-like targets that are close together, like parts of a wall. This bundle of targets is however not always separated by the same range. Especially in the case of a wall, the traces of visible points will cross each other as they slide on the sine arc (see figure 4.16). When the points are close together, only the brightest spots will be seen as peaks, and the trace of the detected peak matches will describe a squiggly motion. This causes the estimated Doppler speed to wander around the common speed. To combat this effect, a higher accumulation distance can be used during oversampling preprocessing, so the peaks move together so closely that they actually form a single target.

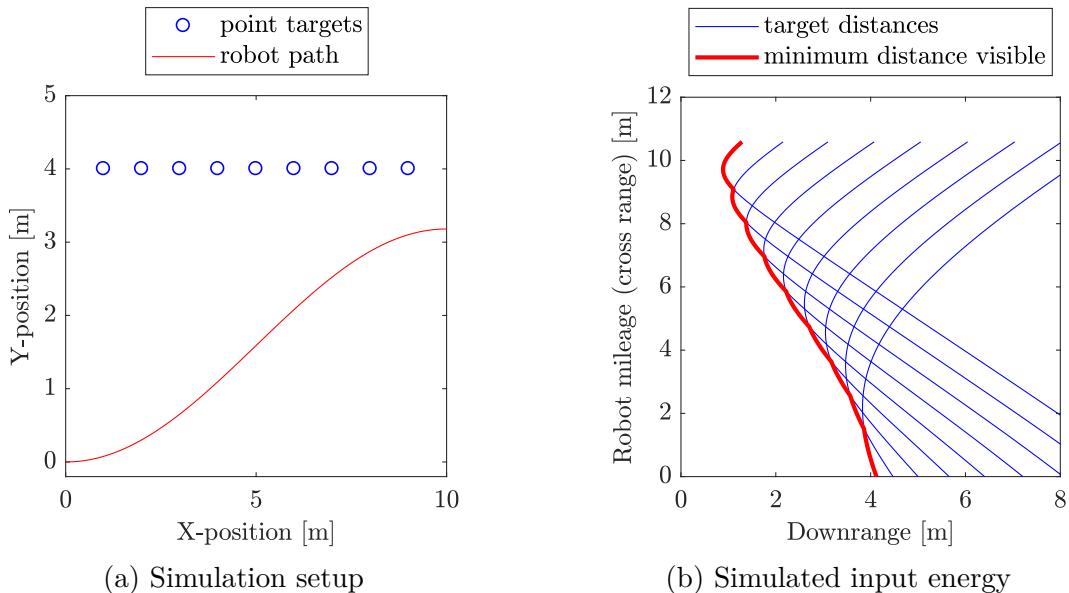


Figure 4.16: The target echo at the closest range usually has the brightest intensity. This can lead to errors in Doppler speed estimation.

4.6 DOA Estimation

In geometries as described in section 3.3, the next step is to estimate the Direction of Arrival to resolve reprojection angle ambiguities. As described in section 2.2.4, the Direction of Arrival (DOA) angle can be measured from the phase difference at the receiving antennas of a multistatic radar.

In the case of RIC60A the antenna separation $d = 1.16$ mm and wavelength $\lambda = \frac{c}{60\text{GHz}} = 5.0$ mm (with speed of light c).

Figures 4.17 and 4.18 show the range profile and phase shift of the “Basement” scan. The phase shift is very noisy in the regions without a target peak in the range profile but exhibits steady values following a gradient over targets.

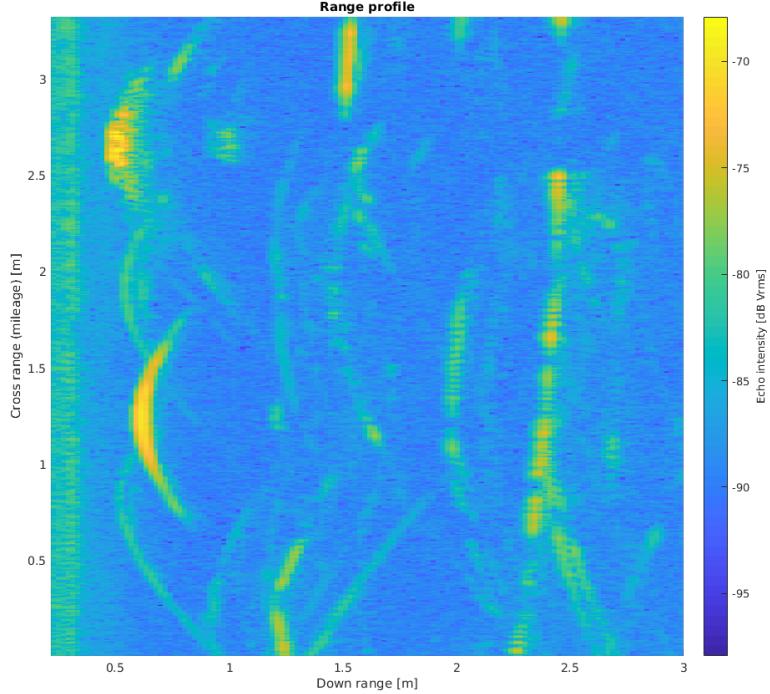


Figure 4.17: Range profile of Attic scan with color coded echo intensities at cross range over down range

Four steps are performed to get a reasonable estimation of direction of arrival.

Peak detection. In a first step, target peaks in the range profile are detected. The algorithm records for each peak in each range scan line its fitted interpolated location, full width at half maximum, and matching peaks in adjacent lines regarding value and location.

Down range averaging. In each range scan and at every detected target peak, the phase shift is averaged over the width of the respective detected peak. The average is weighted using the Gaussian fit from subsample peak interpolation.

Cross range averaging. In every range scan line, each peaks phase shift is averaged over a configurable accumulation distance in cross range dimension. This is done by taking the arithmetic mean of all the phase shifts at all matching peaks (regarding value and down range location) within accumulation distance in cross range dimension.

Cut out. Noisy values at non-target peak range bins are masked out.

4 Implementation

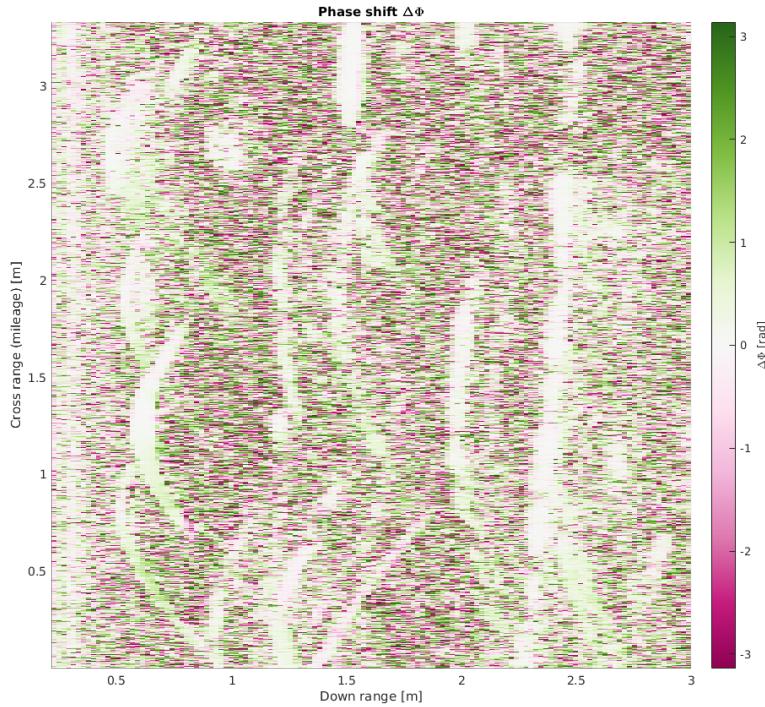


Figure 4.18: Phase profile of Attic scan with color coded inter-antenna phase difference at cross range over down range

DOA calculation. In each scan line, each peaks direction of arrival θ is calculated from the smoothed phase shift values, using equation 2.21.

Figure 4.19 shows the result of these four steps applied on the data of the “Basement” scan. The direction of arrival seems plausible: In this side-facing scan the robot passed some metal cans. Apparently the radar sensor was not mounted perfectly orthogonal to the robot’s movement direction (which is not necessary for the reprojection method), but was slightly off. This can be seen at the closest points of the target arcs. At the pericenter the line of sight to a target is orthogonal to the robot’s movement direction, but the DOA value shows to be around 5° to 10° .

Note that if the radar sensor is mounted inverted (rotated by 180°), DOA values have to be multiplied by -1 to keep right and left where they are.

4.7 Reprojection Mapping

4.7.1 Orientation parameters

Before the reprojection can be executed, the physical orientation of the radar sensor needs to be known to the algorithm. In the implementation, the two Boolean parameters `forward_looking` and `mount_inverted` control the behaviour. If the radar’s squint angle

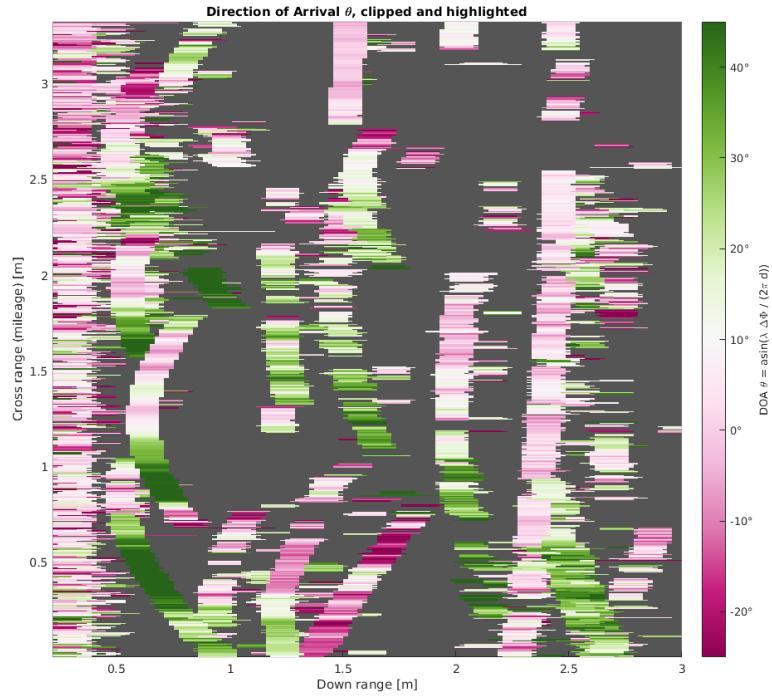


Figure 4.19: Direction of arrival estimation for Attic scan

and angle sensitivity are such that the field of view reaches both sides of the robot path, the `forward_looking` parameter needs to be enabled. This enables the processing of DOA data to find the sign of every target's reprojection angle i.e. if it is to the left or right side of the robot's motion path. If the radar is mounted in an upside-down configuration the squint angle is not affected, but if the DOA values are processed they need to be mirrored (multiplication with -1) because the left and right antennas are switched. Otherwise, targets will be projected to the wrong side of the robot's path.

4.7.2 Projection direction

The radar reprojection can be executed as forward or backward mapping. The proof of concept implementation has an optional parameter `ProjectionMethod` to switch between forward and backward mapping.

Backward mapping

If backward mapping is enabled, the reprojection algorithm still operates range scan line based, but iterates over each pixel in the map. While this is computationally much more intensive, it allows to add negative information by reducing the map's value at pixels that are known to not contain a target because the range scan line does not feature a peak at that range.

4 Implementation

```

foreach range_scan_line in range_scan_lines
    foreach pixel in map
        pixel_angle = robot.get_angle_to(pixel)
        distance = robot.get_distance_to(pixel)
        if distance < max_range && pixel_angle in field_of_view_range
            range_bin = range_scan_line.interpolate_at(distance)
            if range_bin.has_peak
                peak_angle = range_bin.peak.doppler.to_angle
                if peak_angle == pixel_angle
                    map.at(pixel).add(range_bin.value)
            else
                map.at(pixel).reduce_value

```

Forward mapping

In forward mapping, the reprojection algorithm iterates over each range bin in each range scan line. Detected peaks are cut out and reprojected to a position on the map which is calculated from relative Doppler speed, range, and robot position. The projection target coordinates don't usually fall exactly on the map grid points. The implementation uses sample splitting to distribute a value over the nearest pixels in this case.

```

foreach range_scan_line in range_scan_lines
    foreach peak in range_scan_line
        foreach range_bin in peak
            target_coords = get_coords(robot.position, range_bin.range, peak.doppler)
            weights, neighborhood = split_sample(target_coords)
            map.at(neighborhood).add(weights, range_bin.value)

```

4.7.3 Sample splitting

To avoid aliasing when projecting a pixel in the forward projection direction, the sample is split over the four closest map pixels. The split is weighted with the distance of the target coordinates to the closest pixel centers.

If the target coordinate is $p_{target} = (x_t, y_t)$, then the horizontal and vertical distributions ν_h and ν_v , respectively, are

$$\nu_h = \frac{x_t - \lfloor x_t \rfloor}{\lceil x_t \rceil - \lfloor x_t \rfloor} \quad (4.9)$$

$$\nu_v = \frac{y_t - \lfloor y_t \rfloor}{\lceil y_t \rceil - \lfloor y_t \rfloor} \quad (4.10)$$

The pixel weights $p_{x,y}$ are then

$$\begin{aligned} p_{\lfloor x_t \rfloor, \lfloor y_t \rfloor} &= \nu_v \nu_h \\ p_{\lfloor x_t \rfloor, \lceil y_t \rceil} &= \nu_v (1 - \nu_h) \\ p_{\lceil x_t \rceil, \lfloor y_t \rfloor} &= (1 - \nu_v) \nu_h \\ p_{\lceil x_t \rceil, \lceil y_t \rceil} &= (1 - \nu_v) (1 - \nu_h) \end{aligned}$$

such that

$$\sum_{\substack{x \in \{\lceil x_t \rceil, \lfloor x_t \rfloor\} \\ y \in \{\lceil y_t \rceil, \lfloor y_t \rfloor\}}} p_{x,y} = 1 \quad (4.11)$$

In the special case where $y_t = \lceil y_t \rceil = \lfloor y_t \rfloor$ there are only two instead of the four neighboring pixels. Their weights $p_{x,y}$ are

$$\begin{aligned} p_{\lfloor x_t \rfloor, y_t} &= \nu_h \\ p_{\lceil x_t \rceil, y_t} &= (1 - \nu_h) \end{aligned}$$

The same applies when $x_t = \lceil x_t \rceil = \lfloor x_t \rfloor$:

$$\begin{aligned} p_{x_t, \lfloor y_t \rfloor} &= \nu_v \\ p_{x_t, \lceil y_t \rceil} &= (1 - \nu_v) \end{aligned}$$

And lastly, if $(y_t = \lceil y_t \rceil = \lfloor y_t \rfloor) \wedge (x_t = \lceil x_t \rceil = \lfloor x_t \rfloor)$,

$$p_{x_t, y_t} = 1$$

4.7.4 Range compensation

As evident from the classical radar equation 2.3, echo intensity decreases with the fourth power of distance. This has the effect that reprojected targets appear brighter when they are mapped from close distance; but most importantly, when targets are detected from a far distance, mapped intensities are decreased due to the map's averaging.

This attenuation can be compensated with a range-based compensation factor f_r with

$$f_r(d_{down}) = \left(c_a + \left(\frac{d_{down} c_b}{c_c} \right)^{-4} \right)^{-1}$$

Figure 4.21 shows the range profile of the “Torture Chamber” scan both as oversampled raw values (top subplot) and with range compensation enabled (bottom subplot). The middle subplot details the range scan line at one cross range highlighted by the red lines. Inspecting this detail graph reveals that both target peaks and noise floor are lowered in the near range, while the noise floor stays constant in the far range. This helps to keep a target’s intensity at at least similar values over all ranges in the map.

TODO split in subfigures, adjust refs, annotate

4 Implementation

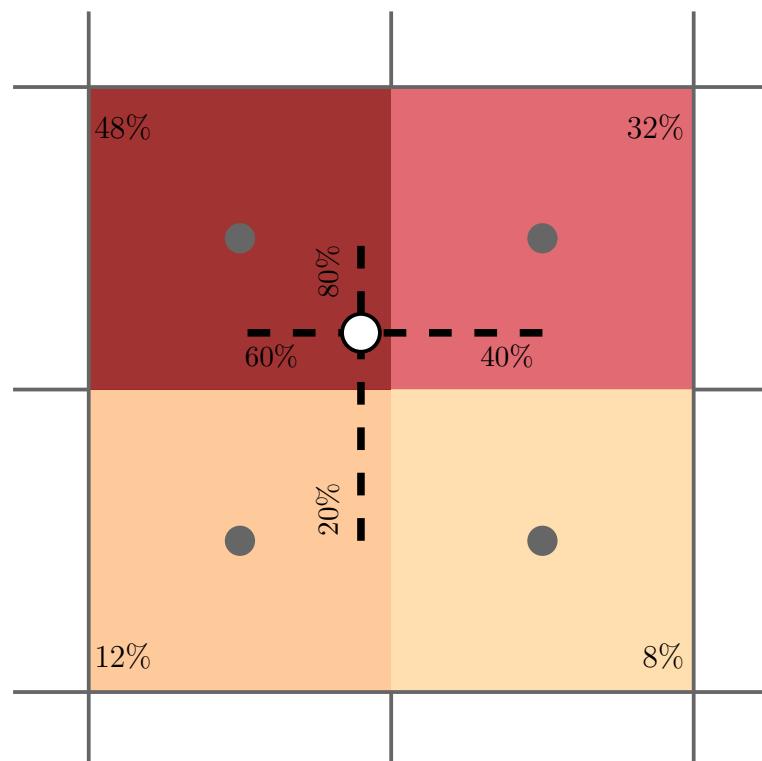


Figure 4.20: Illustration of a pixel value (white circle) being distributed over the four nearest grid neighbors. The shade of the neighbor's area indicates the individual distribution weight.

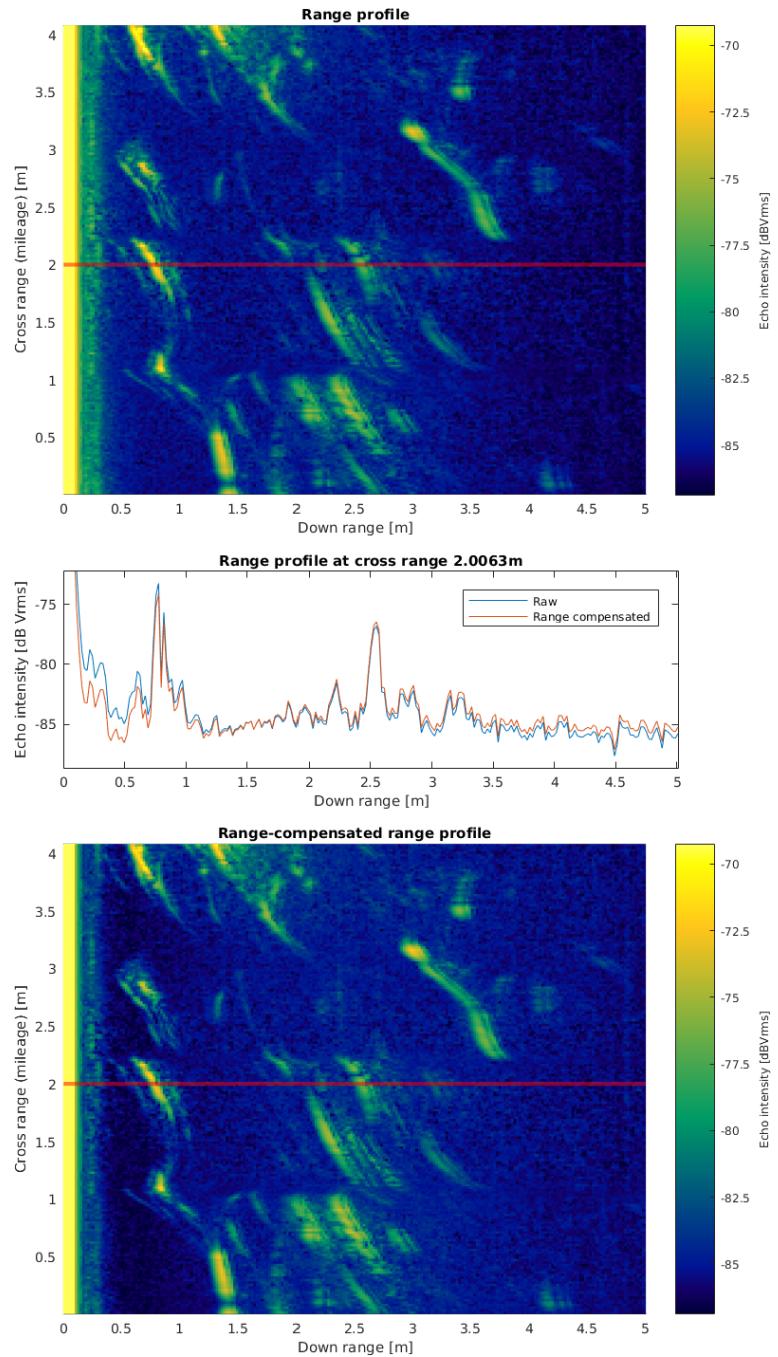


Figure 4.21: Range compensation

4.7.5 Angle sensitivity compensation

The intensity of a target peak depends on its angle with respect to the antenna. The angle is unknown before the Doppler speed is estimated, so the knowledge about echo attenuation caused by antenna angle sensitivity can not be used to improve peak detection. But the echo intensity influences how a target is represented on the reprojection map. Since the map averages all reprojections to any given point a low intensity echo will reduce the visibility of a target on the map. This can however be compensated by multiplying detected target peak heights with a factor that is based on the angle the target is believed to be seen under.

The angle compensation factor curve was found by experiment. A strong point target (retroreflector) was placed at a known range away from the robot. The robot was then made to rotate around itself, such that the target comes into view and leaves again. Meanwhile, the radar scans, together with robot odometry were recorded.

The radar was not mounted over the center of rotation of the robot. This way, the radar did describe a circular path whose mileage can be calculated. The angle compensation measurement can hence be visualized in figure 4.21 in the usual range profile with echo intensities over cross versus down range. The range of the retroreflector varies with twice the distance of the radar to the robot's rotation center, but only the orientation is interesting - the range can just be summed up over the range bins the target is visible in (in this case, 0.45 m to 0.85 m).

In figure 4.21, the same range scan lines are sorted by robot orientation during the scan. After the explained summing in down range dimension the intensity (absolute value of complex signal) data of both antennas is binned separately over 60 orientations from $-\pi$ to π .

The manufacturer later provided angle sensitivity measurements of the IC's manufacturing batch. The measurements show that the experimental approach produced valid results.

The compensation factor f_a for each angle is composed using the formula

$$m = \max(\max(s_{Rx1}), \max(s_{Rx2}))$$

$$f_a = \frac{1}{2} \left(\frac{m}{s_{Rx1}} + \frac{m}{s_{Rx2}} \right)$$

Multiplying a peak which is to be reprojected at angle α with angle compensation factor $f_a(\alpha)$ results in a peak height that is independent of observation angle.

4.7.6 Angle compensation window

Detected peaks are not in a single range bin, but form a curve over several range bins. Multiplying each of these range bin's values with the same f_a does work, but leaves hard

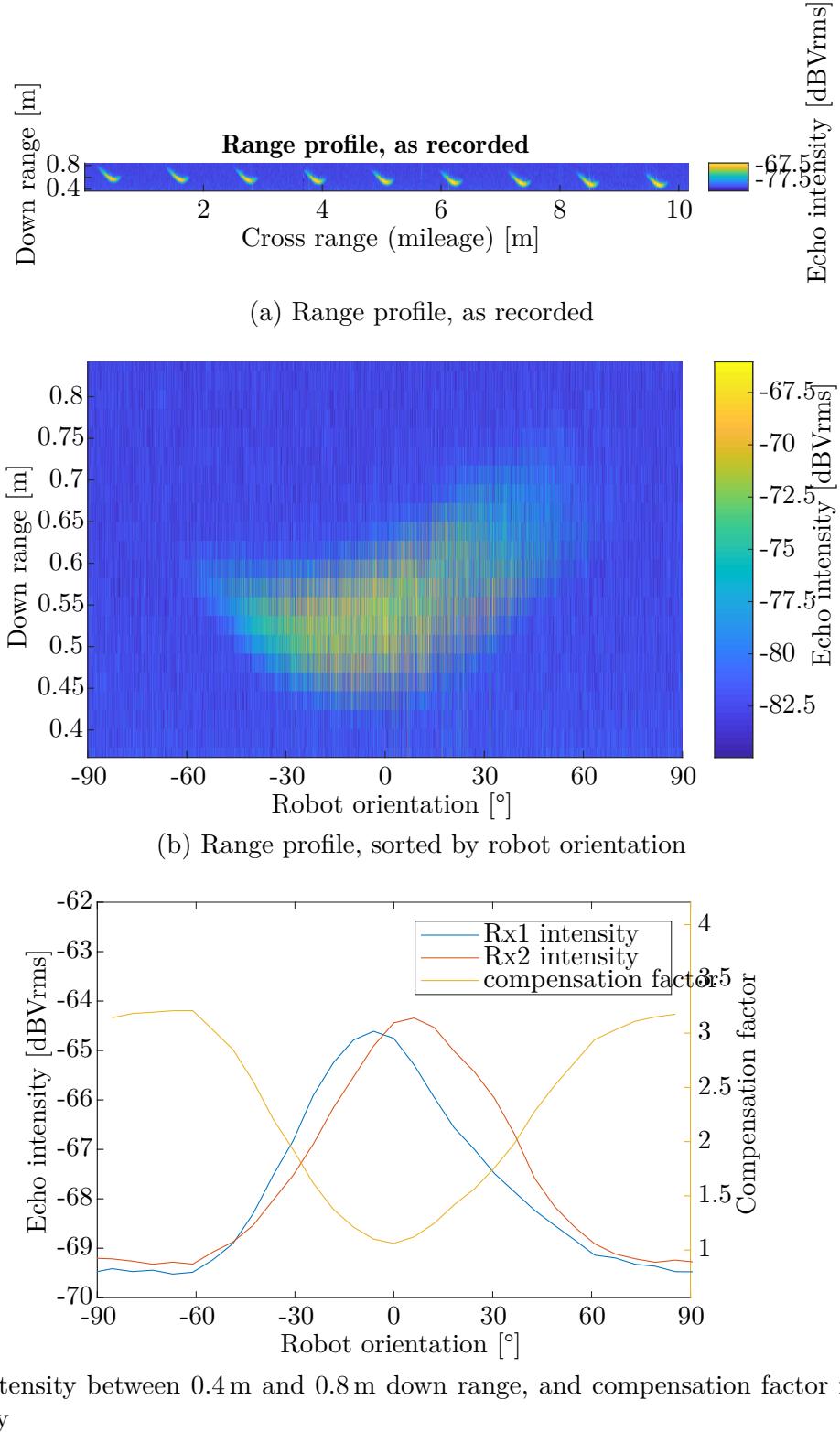


Figure 4.22: Angle compensation factor

4 Implementation

edges. It is better to multiply the peak with a window function w of height f_a :

$$w(x, f_a) = 1 + (f_a - 1) e^{-\frac{(x-p_x)^2}{p_w^2}} \quad (4.12)$$

where p_x is the peak's subsample-interpolated peak location in down range space and

$$p_w = \frac{\left(\frac{fwhm}{4}\right)^2}{4\ln(2)} \quad (4.13)$$

is the peak's width, where $fwhm$ is the full width at half maximum as found by the subsample-interpolated peak fit.

Figure 4.23 show a glass wall in the “Racetrack” scan. In figure 4.23a, only range compensation is applied. In figure 4.23b, angle compensation is switched on. In figure 4.23c, both angle compensation and windowing are switched on. The figure shows how angle compensation helps to keep the echo intensity of a mapped object at the same level, regardless of the angle at which it is seen by the radar.

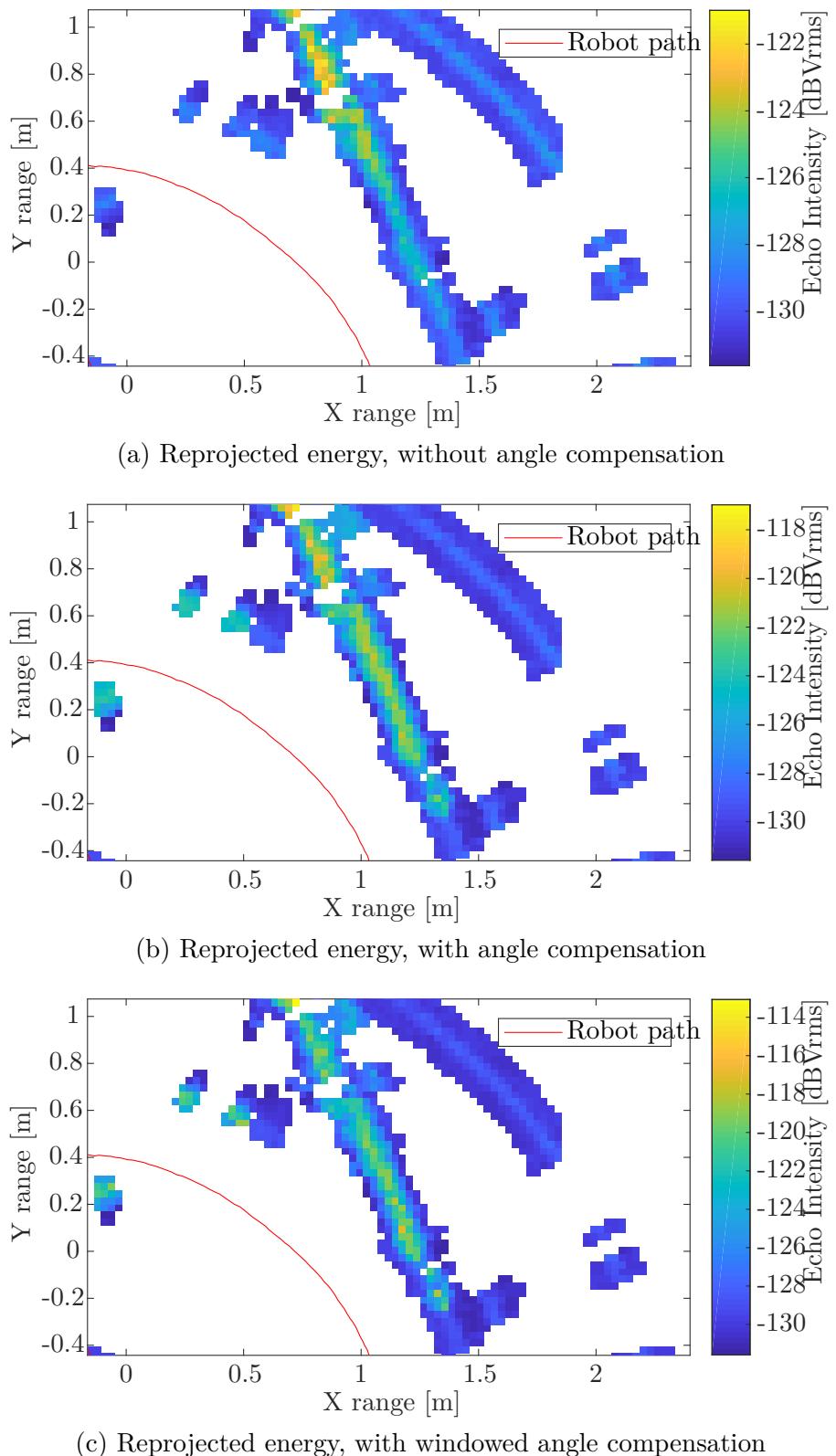


Figure 4.23: Effect of angle compensation and windowed angle compensation

5 Results and Evaluation

5.1 Evaluation

Before the results of the presented proof-of-concept implementation are listed it needs to be clear what dimensions are important so that a distinction between good and bad results can be made.

A good reprojection map has several qualities that make it usable as an obstacle gridmap. First, it needs to be **timely**. That means that the time between the appearance of an object in the robot's path and it being mapped in an obstacle map needs to be small enough that the robot can still avoid collisions. The **false alarm rate**, representing false positive detections, needs to be low enough to not negatively influence the robot's navigation behaviour by making it circumnavigate too many phantom obstacles. Vice versa, the **missed target rate**, or false negative detection needs to be low, or the radar sensor will not bring an advantage over conventional obstacle sensors. The resulting map of course also needs to be **spatially correct**. This means that a detected obstacle is mapped at its true location. Otherwise, phantom obstacles or incorrectly sized obstacles will degrade map quality. The map becomes more useful if it contains **diverse types of obstacles**, and not just one class of objects, like walls. On the other hand, it should not contain **irrelevant information**, for example wall humidity is not interesting for obstacle detection. Lastly, only if the map is **comparable to other sensors** informed comparison can take place. For example, a glass wall is easily visible to the human eye, so their position in the radar reprojection map can be verified. But metal struts in walls, which, while not presenting an obstacle to the robot (vacuum robots usually are not designed to breach walls), could be useful landmarks for later slam applications, are not visible in other maps. Hence the quality of metal strut mapping can not be asserted, but only assumed based on human knowledge of where a wall's metal struts might be.

There are some classes of targets that are particularly interesting in the evaluation of results. The first one is stacks of **metal cans**, because they appear in many of the scans. This is a kind of artificial obstacle that was designed to have a high probability of visibility in both radar scans (the curved metal surface is very reflective from every direction) and lidar scans (the can towers are high enough to cross the laser beam, and wide enough to not be missed even in some distance). Another easy target are **walls**. They are easy to see in the lidar scan, and detections in the radar reprojection map can be easily compared. A special case of walls are **glass walls** or windows. They present an impenetrable obstacle in a robots path, but almost always, neither a laser scanner nor a vision sensor can detect

5 Results and Evaluation

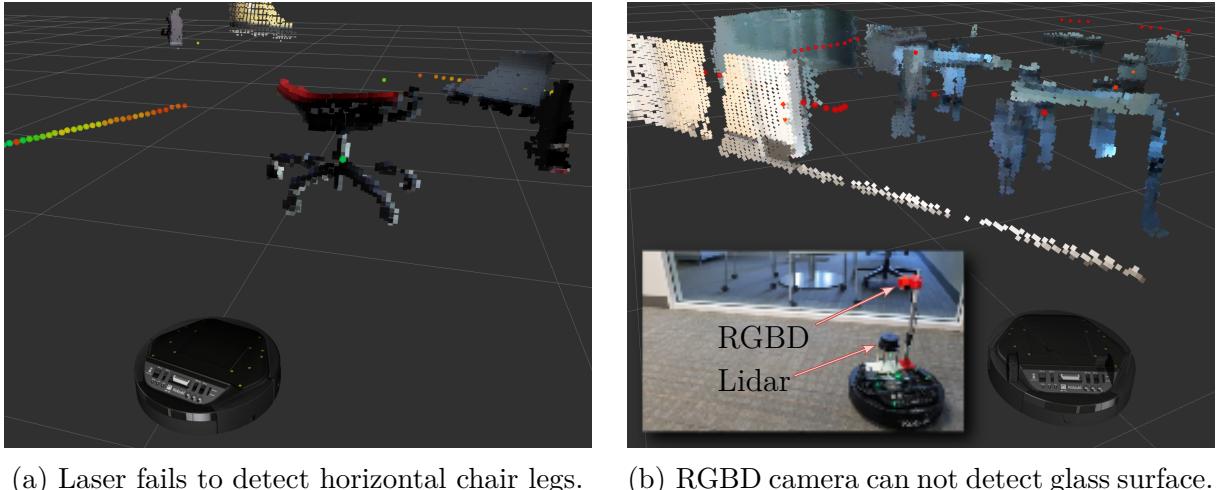


Figure 5.1: RViz screenshots showing traditional sensor shortcomings.

them (see figure 6.7b). Another real world obstacle that escapes lidar scans are **office chair legs**. While the pillar is visible with laser, the horizontally stretching legs and rollers are usually too low to be detected (see figure 6.7a). In the same category of low profile obstacles are **cables** lying on the floor. A vacuum robot can easily entangle in them and get stuck. Lastly, it would be interesting to see negative obstacles including **cliffs** and dips. Today, robots need an extra set of sensors (usually IR distance sensors at the front of the robot, aiming at the floor) to detect this kind of obstacle very reliably.

5.2 Results

During development of the reprojection method, over 30 scans were taken. The environment for the scans is the BSH office in the Bosch Research and Technology Center in Palo Alto, which is fairly representative of a typical office environment. It has carpet floors, desks, office chairs, walls, corridors, and even glass walls.

The scans of this environment are assigned code names in alphabetical order for easy referencing. Table 5.2 shows the parameters with which each of the scans was taken: Sweep time, sensor orientation (**Horizontal / Vertical / Horizontal Inverted beam fan**), squint angle (center of sensor's FOV, so 90° is sideways-looking and 0° is forward-looking), mounted horn antenna extension (which narrows the beam shape), availability of `/map` rosmessages with lidar slam information, and finally availability of `/tf` rosmessages which carry the slam-based odometry drift correction.

The scans are presented through the following plot types. An exemplary result is shown in figure 5.2 and more result plots can be found in appendix C

Input Energy This plot (e.g. figure 5.2a) visualizes the preprocessed (i.e. oversampled, range and transmit crosstalk compensated) input energy of a scan. To understand the graph, it helps to first think of it as the stacked range profile of a range sensor that looks in a 90° angle to the right of the robot. The x-axis represents down-range (in m), which is the range axis in the 1D range profile as extracted from the FMCW beat frequency (see figure 2.7b for an example). Each range profile line's echo intensity (shown on the y-axis in figure 2.7b) is visualized in a logarithmic `hparula`¹⁾ color scale. The unit is dBVRms. These range lines are stacked over the Y-axis direction, which is the cross-range dimension. This dimension shows how far the radar sensor has traveled from the scan's origin (i.e. mileage, in m). If the radar was moved at a constant velocity, it can be thought of as the time axis, however denoting and processing it as mileage has the benefit of radar velocity independence. Note that this plot does not show any information on where (at which angle) a target is, but only at what distance it is. Target peaks appear and disappear over the course of the y-axis, because they enter or leave the radar antenna's field of view.

Doppler estimation To calculate reprojection angles, a target's Doppler speed must be estimated. This graph (e.g. figure 5.2b) shows the estimation from the peak gradient algorithm: Every detected peak has a certain speed associated with it. This speed is plotted as color over the FWHM around the peak's location in down-range/cross-range space. The color follows a symmetric linear pink/white/green color scale²⁾. The estimated Doppler speed is not shown in its smoothed version, which is averaged over cross range as described in section 4.4.2 and peak FWHM as described in section 4.5.6. The background of the graph is dark grey to make sure the white middle of the color scale is visible, as well as to help highlight where Doppler peaks were actually detected.

DOA estimation The third input graph (e.g. figure 5.2c) shows DOA data as estimated in section 4.6, which in the current implementation is solely used to resolve the left-right reprojection angle ambiguity in the forward-looking geometry (see section 3.3). The (darkened) graph background visualizes the phase difference of the oversampled complex analytic input signal of both receiving radar antennas. Similar to the Doppler graph, DOA information over detected peaks is smoothed over cross-range and each peak's FWHM. The color is the same symmetric pink/white/green as the Doppler plot. The color scale is from $-\pi/3$ to $\pi/3$ radian; values above or below are clipped (but this happens only for the noisy background signal).

Reprojection map This is the main output of the reprojection mapping implementation, as described in section 4.7. The graph (e.g. figure 5.2d) shows a top-view 2D map,

¹⁾ `hparula` is a monotonic, linear-grayscale intensity colormap with enhanced contrast. It is available at <https://www.mathworks.com/matlabcentral/fileexchange/61768>

²⁾ Specifically, PiYG via the `cbrwewer` function, which is available at <https://www.mathworks.com/matlabcentral/fileexchange/34087>

5 Results and Evaluation

with the origin (0,0) at the scan starting point. Axes are in m. The color scale is again `hparula`, with echo intensities in dBVrms. The red line shows the radar motion path.

Reprojection map overlay If the `/map` is available for a scan, this graph (e.g. figure 5.2e) shows the reprojection map overlaid over the lidar occupancy gridmap as described in section 4.5.7. The lidar map has a grayscale colormap, where black means occupied, white means clear, and anything in between shows occupancy probability. Uncharted, unknown values are a “tasteful blueish greenish grayish color”³⁾. The red path is again the radar motion path.

Name	Sweep	Orient.	Squint	Horn	Map	TF
Attic	5.0 ms	H	90°	✓	✗	✗
Basement	5.0 ms	H	90°	✓	✗	✗
Cafeteria	5.0 ms	H	90°	✓	✗	✗
Dungeon	5.0 ms	V	20°	✓	✗	✗
Entryway	5.0 ms	V	20°	✓	✗	✗
Fallout shelter	5.0 ms	V	20°	✓	✗	✗
Garden	5.0 ms	V	20°	✓	✗	✗
Home cinema	5.0 ms	V	20°	✓	✗	✗
Indoor swimming pool	5.0 ms	V	20°	✓	✗	✗
Jail cell	5.0 ms	H	20°	✓	✗	✗
Kitchen	5.0 ms	H	20°	✗	✗	✗
Lobby	5.0 ms	H/I	0°	✗	✗	✗
Mancave	5.0 ms	H/I	0°	✗	✗	✗
Nirvana	5.0 ms	H/I	0°	✗	✓	✗
Orbit	5.0 ms	H/I	0°	✗	✓	✗
Public Restroom	5.0 ms	H/I	0°	✗	✓	✓
Queue	20 ms	H/I	0°	✗	✓	✓
Racetrack	2.0 ms	H/I	0°	✗	✓	✓
Sauna	2.5 ms	H/I	0°	✗	✓	✓
Torture Chamber	2.5 ms	H/I	45°	✗	✓	✓
Underground	2.5 ms	H/I	45°	✗	✓	✓
Virtual Reality	2.5 ms	H/I	45°	✗	✓	✓
Washroom	2.5 ms	V	45°	✗	✓	✓
Xray-Room	2.5 ms	V	0°	✗	✓	✓
Y is there a cable on the floor	5.0 ms	V	0°	✗	✓	✓

Table 5.2: Scan parameters

³⁾ Legal -1 value from http://docs.ros.org/jade/api/rviz/html/c%2B%2B/map__display_8cpp_source.html#l00212

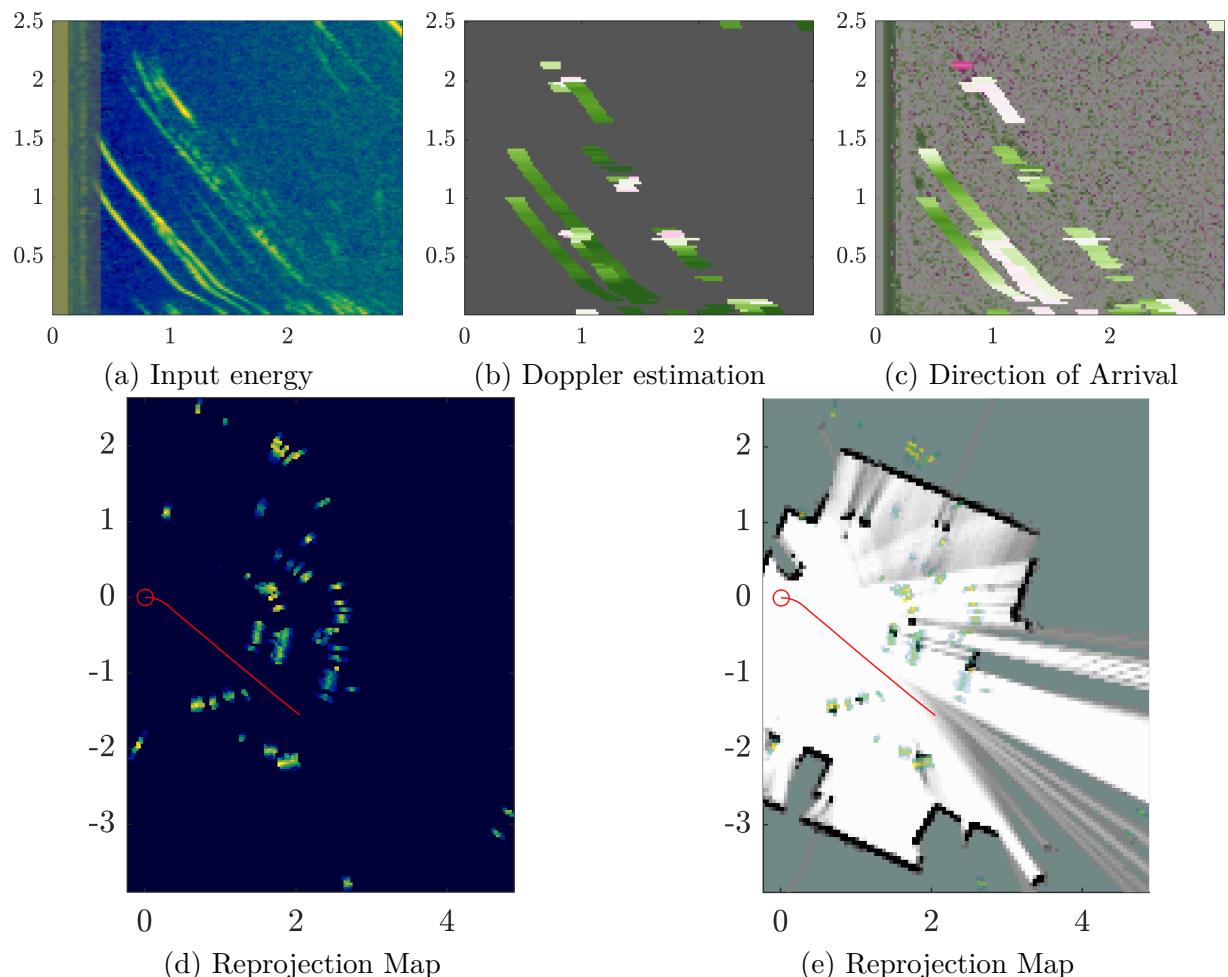


Figure 5.2: Nirvana scan

6 Discussion and Comparison

6.1 Discussion

6.1.1 Reflection directionality

Many objects only become visible when their surface is oriented perpendicularly to the incident radar waves, so that enough scattered EM energy makes its way back to the sensor. This is very visible in the Underground scan, where a glass wall is detected as the robot passes it, but not while the robot sees it at an angle.

In the Torture Chamber scan, the same effect is visible for chair legs, especially for the chair at the scene center. the legs appear in clear form as soon as the radar sees the leg from a point that is orthogonal to the office chair legs.

6.1.2 Material-dependent echo strength

Some materials, like metal, are obviously better at reflecting radar waves than others, like Styrofoam. Metal objects cause particularly strong echos which are visible from a higher distance. This can be observed in the hallway scans (e.g. Orbit, Public Restroom, Queue, Racetrack, Sauna, Underground), where the metal frames of doors and glass walls stand out in the scan.

6.1.3 Doppler vs Direction of Arrival data quality

In forward-facing geometry (scans D-T), the DOA is necessary to resolve the sign of a target's reprojection angle. This works fairly well, for example at the start of Sauna (see figure 6.1b), the closer target passes on the right (more pink), while the other targets stay to the left side of the robot (more green).

In fact, for the side-facing case, the smoothed DOA data also turned out to be very good. It could even be used to calculate a more precise reprojection angle.

6 Discussion and Comparison

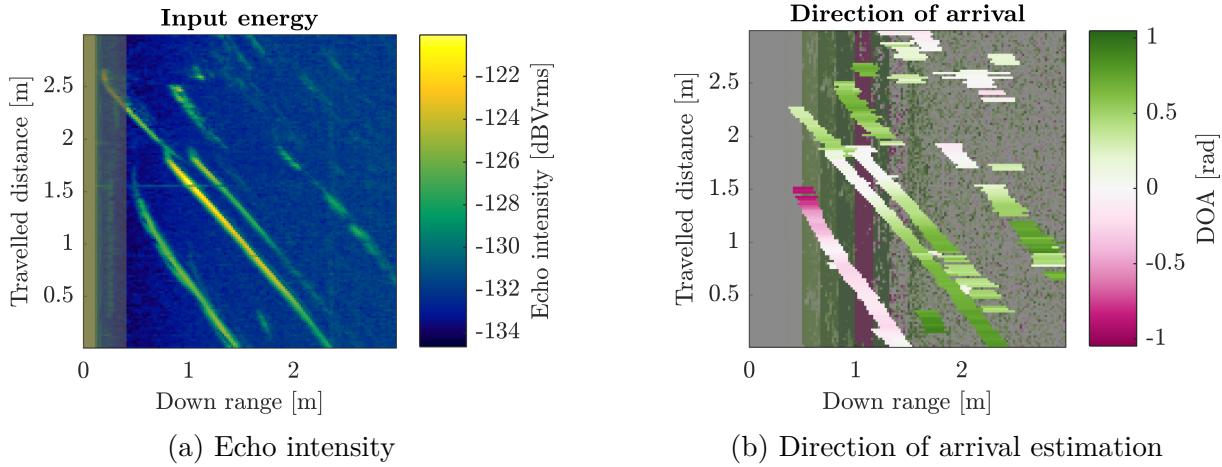


Figure 6.1: Range profile and DOA of Sauna scan.

6.1.4 Multipath effects

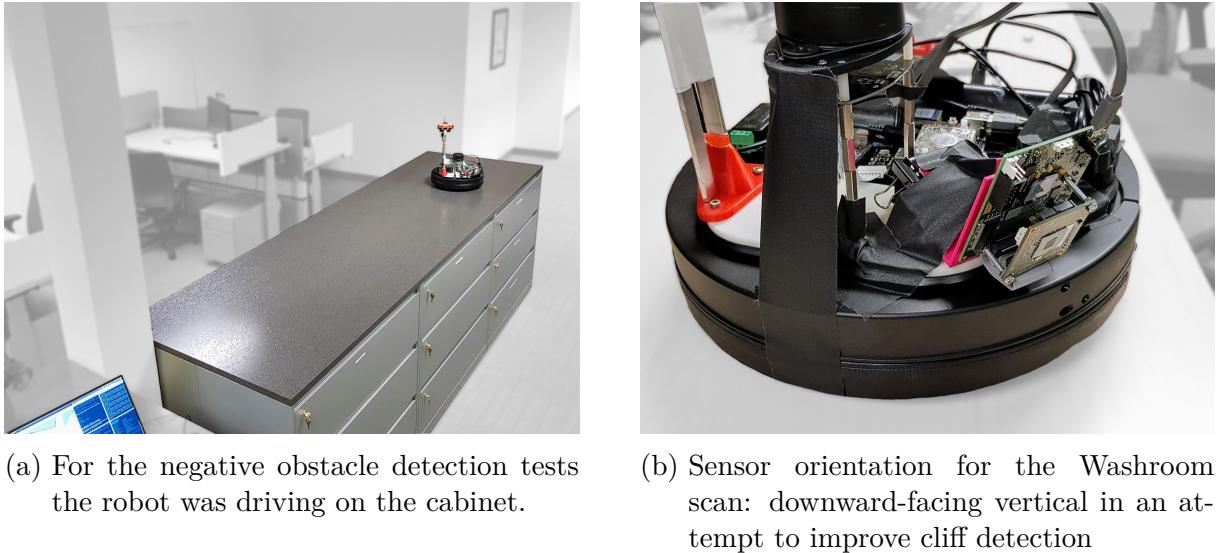
Multipath effects are a well-known problem in ground-based radar applications [34]. In situations where multi-path effects are likely, there is a higher possibility that multiple versions of a target's echo are visible, which can lead to detection of incorrect angle and ranges. Luckily, in the recorded data almost no multipath effects are obvious. The only scan that shows some effects is the Torture Chamber. There, it seems like the radar waves bounce around a bit in the (2 m, 2 m) area under the desk. The effect is that some targets are detected behind the wall behind the desk.

6.1.5 Object penetration

Some objects are penetrated by the radar waves. For example, in the Attic and Basement scans, both the front and the back wall of the plastic bottle can be seen. However, the plastic bottle was relatively close to the sensor. On the other hand, in the scans with glass walls in them (P,Q,R,S,U), no significant radar echo is picked up from the (metal) chair legs behind the glass wall. This is because a typical glass pane attenuates the 60 GHz signal by about 5.5 dB [54]. In effect, a radar sensor with higher transmission power might be able to see through walls, but in the conducted experiments radar echos were too faint to be picked up after the first bigger object (like a wall).

6.1.6 Negative obstacles

With scans V,W,X the negative obstacle detection capability was analyzed in the environment shown in figure 6.2a. Cliffs, steps, and ditches are types of negative obstacles that cannot be traversed by the robot. In [63], Jiang et al. claim that it is possible to detect this with UWB signals of various carrier frequencies. The experiments carried out



for this thesis however did not show the same signal behaviour and it was not possible to reliably detect cliffs.

The Virtual Reality scan was carried out in the standard configuration with a horizontal, slightly squinting, and not downwards angled sensor. The assumption was that a part of the strong signal in the 10 cm to 20 cm range were reflections from the floor, that should disappear when the floor ends at a cliff. Visual inspection of the range profile however shows only a extremely slight change in signal, e.g. at cross range 0.8 m to 1.1 m, down range 0.2 m to 0.25 m., where the floor could not reflect due to the radar sensor overhanging the cliff.

The Washroom scan has the sensor mounted in a vertical configuration and downward facing (see figure 6.2b) instead to increase sensitivity to echo scattering from below. The echo intensity for cross range 3.5 m to 4.5 m is indeed just barely lower than 0 m to 2 m, which matches up to where the sensor was over the edge and over floor, respectively.

TODO annotation-arrow pointing to darker region?

To limit the transmit crosstalk's blinding effect, the sensor was mounted on a much higher position (on the RGBD camera mount) in the Xray Room scan. One effects is that the robot chassis itself is constantly visible at a distance of 0.35 cm down range. At 0.45 cm down range, the floor echo is visible. There is one dip in intensity at 2.5 m cross range, where the sensor was not over floor. Overall the signal is however not as conclusive as in the Washroom scan.

TODO annotation-arrow pointing to darker region?

Maybe the signal could be improved with improved background subtraction (see #REF). However, the three scans show that it is very hard to detect negative obstacles with this sensor. A radar sensor of this type will hence not be a viable replacement for regular cliff detection sensors like the floor facing infrared distance sensors in the Kobuki base.

6 Discussion and Comparison

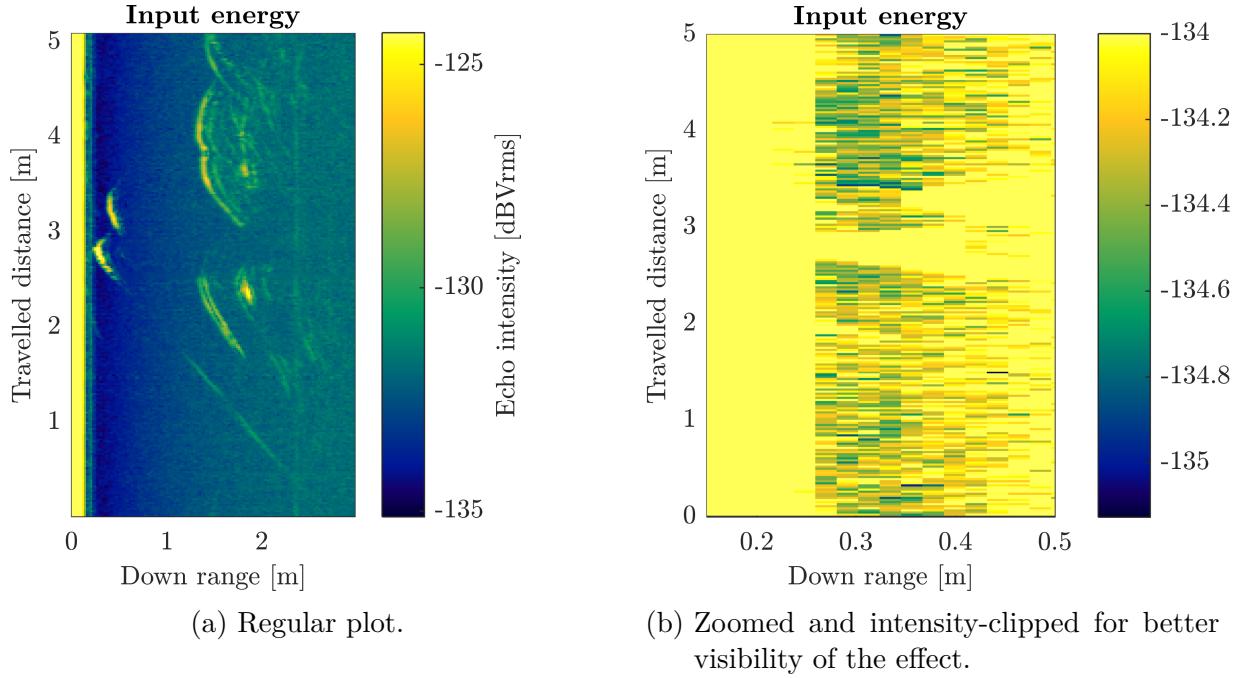


Figure 6.3: Input energy for the Washroom scan. Lower echo intensity at negative obstacles is hardly visible in the standard configuration.

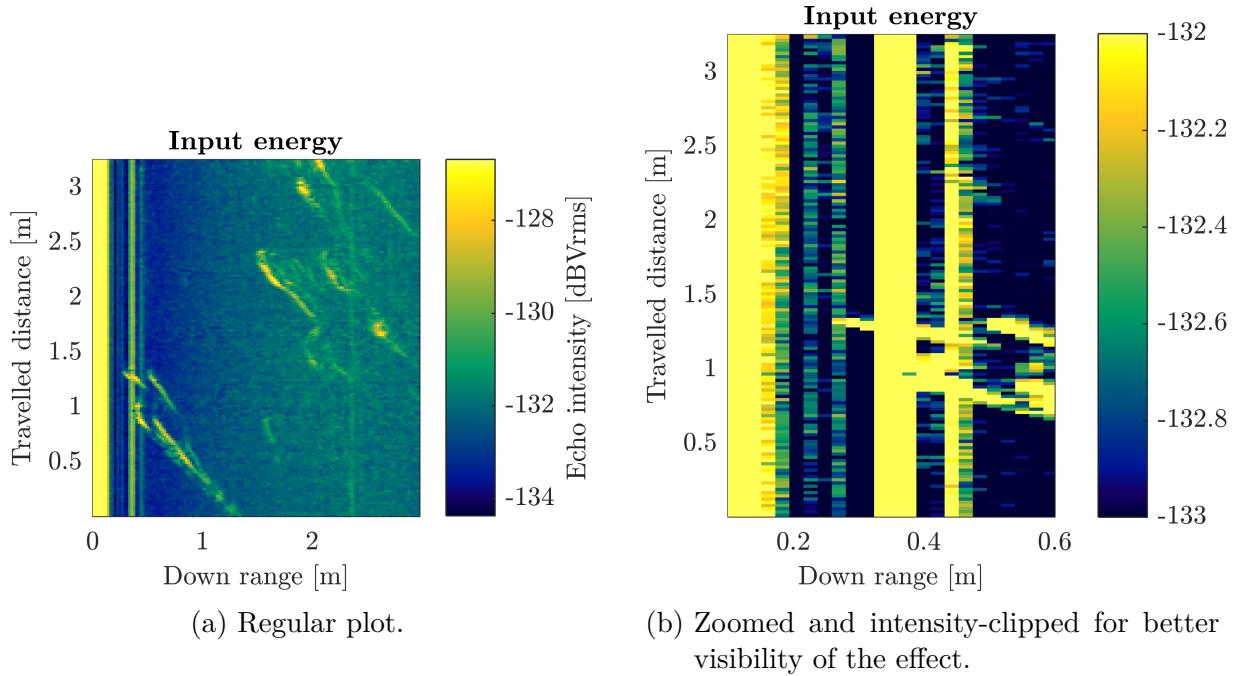


Figure 6.4: Input energy for the XRay Room scan. Negative obstacles effect is not better with a higher-mounted vertical configuration.

6.1.7 Cable detection

Cables on the floor are another interesting target that falls into the category of obstacles being a very common occurrence in the real world, but are hard to detect with conventional obstacle sensors. The Y (Is There A Cable On The Floor) scan deals with the detection of this kind of obstacle. For this, the same camera-mounted vertical configuration as in X Ray Room was used. Again, there is a constant robot chassis echo at down range 0.35 m. As the robot is driving closer towards the power cable on the floor, the cable's echo is visibly coming closer before it disappears under the robot's chassis. The echos at 0.9 m down range show the two can towers at the end of the cable.

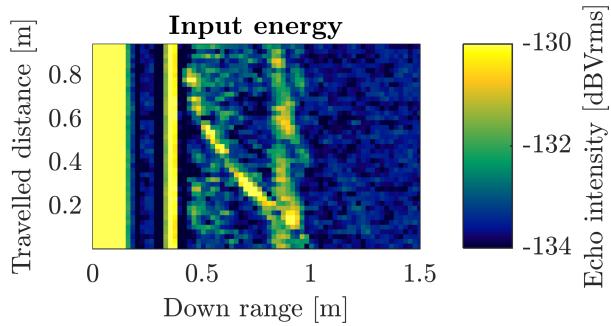


Figure 6.5: Input energy of the Y (Is There A Cable On The Ground) scan. The center arc represents the echo of a cable on the floor as the robot approaches.

Both the X Ray Room and Y (Is There A Cable On The Floor) introduce a new geometry that lifts the radar sensor out of the two dimensional mapping plane. The geometry is better described with a 3D case, for which more than

6.1.8 Minimum distance

The constant noise from the transmission crosstalk leads to a high minimum detection distance as explained in #REF. The effect is that targets can not be projected onto the map if the robot is too close to them. This is an issue in the Sauna scan, where the Glass wall right at the beginning of the robot path cannot be mapped in its entirety.

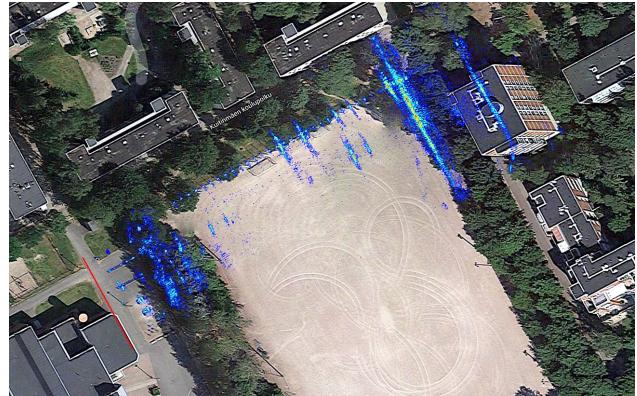
6.2 Comparison with other mapping techniques

While some of the radar reprojection maps speak for themselves, they make more sense when compared to other mapping techniques. In the following, SAR techniques, Laser slam, and RGBD slam are compared to the radar reprojection.

6 Discussion and Comparison



(a) Forstén’s SAR setup. Source: [62]



(b) Forstén’s SAR imagery agrees well with aerial imagery. Source: [62]

Figure 6.6: After omega-k processing and autofocusing, Forstén’s synthetic aperture radar can create nice-looking broadside SAR imagery.

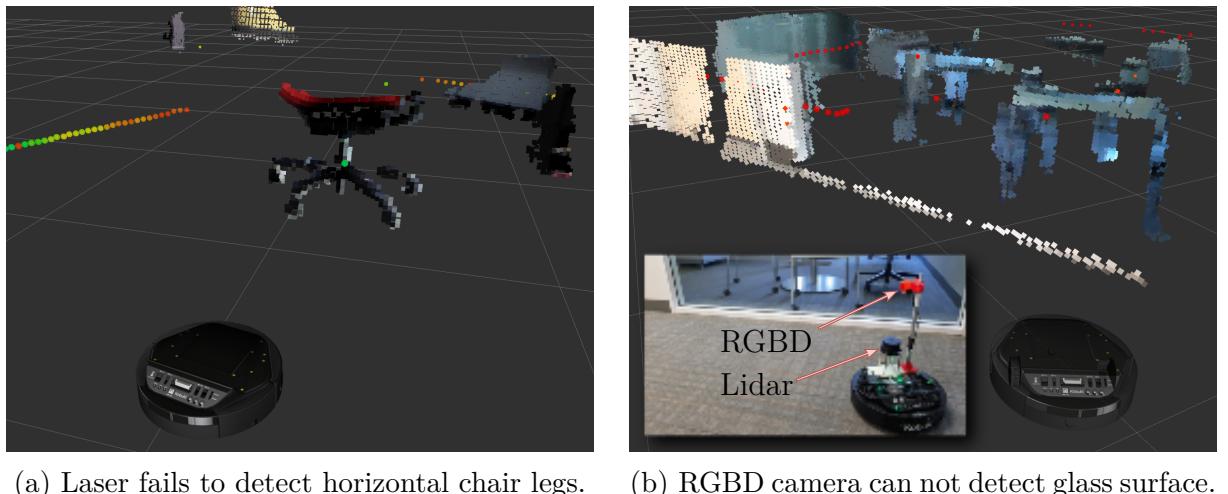
6.2.1 SAR

Synthetic aperture radars make a lot of sense in other applications where the radar is moved over or through a map. The big difference to this application is that “professional” SAR applications have radar sensors that sit in vehicles that are not in the mapping plane. Airplanes, satellites and even Submarines scan the earth like that.

There are a few examples for UWB radars being moved sideways (usually on a rail) in an effort to scan a scene with synthetic aperture radar. Gregory L. Charvat’s “tin can” radar [49] might be the most famous one, with many examples at <http://glcharvat.com/shortrange/>. Another great resource was Henrik Forstén’s Homemade Synthetic Aperture Radar, documented in [62] (see figure 6.6). He used an Omega-k algorithm [21] and Stolt interpolation [9] to correct the range migration arcs.

Forstén was able to greatly improve his data quality by use of an minimum-entropy based auto-focusing algorithm. The trick with this is that the radar needs to move in a very straight line, where the “error in path linearity should be around less than tenth of a wavelength” [62]. In Forstén’s radar, this is about 5 mm. However with the 60 GHz Omnidaradar this is around 0.5 mm. Keeping a straight line with less than half a millimeter of linearity error is not realistically achievable on the Kobuki platform.

One big inherent problem with synthetic aperture radar algorithms is that basically all of them assume the radar to move in a straight line. While changing the squint angle helps to deal with issues such as earth curvature in satellite applications, SAR with curved or even arbitrary paths is a challenging topic, particularly because auto-focusing, which again relies on phase information, becomes more difficult [8].



(a) Laser fails to detect horizontal chair legs.

(b) RGBD camera can not detect glass surface.

Figure 6.7: RViz screenshots showing traditional sensor shortcomings.

6.2.2 RGBD

The Kobuki robot was also carrying an Astra Orbbec Mini¹⁾ structured light depth camera. Using the Rtabmap [53] ROS package²⁾, some 3D scans of the office environment were made. section 2.1 already showed the shortcomings of lidar and RGBD sensors, but in this section the same figures 6.7a and 6.7b are compared with radar reprojection.

Figure 6.8 shows clearly that the radar reprojection contains horizontal chair legs, while the lidar scan does not. The scene in figure 6.7b matches with the Underground scan, where the glass wall is clearly visible.

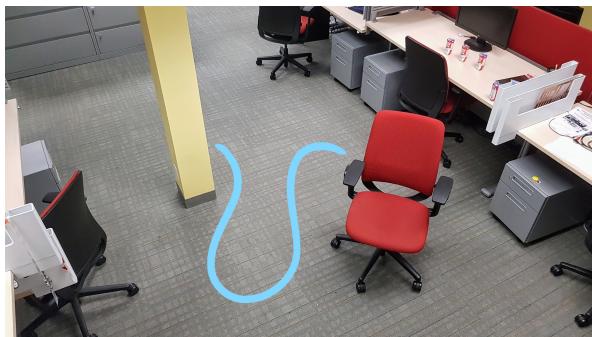
6.2.3 Lidar

As stated in section 4.1.1, the Kobuki robot used in the experiments was equipped with an RPLidar and a computing platform powerful enough to perform slam. Lidar slam is the go-to, standard approach when it comes to mapping the environment around a robot. After years of research and product development, even relatively cheap lidar systems like the RPLidar have acceptable range resolution. While they can't provide ground truth data (see problems with lidar data in section 2.1), it makes sense to compare the radar reprojection maps with laser scan maps. This is why all scans after Mancave have lidar maps associated. Note that there is a bug in the rosbag to Matlab conversion that sometimes causes the complete map to be rotated against the lidar map by some degrees.

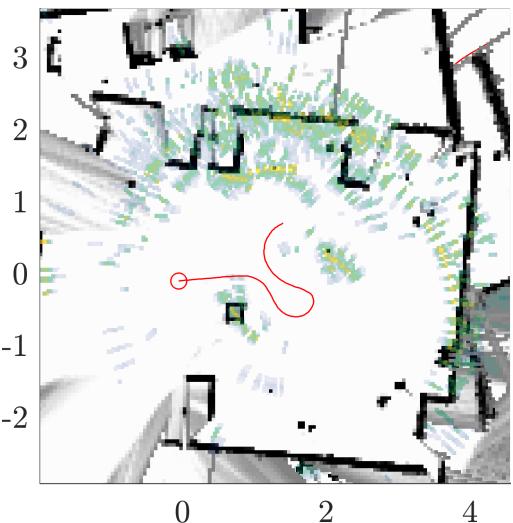
¹⁾ <https://orbbec3d.com/astra-mini/>

²⁾ http://wiki.ros.org/rtabmap_ros

6 Discussion and Comparison



(a) View of the Torturechamber scan environment. Blue path is the robot's path.



(b) Torturechamber reprojection map over lidar slam gridmap.

Figure 6.8: In the Torturechamber scan, horizontal chair legs are visible in the reprojection map but not in the lidar scan.

7 Conclusion

While the presented implementation was shown to work, there are many areas which in which it could be extended with interesting features. Some worthwhile goals for future work on the topic are outlined in section 7.1.

7.1 Future Work

7.1.1 Dynamic target rejection

As stated in section 3.8, reprojection mapping only works for static environments. It is however not difficult to relax this strict requirement a little: Targets moving faster than the robot itself can easily be detected and ignored. These dynamic targets should of course be tracked with the peak matching of the peak gradient algorithm, to make sure they are ignored even after they slow down to below robot speed.

7.1.2 Online mapping

The proof of concept implementation processes pre-recorded data. However the algorithm is by no means limited to offline processing. Being very much iterative and range scan line based it requires only the knowledge of current and past, but not future scans. A live version of the algorithm was not built, because the implementation was done in Matlab, which does not run on arm processors natively. Matlab's Robotics Toolbox does include a way to receive ROS messages live, but it is very slow and would miss a lot of messages. This was tested by replaying a rosbag, which works only at less than 10% replay speed. This means that a 6 minute recording takes around 60 minutes to process. On the other hand, just reading in all the messages in a rosbag takes around 3 minutes, which is the reason the implementation was not done live. In a real system (i.e. not replayed from a rosbag) sending raw scan messages over the network requires a lot of bandwidth, so the sampling frequency also drops considerably.

An online system would probably have to be designed as a ROS node that runs on the embedded platform.

Another topic that needs to be looked at for an online version is the size of the reprojection map. It should automatically expand if a wider area is necessary. This could be handled in

7 Conclusion

a nice way with ROS’s `nav_msgs/OccupancyGrid` messages and/or the [grid_map package](#). This would also allow pretty and useful visualizations with RViz.

7.1.3 ROS nodelet

The `omniradar_node` ROS node spends quite some time on copying the radar echo into the ros message that is to be sent out. As Austin Hendrix points out in *ROS answers*¹⁾, “ROS doesn’t provide intra-process, zero-copy publishing. Nodelets can be run multi-threaded, so it is possible to have zero-copy between different nodelets within a single nodelet manager”. Recording a rosbag still involves copying the data, but in an online system a nodelet-based reprojecting algorithm can be expected to bring a reasonable performance improvement.

7.1.4 Auto-thresholding

As section 4.5.5 describes, the proof-of-concept implementation uses a constant threshold as minimum peak height. There has been extensive work done to create adaptive thresholds with a class of algorithms called constant false alarm rate (CFAR) [20, 34]. It would be interesting to implement this to make the reprojection robust against noise at unexpected levels.

7.1.5 Realtime

An obstacle sensor’s job is to provide information on impeding collisions before it’s too late. Thus it would be great to have the system run under realtime constraints, so it can guarantee range scanning and reprojection mapping to be finished within a known time frame.

7.1.6 Interference Investigation

Since the 60 GHz band is ISM, other sources can be present, e.g. the IEEE 802.11ad standard [50] (“WiGig”) which enables very high throughput wireless LAN operation in frequencies around 60 GHz. Commercial products using WiGig are already available and could cause some interference with the 60 GHz radar sensor. Detecting or avoiding interference in the range signal would be an important topic if this causes a lot of trouble.

¹⁾ <https://answers.ros.org/question/208801/how-to-have-no-copy-publishing-over-multiple-cores/?answer=208805#post-id-208805>

7.1.7 Ultrasound

Usually, ultrasound sensors measure the distance to the closest object. However, K.C. Lee's project log of their "Sonar for the visually impaired" project [64] shows how cheap sensors can be hacked to read a range profile that looks very similar (see figure figure 7.1a) to what is used for the radar reprojection method. It might be possible to adapt the algorithms in this thesis to use ultrasound sensors.

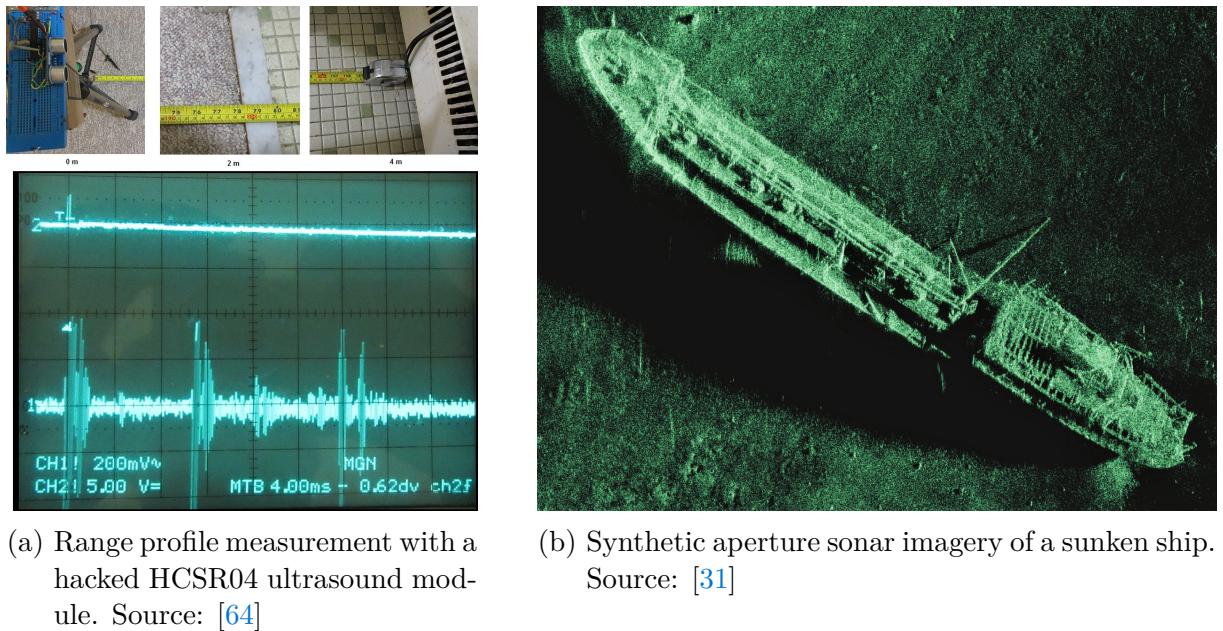


Figure 7.1: Ultrasound might be a very interesting data source for reprojection mapping.

Most ultrasound range sensors use the time of flight of a pulsed echo, but FMCW-based ultrasound modules have been proposed [47] recently. With a range resolution of

$$dR = \frac{c_{sound,air}}{2BW} = \frac{330 \text{ m/s}}{2 \cdot 12.5 \text{ kHz}} = 1.32 \text{ cm} \quad (7.1)$$

the proposed sensor would be comparable to the Omnidaradar sensor's range resolution and accuracy. However, for measurements to be very accurate the speed of sound in air must be known as it depends on humidity and temperature [4].

For forward-facing geometry DOA is necessary to resolve the sign of the reprojection angle. Sound waves do not carry phase information, but with a transducer array, direction of arrival can still be estimated [28].

Extension to ultrasound sensors would be a very interesting topic for further work. It might even be possible to use light waves with interferometric modulated flash lidars and the right optics.

7.2 Conclusion

This thesis showed that mobile robots still struggle to detect some obstacles which are invisible to their conventional sensors. The fact that these are not special edge cases but very common objects like chair legs and windows highlights how important it is to improve mapping in these real-world environments. Radar sensing fills a gap in the design space by exposing these hidden obstacles. It is already a tried and tested technology and has been deployed in many diverse fields, including far-range military reconnaissance, speed limit enforcement, and weather forecasting for decades. However, with new advances in system integration, miniaturization, and cost optimization, ultra-wide bandwidth radar sensing becomes very interesting even for consumer-grade ground robots.

With reprojection mapping, this thesis proposes a novel method for scanless radar-based mapping that further reduces the barrier that kept radar sensors out of robots for so long by obviating the need to actively steer the radar beam via mechanical scanning or beamforming. The most limiting factor in the application is the need for a static environment to assert that any target can be localized through its Doppler speed. In practice this constraint is not too hard to meet because many indoor robots are designed to operate unsupervised in environments without moving objects, like an office at night or an apartment during working hours.

Even though the ideas behind reprojection mapping are not complex or require complicated math, this kind of scanless mapping has never been shown before. One can assume the reason to be that in high-performance, high-cost radar mapping applications it is easier to use more specifically designed expensive or complex scanning sensors. On the other end of the spectrum, low-cost and high-volume products could use cheaper and proven traditional sensors like laser scanners. Only the advent of cheaper and smaller radar range sensors allowed the expansion of robotic design space into the corner in which reprojection mapping becomes useful.

That reprojection mapping is useful, and moreover that it works as designed, is shown with a proof-of-concept implementation. With this implementation, several obstacle maps were created from the radar scans of the experiments for this thesis. The maps prove that some previously undetectable obstacles, like glass walls and office chair legs, can now be detected and mapped.

The next step on the way to a mobile indoor robot proficiently navigating real-world environments is the implementation of an online version of reprojection mapping. This will also show if the results need to be further improved with more advanced noise rejection.

A libomniradar installation

If the driver source is available, installing the files can be accomplished from the source directory with

```
mkdir build && cd build  
cmake -DCPP_BINDINGS=ON -DMATLAB_BINDINGS=OFF ..  
make  
sudo make install
```

The driver can then be included in an application. Note that since it is dynamically linked, the ftd2xx, pthread and dl libraries dependencies also need to be linked. In a Catkin¹⁾-CMakeLists.txt this would look like

```
target_link_libraries(  
    ${PROJECT_NAME}_node  
    omniradar  
    ftd2xx  
    pthread  
    dl  
    ${catkin_LIBRARIES}  
)
```

A.1 VCO tuning curve import

Section 4.3.1 mentions that the VCO tuning curve needs to be known in order to correctly predistort the VCO. Omnidaradar provides Matlab scripts that can measure this curve, the data just needs to be converted to a format that is appropriate for a C++ application using libomniradar, such as the `omnidaradar` rosnode. The easiest way to bring this tuning curve into the C++ domain is to print²⁾ it as C style array, with

```
[ '#pragma once' 10 'std::vector<double>' 10 ...  
 'vco_tune {', sprintf('%.100g, ', VC0tune), '};' ]
```

and saving the resulting string as `vco_tune.h` include file.

¹⁾ Catkin is the CMake-based ROS build system

²⁾ [‘A’ 10 ‘B’] is a quick way to print a ‘
n’ (newline character) between A and B

B Usage

The following list shows the procedure for acquiring a radar reprojection map with the setup and implementation described in section 4.1 and chapter 4, respectively.

1. Start robot, connect three sessions with `ssh zero2-pa`
2. Start ROS core with `roscore`
3. Start the node, with keyboard teleoperation and Cartographer Laser Slam: `roslaunch omniradar omniradar_teleop_lidarslam.launch`
4. Use `rqt` with the *Dynamic Reconfigure* plugin to set the Omniradar node to generate the preferred number of sweeps and sweep duration. The configuration string can also be changed. The defaults are fine (One 5 ms sweep)
5. `rosbag record /omniradar_node/radar_raw /odom /tf /map -O scan` will record a rosbag “scan.bag” containing all ROS messages with radar data, Kobuki odometry and slam map and transforms from Cartographer.
6. In the terminal running `roslaunch`, use the arrow keys to move the robot around (\uparrow and \downarrow to increase and decrease speed, \leftarrow and \rightarrow to increase and decrease rotation speed). E resets speed to zero and makes the robot halt.
7. After recording some interesting data, stop the rosbag record (*Ctrl+C*). Open a new terminal on your local machine and run `ssh zero2-pa "tar zcf - scan.bag" | tar zxf -`. The rosbag will be transferred to your machine. Using ssh with the tar pipe is the fastest way to transfer the data (around 100 Mbps on the BSH wifi). Compressing first (`rosbag compress scan.bag`) and then sending takes a while on the not-so-powerful Odroid platform.
8. Optionally filter out unwanted transforms from the rosbag to speed up later processing: `rosbag filter scan.bag scan_filtered.bag '(topic == "/tf" and m.transforms == "odom") or topic != "/tf"`
9. In Matlab, run `radar_data = radar_bag2array("/path/to/your/scan.bag");`. This will read the bag sequentially into memory and extract the data: The robot position is recorded from odometry information and corrected using the `/map → /odom` transform as reported by slam localization. Cross range mileage is calculated as cumulative sum of distance between radar positions (as the radar is not mounted over the robot’s rotation center, the radar mileage is different from robot mileage as soon as any rotational velocity is present). Lastly, all values are interpolated at the radar message timestamps. It is a good idea to save the function’s output, using `save('radar_data/radar_data_scan.mat', 'radar_data')`
10. The radar data can now be analyzed. The `plot_world_projection` script can be used to get a good overview over raw data, Doppler speeds, direction of arrival, and reprojection map.

B Usage

11. To compare radar reprojection map and laser slam map, try the `test_slam_overlay` with the correct bag filename (the lidar map is extracted from that bag)

C List of Scans

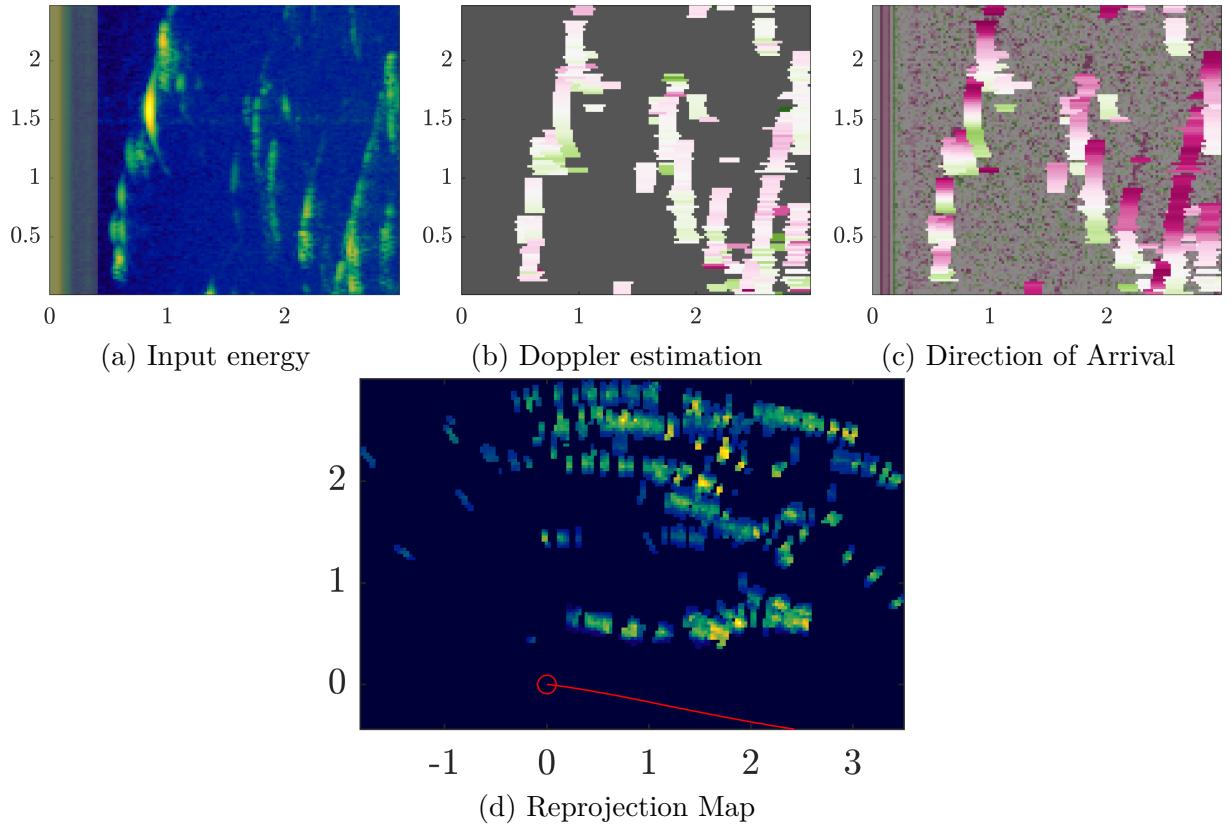


Figure C.1: Attic scan

TODO put all the scans here

C List of Scans

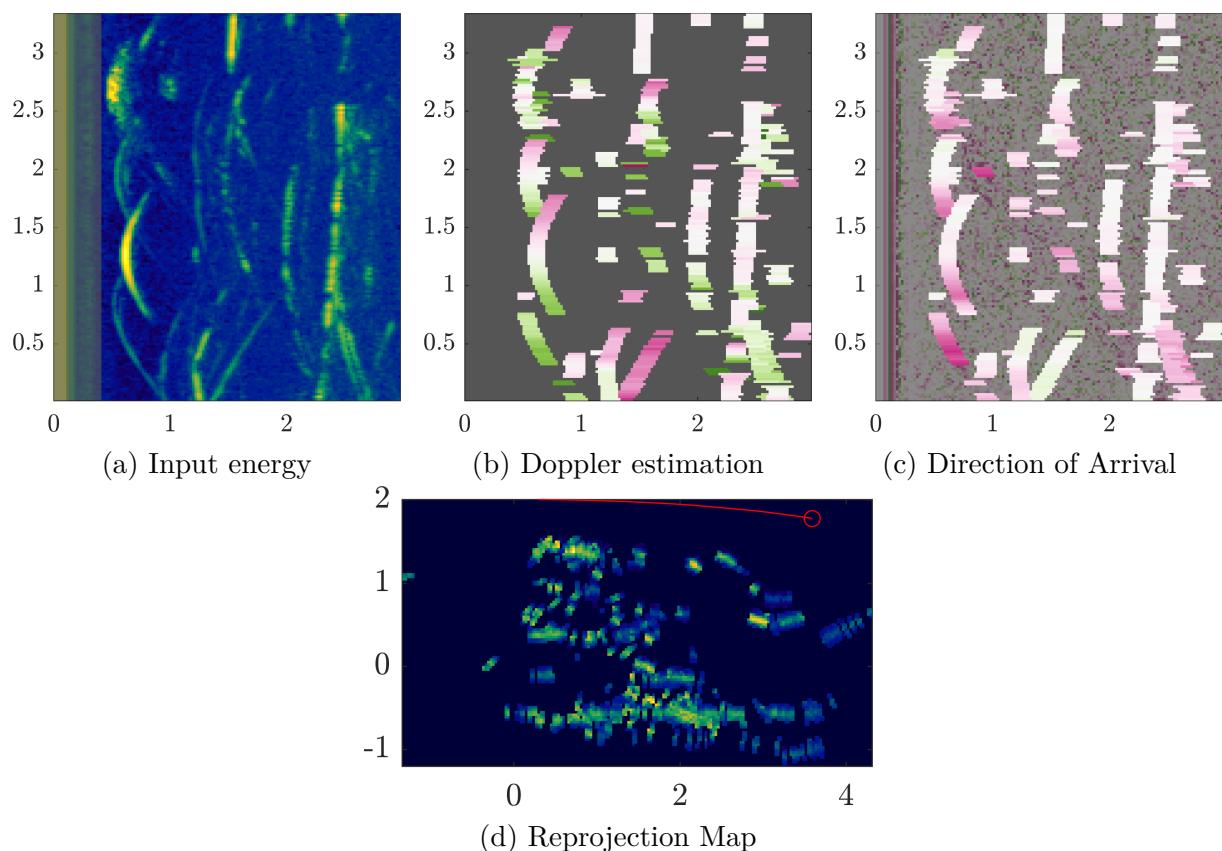
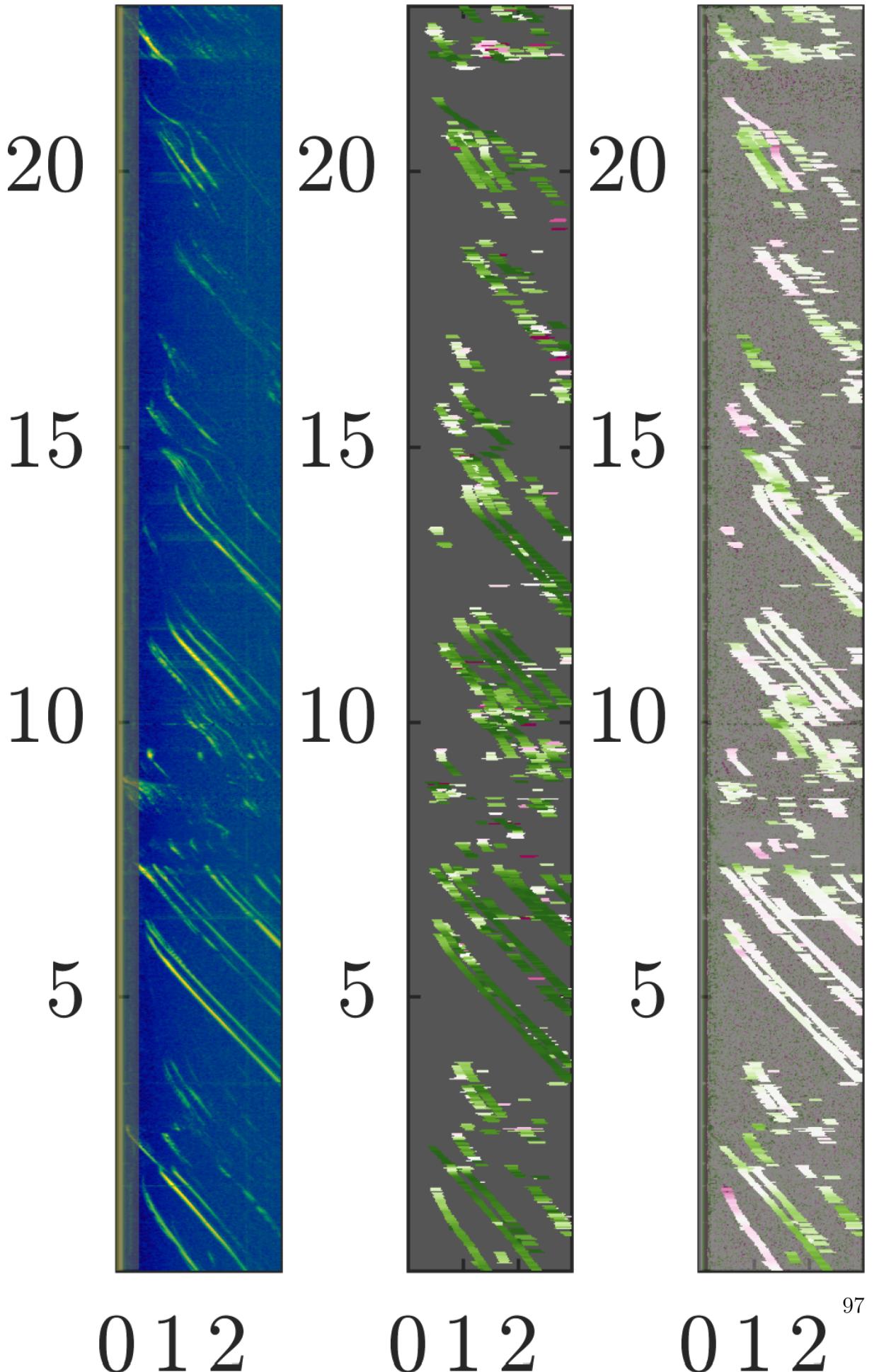


Figure C.2: Basement scan



(a) Input energy

(b) Doppler estimation

(c) Direction of Arrival

C List of Scans

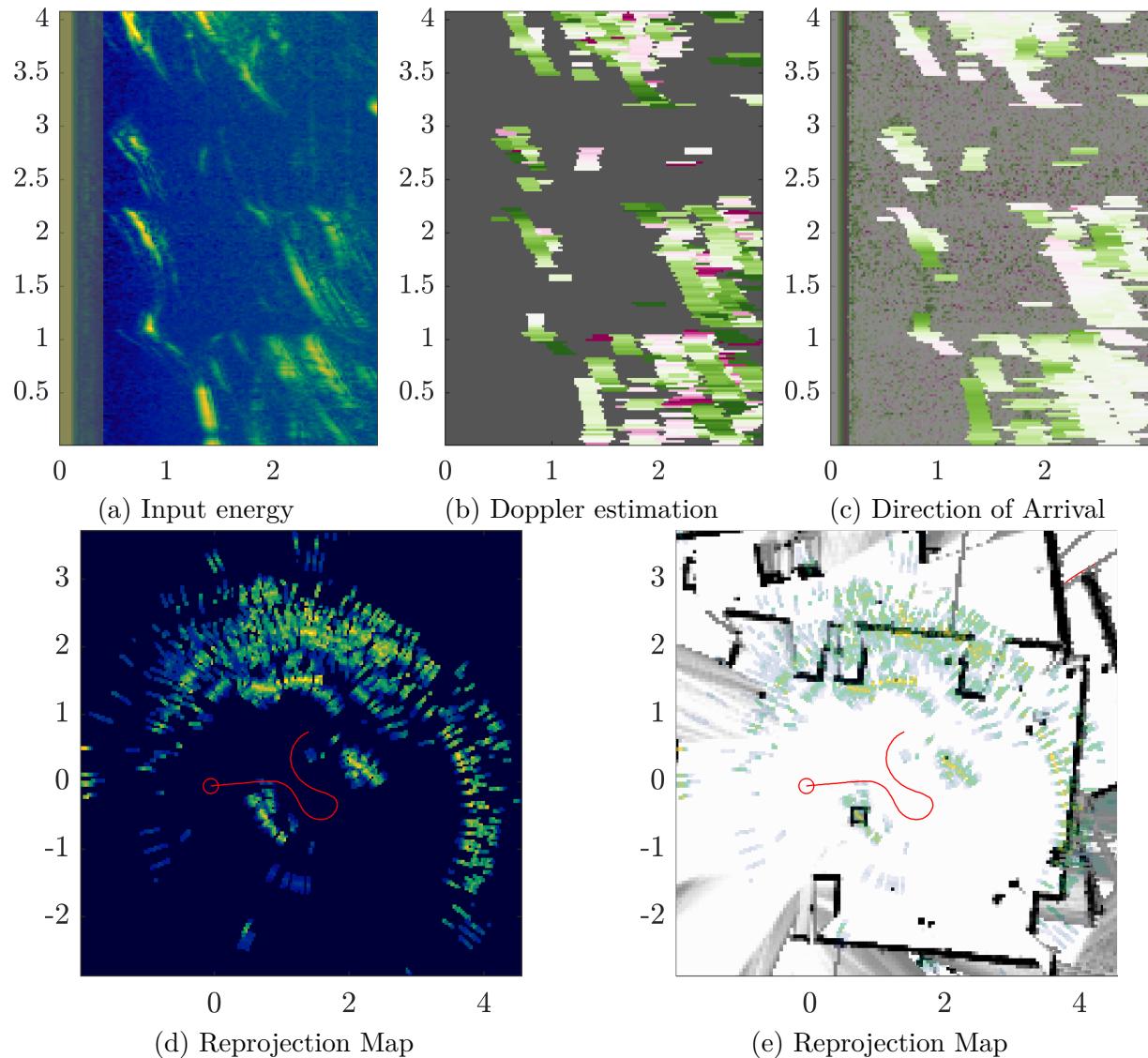


Figure C.4: Torturechamber scan

List of Figures

2.1	Screenshots showing traditional sensor shortcomings.	4
2.2	Pulse radars measure the time between transmission and reception of a short EM burst. Adapted from [34] p. 52.	7
2.3	Simplified FMCW architecture. Adapted from [57].	7
2.4	FMCW radars detect targets in the beat frequency, which is a frequency mix of the transmitted and received modulation. Adapted from [34] p. 57.	8
2.5	Target motion introduces a Doppler shift in the received signal, which changes the beat frequency during up- and down-sweeps. Adapted from [34] p. 57.	10
2.6	By reflecting a portion of the transmission signal back to the radar, each target (green, yellow and blue lines) contributes a frequency corresponding to its range in the beat spectrum / range response.	11
2.7	Beat signal and spectrum / range profile in a real-world measurement.	12
2.8	Phase difference and I/Q signal	13
2.9	RF attenuation in different media.	14
2.10	WiGig Channel Plan and Frequency Allocations by Region. Source: [39]	15
2.11	Illustration of different SAR operation modes which are used to increase the swath width (ScanSAR) or improve azimuth resolution (Spotlight) compared to Stripmap mode. Adapted from [46]	17
2.12	Summary of SAR processing steps where the range compressed data result from a convolution of the raw data with the range reference function. In a second step the azimuth compression is performed through a convolution with the azimuth reference function, which changes from near to far range. Source: [46]	18
2.13	Scanning radars.	20
2.14	Superimposed radar and lidar plan position indicator display.	21
2.15	Comparison of radar and lidar occupancy gridmaps. Source: [58]	21
3.1	Reprojection geometries	25
3.2	Ambiguity resolution through target tracking during turns	25
3.3	3D geometry and ambiguity reduction through 2D-DOA.	26
3.4	3D rendering of a multi-sensor scenario: Two sensors are enough to single out a target's location, while a trilateration approach would need a third sensor to narrow down the location from the intersection circle (green) of the trilateration spheres (orange and blue)	27

List of Figures

3.5 Walls reflect only at limited, material-dependent backscatter angles. They thus appear as point targets with a Doppler speed equal to the radar's motion component orthogonal to the wall.	29
4.1 The supporting hardware	32
4.2 Vayyar's Walabot Pro sensor is claimed to have Target localization and tracking. Source: https://api.walabot.com/_features.html#_examples	36
4.3 Static range test with the Walabot sensor	38
4.4 Dynamic range test with the Walabot sensor	38
4.5 Decapped Omnidar IC (top left); five cent coin as size reference (top right); schematic with antenna locations (bottom left); and antenna directionality pattern of Omnidar's RIC60-A (bottom right). Adapted from [61] p.9	39
4.6 Radar sensor on mount and model of the mount.	40
4.7 Chirp efficiency η (measurement time per time spent) for various chirp lengths. For this data, $n_{chirp} = 2$	43
4.8 Effect of chirp length on SNR; with and without Horn extension. The range profiles depict the same scene and are recorded over 10 s each.	43
4.9 Schematic overview of the information flow from the hardware to the re-projection map	44
4.10 Omnidar rosnode node graph with leaf topics	47
4.11 Dynamic reconfigure options for the Omnidar rosnode	47
4.12 Range profile quality comparison with different averaging functions. Higher peak-to-trough ratio is better.	51
4.13 Peak detection and subsample peak interpolation to find peak distance between two range profiles	53
4.14 Attenuating effect of range compensation on transmit peak in a range profile	57
4.15 DOA of Attic scan. The DOA of higher peak values is applied over the FWHM, superseding the values of lower peak values.	58
4.16 The target echo at the closest range usually has the brightest intensity. This can lead to errors in Doppler speed estimation.	60
4.17 Range profile of Attic scan with color coded echo intensities at cross range over down range	61
4.18 Phase profile of Attic scan with color coded inter-antenna phase difference at cross range over down range	62
4.19 Direction of arrival estimation for Attic scan	63
4.20 Illustration of a pixel value (white circle) being distributed over the four nearest grid neighbors. The shade of the neighbor's area indicates the individual distribution weight.	66
4.21 Range compensation	67
4.22 Angle compensation factor	69
4.23 Effect of angle compensation and windowed angle compensation	71
5.1 RViz screenshots showing traditional sensor shortcomings.	74
5.2 Nirvana scan	77

List of Figures

6.1	Range profile and DOA of Sauna scan.	80
6.3	Input energy for the Washroom scan. Lower echo intensity at negative obstacles is hardly visible in the standard configuration.	82
6.4	Input energy for the XRay Room scan. Negative obstacles effect is not better with a higher-mounted vertical configuration.	82
6.5	Input energy of the Y (Is There A Cable On The Ground) scan. The center arc represents the echo of a cable on the floor as the robot approaches.	83
6.6	After omega-k processing and autofocusing, Forstén's synthetic aperture radar can create nice-looking broadside SAR imagery.	84
6.7	RViz screenshots showing traditional sensor shortcomings.	85
6.8	In the Torturechamber scan, horizontal chair legs are visible in the reprojection map but not in the lidar scan.	86
7.1	Ultrasound might be a very interesting data source for reprojection mapping.	89
C.1	Attic scan	95
C.2	Basement scan	96
C.3	Sauna scan	97
C.4	Torturechamber scan	98

List of Tables

4.3	File installation destinations of libomniradar	45
4.4	Comparison of mean types by RMS relativ to harmonic mean	52
5.2	Scan parameters	76

Bibliography

- [1] J. Capon. “High-resolution frequency-wavenumber spectrum analysis”. In: *Proceedings of the IEEE* 57.8 (1969), pp. 1408–1418. ISSN: 0018-9219. DOI: [10.1109/PROC.1969.7278](https://doi.org/10.1109/PROC.1969.7278). URL: <http://ieeexplore.ieee.org/document/1449208/>.
- [2] A Paulraj, R Roy, and T Kailath. “Estimation Of Signal Parameters Via Rotational Invariance Techniques - Esprit”. In: *Nineteenth Asilomar Conference on Circuits, Systems and Computers* (1985), pp. 83–89. ISSN: 1058-6393. DOI: [10.1109/ACSSC.1985.671426](https://doi.org/10.1109/ACSSC.1985.671426).
- [3] R. Schmidt. “Multiple emitter location and signal parameter estimation”. In: *IEEE Transactions on Antennas and Propagation* 34.3 (Mar. 1986), pp. 276–280. ISSN: 0096-1973. DOI: [10.1109/TAP.1986.1143830](https://doi.org/10.1109/TAP.1986.1143830). URL: <http://ieeexplore.ieee.org/document/1143830/>.
- [4] Dennis A Bohn. “Environmental effects on the speed of sound”. In: *Journal of the Audio Engineering Society* 36.4 (1988), pp. 223–231. URL: <http://www.rane.com/pdf/eespeed.pdf> http://bsd.rane.com/pdf/ranenotes/Enviromental_Effects_on_the_Speed_of_Sound.pdf <http://www.aes.org/e-lib/browse.cfm?elib=5156>.
- [5] International Telecommunications Union. *ATTENUATION BY ATMOSPHERIC GASES*. 1997. URL: https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.676-3-199708-S!!PDF-E.pdf.
- [6] David D. Ferris Jr. and Nicholas C Currie. “Microwave and millimeter-wave systems for wall penetration”. In: *Targets and Backgrounds: Characterization and Representation IV*. Ed. by Wendell R. Watkins and Dieter Clement. April. July 1998, pp. 269–279. ISBN: 0-8194-2824-8. DOI: [10.1117/12.327159](https://doi.org/10.1117/12.327159). URL: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=949255>.
- [7] M.W.M.G. Dissanayake et al. “A solution to the simultaneous localization and map building (SLAM) problem”. In: *IEEE Transactions on Robotics and Automation* 17.3 (June 2001), pp. 229–241. ISSN: 1042296X. DOI: [10.1109/70.938381](https://doi.org/10.1109/70.938381). URL: <http://ieeexplore.ieee.org/document/938381/>.
- [8] S.R.J. Axelsson. “Mapping performance of curved-path SAR”. In: *IEEE Transactions on Geoscience and Remote Sensing* 40.10 (Jan. 2002), pp. 2224–2228. ISSN: 0196-2892. DOI: [10.1109/TGRS.2002.803740](https://doi.org/10.1109/TGRS.2002.803740). URL: <http://ieeexplore.ieee.org/document/1105907/>.

Bibliography

- [9] Ian G Cumming and Frank H Wong. *Digital Processing of Synthetic Aperture Radar Data Algorithms and Implementation*. Norwood, MA, USA: Artech House, 2004, p. 660. ISBN: 9781580530583. URL: <https://pdfs.semanticscholar.org/23eb/33af4f0495edff01402ec8eb0.pdf>.
- [10] E. Jose and M.D. Adams. “Millimetre Wave RADAR spectra simulation and interpretation for outdoor SLAM”. In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*. IEEE, 2004, pp. 1321–1326. ISBN: 0-7803-8232-3. DOI: [10.1109/ROBOT.2004.1308007](https://doi.org/10.1109/ROBOT.2004.1308007). URL: <http://ieeexplore.ieee.org/document/1308007/>.
- [11] E. Jose and M.D. Adams. “Multiple line-of-sight predicted observations with millimetre wave RADAR for outdoor SLAM”. In: *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004*. Vol. 1. IEEE, 2004, pp. 155–160. ISBN: 0-7803-8653-1. DOI: [10.1109/ICARCV.2004.1468815](https://doi.org/10.1109/ICARCV.2004.1468815). URL: <http://ieeexplore.ieee.org/document/1468815/>.
- [12] E. Jose and M.D. Adams. “Relative radar cross section based feature identification with millimetre wave radar for outdoor SLAM”. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. Vol. 1. IEEE, 2004, pp. 425–430. ISBN: 0-7803-8463-6. DOI: [10.1109/IROS.2004.1389389](https://doi.org/10.1109/IROS.2004.1389389). URL: <http://ieeexplore.ieee.org/document/1389389/>.
- [13] E. Jose and M.D. Adams. “An augmented state SLAM formulation for multiple line-of-sight features with millimetre wave RADAR”. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 3087–3092. ISBN: 0-7803-8912-3. DOI: [10.1109/IROS.2005.1545232](https://doi.org/10.1109/IROS.2005.1545232). URL: <http://ieeexplore.ieee.org/document/1545232/>.
- [14] Marek Bury, Yevhen Yashchyshyn, and Jozef Modelska. “A Simple Approach for Elimination of Fixed Objects’ Reflections in UWB Imaging System”. In: *EUROCON 2007 - The International Conference on "Computer as a Tool"*. IEEE, 2007, pp. 299–302. ISBN: 978-1-4244-0812-2. DOI: [10.1109/EURCON.2007.4400484](https://doi.org/10.1109/EURCON.2007.4400484). URL: <http://ieeexplore.ieee.org/document/4400484/>.
- [15] C. Craeye and A. Boryssenko. “Direct radar positioning using strongly coupled ultra-wideband arrays”. In: *2008 IEEE Antennas and Propagation Society International Symposium*. Vol. 1. October. IEEE, July 2008, pp. 1–4. ISBN: 978-1-4244-2041-4. DOI: [10.1109/APS.2008.4619931](https://doi.org/10.1109/APS.2008.4619931). URL: <http://ieeexplore.ieee.org/document/4619931/>.
- [16] Piotr Gorniak and Wojciech Bandurski. “Direct Time Domain Analysis of an UWB Pulse Distortion by Convex Objects With the Slope Diffraction Included”. In: *IEEE Transactions on Antennas and Propagation* 56.9 (Sept. 2008), pp. 3036–3044. ISSN: 0018-926X. DOI: [10.1109/TAP.2008.928790](https://doi.org/10.1109/TAP.2008.928790). URL: <http://ieeexplore.ieee.org/document/4618693/>.

- [17] Andrew D Payne. “Development of a Real-Time Full-Field Range Imaging System”. PhD thesis. University of Waikato, Hamilton, New Zealand, 2008. ISBN: 9783642005770. DOI: [10.1007/978-3-642-00578-7_7](https://doi.org/10.1007/978-3-642-00578-7_7). URL: <http://hdl.handle.net/10289/3521>
- [18] R. Rouveure, M. O. Monod, and P. Faure. “Mapping of the environment with a high resolution ground-based radar imager”. In: *MELECON 2008 - The 14th IEEE Mediterranean Electrotechnical Conference*. IEEE, May 2008, pp. 822–828. ISBN: 978-1-4244-1632-5. DOI: [10.1109/MELCON.2008.4618537](https://doi.org/10.1109/MELCON.2008.4618537). URL: <http://ieeexplore.ieee.org/document/4618537/>.
- [19] Jochen Seitz et al. “UWB feature localization for imaging”. In: *Proceedings of The 2008 IEEE International Conference on Ultra-Wideband, ICUWB 2008* 2 (2008), pp. 199–202. DOI: [10.1109/ICUWB.2008.4653385](https://doi.org/10.1109/ICUWB.2008.4653385).
- [20] Merrill Ivan Skolnik. *Radar Handbook*. McGraw-Hill, 2008, p. 1328. ISBN: 978-0-07-148547-0. URL: <https://books.google.com/?id=76uF2Xebm-gC>.
- [21] Matthew A. Tolman. “A Detailed Look at the Omega-k Algorithm for Processing Synthetic Aperture Radar Data”. PhD thesis. Brigham Young University, 2008. URL: <http://scholarsarchive.byu.edu/cgi/viewcontent.cgi?article=2578&context=etd>.
- [22] Bu-Chin Wang. *Digital Signal Processing Techniques and Applications in Radar Image Processing*. Hoboken, NJ, USA: John Wiley & Sons, Inc., Aug. 2008, 375 pages. ISBN: 9780470377765. DOI: [10.1002/9780470377765](https://doi.org/10.1002/9780470377765). URL: <http://doi.wiley.com/10.1002/9780470377765>.
- [23] Tobias Deißler and Jörn Thielecke. “Feature based indoor mapping using a bat-type UWB radar”. In: *2009 IEEE International Conference on Ultra-Wideband*. Vol. 2009. IEEE, Sept. 2009, pp. 475–479. ISBN: 978-1-4244-2930-1. DOI: [10.1109/ICUWB.2009.5288802](https://doi.org/10.1109/ICUWB.2009.5288802). URL: <http://ieeexplore.ieee.org/document/5288802/>.
- [24] Franck Gerossier et al. “Trajectory-oriented EKF-SLAM using the Fourier-Mellin Transform applied to Microwave Radar Images”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2009, pp. 4925–4930. ISBN: 978-1-4244-3803-7. DOI: [10.1109/IROS.2009.5354548](https://doi.org/10.1109/IROS.2009.5354548). URL: <http://ieeexplore.ieee.org/document/5354548/>.
- [25] sarmap.ch. *Synthetic Aperture Radar and SARscape*. 2009. URL: <http://www.sarmap.ch/pdf/SAR-Guidebook.pdf>.
- [26] Tobias Deißler et al. “UWB radar object recognition for SLAM”. In: *Radar Symposium (IRS), 2010 11th International*. 2010, pp. 1–4. ISBN: 978-1-4244-5613-0.
- [27] Ebi Jose et al. “Predicting Millimeter Wave Radar Spectra for Autonomous Navigation”. In: *IEEE Sensors Journal* 10.5 (May 2010), pp. 960–971. ISSN: 1530-437X. DOI: [10.1109/JSEN.2009.2037013](https://doi.org/10.1109/JSEN.2009.2037013). URL: <http://ieeexplore.ieee.org/document/5438923/>.

Bibliography

- [28] Vitaliy Kunin. "Sound and Ultrasound Source Direction of Arrival Estimation and Localization". PhD thesis. Illinois Institute of Technology, 2010, p. 136. URL: <http://www.ece.iit.edu/~ecasp/publications/Thesis/Vitaliy%20Kunin%20MS%20Thesis.pdf>.
- [29] Stefano Marano et al. "NLOS identification and mitigation for localization based on UWB experimental data". In: *IEEE Journal on Selected Areas in Communications* 28.7 (Sept. 2010), pp. 1026–1035. ISSN: 0733-8716. DOI: [10.1109/JSAC.2010.100907](https://doi.org/10.1109/JSAC.2010.100907). URL: <http://ieeexplore.ieee.org/document/5555901/>.
- [30] John Mullane et al. "X-band radar based SLAM in Singapore's off-shore environment". In: *2010 11th International Conference on Control Automation Robotics & Vision*. IEEE, Dec. 2010, pp. 398–403. ISBN: 978-1-4244-7814-9. DOI: [10.1109/ICARCV.2010.5707835](https://doi.org/10.1109/ICARCV.2010.5707835). URL: <http://ieeexplore.ieee.org/document/5707835/>.
- [31] Roy Edgar Hansen. *Introduction to Synthetic Aperture Sonar*. InTech, Sept. 2011. DOI: [10.5772/23122](https://doi.org/10.5772/23122). URL: <http://www.intechopen.com/books/sonar-systems/introduction-to-synthetic-aperture-sonar>.
- [32] Pavlo Molchanov et al. "Ground moving target classification by using DCT coefficients extracted from micro-Doppler radar signatures and artificial neuron network". In: *2011 MICROWAVES, RADAR AND REMOTE SENSING SYMPOSIUM*. IEEE, Aug. 2011, pp. 173–176. ISBN: 978-1-4244-9641-9. DOI: [10.1109/MRRS.2011.6053628](https://doi.org/10.1109/MRRS.2011.6053628). URL: <http://ieeexplore.ieee.org/document/6053628/>.
- [33] P.O. Molchanov et al. "Target classification by using pattern features extracted from bispectrum-based radar Doppler signatures". In: *Radar Symposium (IRS), 2011 Proceedings International* (2011), pp. 791–796.
- [34] Martin Adams et al. *Robotic Navigation and Mapping with Radar*. Artech House, 2012, p. 346. ISBN: 9781608074822. URL: <https://books.google.com/books?id=4sw8IpAjhsC>.
- [35] Tobias Deißler et al. "Infrastructureless indoor mapping using a mobile antenna array". In: *2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*. 2012, pp. 36–39. ISBN: 9783200023284.
- [36] M. Segura, V. Mut, and C. Sisterna. "Ultra wideband indoor navigation system". In: *IET Radar Sonar & Navigation* 6.5 (2012), p. 402. ISSN: 17518784. DOI: [10.1049/iet-rsn.2011.0352](https://doi.org/10.1049/iet-rsn.2011.0352). URL: <http://digital-library.theiet.org/content/journals/10.1049/iet-rsn.2011.0352>.
- [37] Yujin Robot. *Kobuki mechanical drawing*. 2012. URL: http://files.yujinrobot.com/kobuki/hardware/drawings/pdf/kobuki_base_01.pdf.
- [38] M. Adams, J. Mullane, and Ba-Ngu Vo. "Circumventing the Feature Association Problem in SLAM". In: *IEEE Intelligent Transportation Systems Magazine* 5.3 (2013), pp. 40–58. ISSN: 1939-1390. DOI: [10.1109/MITS.2013.2260596](https://doi.org/10.1109/MITS.2013.2260596). URL: <http://ieeexplore.ieee.org/document/6565468/>.

- [39] Agilent Technologies. *Wireless LAN at 60 GHz - IEEE 802.11ad Explained*. 2013. URL: <http://www.cs.odu.edu/~cs752/papers/milli-008.pdf%20http://cp.literature.agilent.com/litweb/pdf/5990-9697EN.pdf>.
- [40] Tobias Deißler and Jörn Thielecke. “Fusing odometry and sparse UWB radar measurements for indoor slam”. In: *2013 Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. IEEE, Oct. 2013, pp. 1–5. ISBN: 978-1-4799-0778-6. DOI: [10.1109/SDF.2013.6698259](https://doi.org/10.1109/SDF.2013.6698259). URL: <http://ieeexplore.ieee.org/document/6698259/>.
- [41] DesignK. *iClebo Kobuki Rollup Banner*. 2013. URL: http://files.yujinrobot.com/kobuki/marketing/icra_roscon/rollup_banner_0418_ver%20a1.pdf.
- [42] Afzal Godil, Roger Eastman, and Tsai Hong Hong. *NISTIR 7923: Ground Truth Systems for Object Recognition and Tracking*. 2013. DOI: [10.6028/NIST.IR.7923](https://doi.org/10.6028/NIST.IR.7923). URL: http://ws680.nist.gov/publication/get_pdf.cfm?pub_id=913121.
- [43] Armin Hornung et al. “OctoMap: an efficient probabilistic 3D mapping framework based on octrees”. In: *Autonomous Robots* 34.3 (Apr. 2013), pp. 189–206. DOI: [10.1007/s10514-012-9321-0](https://doi.org/10.1007/s10514-012-9321-0). URL: <http://link.springer.com/10.1007/s10514-012-9321-0%20https://octomap.github.io/>.
- [44] V Krishnaveni et al. “Beamforming for Direction-of-Arrival (DOA) Estimation-A Survey”. In: *International Journal of Computer Applications* 61.11 (2013), pp. 975–8887. URL: <http://research.ijcaonline.org/volume61/number11/pxc3884758.pdf>.
- [45] Jan Willem Marck et al. “Indoor Radar SLAM A radar application for Vision and GPS Denied Environments”. In: *Radar Conference, Proceedings Of the 10th European Figure 1* (2013), pp. 471–474.
- [46] Alberto Moreira et al. “A tutorial on synthetic aperture radar”. In: *IEEE Geoscience and Remote Sensing Magazine* 1.1 (Mar. 2013), pp. 6–43. ISSN: 2168-6831. DOI: [10.1109/MGRS.2013.2248301](https://doi.org/10.1109/MGRS.2013.2248301). URL: https://people.eecs.ku.edu/~callen58/826/826_SARtutorial_2013.pdf%20http://ieeexplore.ieee.org/document/6504845/.
- [47] Luigi Battaglini et al. “A low-cost ultrasonic rangefinder based on frequency modulated continuous wave”. In: *20th IMEKO TC4 Symposium on Measurements of Electrical Quantities: Research on Electrical and Electronic Measurement for the Economic Upturn, Together with 18th TC4 International Workshop on ADC and DCA Modeling and Testing, IWADC 2014* (2014), pp. 1122–1126. URL: <http://www.imeko.org/publications/tc4-2014/IMEKO-TC4-2014-344.pdf%20http://www.scopus.com/inward/record.url?eid=2-s2.0-84918821736&partnerID=tZOTx3y1>.
- [48] Aimeric Bisognin et al. “3D printed plastic 60 GHz lens: Enabling innovative millimeter wave antenna solution and system”. In: *2014 IEEE MTT-S International Microwave Symposium (IMS2014)*. IEEE, June 2014, pp. 1–4. ISBN: 978-1-4799-3869-8. DOI: [10.1109/MWSYM.2014.6848450](https://doi.org/10.1109/MWSYM.2014.6848450). URL: <http://ieeexplore.ieee.org/document/6848450/>.
- [49] Gergory L. Charvat. *Small and Short-Range Radar Systems*. 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 2014. ISBN: 9781439865996.

Bibliography

- [50] *International Standard ISO/IEC/IEEE 8802-11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Amendment 3: Enhancements for very high throughput in the 60 GHz band.* 2014. DOI: [10.1109/IEEESTD.2014.6774849](https://doi.org/10.1109/IEEESTD.2014.6774849). URL: <http://ieeexplore.ieee.org/document/6774849/>.
- [51] Qiuchi Jian et al. “Detection of breathing and heartbeat by using a simple UWB radar system”. In: *The 8th European Conference on Antennas and Propagation (EuCAP 2014)*. EuCAP. IEEE, Apr. 2014, pp. 3078–3081. ISBN: 978-8-8907-0184-9. DOI: [10.1109/EuCAP.2014.6902477](https://doi.org/10.1109/EuCAP.2014.6902477). URL: <http://ieeexplore.ieee.org/document/6902477/>.
- [52] Kyle Kauffman and John Raquet. “Navigation via H-field signature map correlation and INS integration”. In: *2014 IEEE Radar Conference*. IEEE, May 2014, pp. 1390–1395. ISBN: 978-1-4799-2035-8. DOI: [10.1109/RADAR.2014.6875817](https://doi.org/10.1109/RADAR.2014.6875817). URL: <http://ieeexplore.ieee.org/document/6875817/>.
- [53] Mathieu Labbe and Francois Michaud. “Online global loop closure detection for large-scale multi-session graph-based SLAM”. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Sept. 2014, pp. 2661–2666. ISBN: 978-1-4799-6934-0. DOI: [10.1109/IROS.2014.6942926](https://doi.org/10.1109/IROS.2014.6942926). URL: <https://introlab.3it.usherbrooke.ca/mediawiki-introlab/images/e/eb/Labbe14-IROS.pdf%20http://ieeexplore.ieee.org/document/6942926/>.
- [54] Jonathan Lu et al. “Propagation characterization of an office building in the 60 GHz band”. In: *The 8th European Conference on Antennas and Propagation (EuCAP 2014)*. IEEE, Apr. 2014, pp. 809–813. ISBN: 978-8-8907-0184-9. DOI: [10.1109/EuCAP.2014.6901885](https://doi.org/10.1109/EuCAP.2014.6901885). URL: <http://ieeexplore.ieee.org/document/6901885/>.
- [55] Omnidaradar. *Interface Board 3.2 Layers*. 2014.
- [56] Honghao Tang. “DOA estimation based on MUSIC algorithm”. In: (2014). URL: <http://www.diva-portal.org/smash/record.jsf?pid=+diva2%3A724272&dswid=-1355>.
- [57] PTM van Zeijl, JW Wakker, and TG Saveliyev. *Radar Development Kit RIC60 Documentation*. 2014.
- [58] Martin Adams. *Robotic Navigation - Experience Gained with RADAR*. 2015. URL: <https://www.rit.edu/kgcoe/iros15workshop/papers/IROS2015-WASRoP-Invited-01-slides.pdf>.
- [59] Constantine A. Balanis. *Antenna Theory: Analysis and Design*. John Wiley & Sons, 2015, p. 1096. ISBN: 9781119178996. URL: <https://books.google.com/books?id=u-xbCwAAQBAJ>.
- [60] Martin Becker et al. “Approaches for Software Verification of An Emergency Recovery System for Micro Air Vehicles”. In: *34th International Conference on Computer Safety, Reliability and Security (SAFECOMP)*. Springer, Cham, Sept. 2015, p. 12. DOI: [10.1007/978-3-319-24249-1_32](https://doi.org/10.1007/978-3-319-24249-1_32). URL: http://link.springer.com/10.1007/978-3-319-24249-1_32%20http://arxiv.org/abs/1506.07867.

- [61] Hans Brouwer. *Microwave sensing with an One-Chip-Radar*. 2015. URL: <http://senseofcontact.nl/wp-content/uploads/sites/47/2015/04/Omniradar20150204Sense-of-Contact.pdf>.
- [62] Henrik Forstén. *Homemade synthetic aperture radar*. 2015. URL: <http://hforsten.com/homemade-synthetic-aperture-radar.html>.
- [63] Zhibiao Jiang et al. “Negative obstacle sensing based on real data of ultra-wideband SAR”. In: *IET International Radar Conference 2015*. Institution of Engineering and Technology, 2015. ISBN: 978-1-78561-038-7. DOI: [10.1049/cp.2015.1194](https://doi.org/10.1049/cp.2015.1194). URL: <http://digital-library.theiet.org/content/conferences/10.1049/cp.2015.1194>.
- [64] K.C. Lee. *Ultrasonic module Hacking*. 2015. URL: <https://hackaday.io/project/5903-sonar-for-the-visually-impaired/log/19976-ultrasonic-module-hacking>.
- [65] Mohamed Hussein Emam Mabrouk. “Signal Processing of UWB Radar Signals for Human Detection Behind Walls”. PhD thesis. University of Ottawa, 2015, p. 143.
- [66] Pavlo Molchanov et al. “Short-range FMCW monopulse radar for hand-gesture sensing”. In: *2015 IEEE Radar Conference (RadarCon)*. IEEE, May 2015, pp. 1491–1496. ISBN: 978-1-4799-8232-5. DOI: [10.1109/RADAR.2015.7131232](https://doi.org/10.1109/RADAR.2015.7131232). URL: <http://ieeexplore.ieee.org/document/7131232/>.
- [67] Stephan Mühlbacher-Karrer et al. *Short Range Capacitive Proximity Sensing*. 2015. URL: <https://www.rit.edu/kgcoe/iros15workshop/papers/IROS2015-WASRoP-Paper03.pdf>.
- [68] Takuya Sakamoto et al. “Range-Doppler surface: a tool to analyse human target in ultra-wideband radar”. In: *IET Radar, Sonar & Navigation* 9.9 (Dec. 2015), pp. 1240–1250. ISSN: 1751-8784. DOI: [10.1049/iet-rsn.2015.0065](https://doi.org/10.1049/iet-rsn.2015.0065). URL: <http://digital-library.theiet.org/content/journals/10.1049/iet-rsn.2015.0065>.
- [69] Hamed Sarbolandi, Damien Lefloch, and Andreas Kolb. “Kinect Range Sensing: Structured-Light versus Time-of-Flight Kinect”. In: *Computer Vision and Image Understanding (CVIU)* (May 2015). URL: <https://arxiv.org/pdf/1505.05459.pdf>.
- [70] Piyush Tewari, Sanjay Soni, and Bajrang Bansal. “Time-Domain Solution for Transmitted Field Through Low-loss Dielectric Obstacles in a Microcellular and Indoor Scenario for UWB Signals”. In: *IEEE Transactions on Vehicular Technology* 64.2 (Feb. 2015), pp. 541–552. DOI: [10.1109/TVT.2014.2323699](https://doi.org/10.1109/TVT.2014.2323699). URL: <http://ieeexplore.ieee.org/document/6815763/>.
- [71] Abbas Albaidhani, Antoni Morell, and Jose Lopez Vicario. “Ranging in UWB using commercial radio modules: Experimental validation and NLOS mitigation”. In: *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, Oct. 2016, pp. 1–7. ISBN: 978-1-5090-2425-4. DOI: [10.1109/IPIN.2016.7743639](https://doi.org/10.1109/IPIN.2016.7743639). URL: <http://ieeexplore.ieee.org/document/7743639/>.
- [72] Cesar Cadena et al. “Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age”. In: *IEEE Transactions on Robotics* 32.6 (Dec. 2016), pp. 1309–1332. ISSN: 1552-3098. DOI: [10.1109/TRO.2016.2624754](https://doi.org/10.1109/TRO.2016.2624754). URL: <http://ieeexplore.ieee.org/document/7747236/>.

Bibliography

- [73] Nikolaus Correll. *Introduction to Autonomous Robots*. 2nd. Magellan Scientific, 2016. ISBN: 9780692700877. URL: <https://github.com/correll/Introduction-to-Autonomous-Robots>.
- [74] Robert Ernst. “Development of a Compact, Highly Integrated 60 GHz FMCW Radar for Human Vital Sign Monitoring”. PhD thesis. Halmstad University, School of Information Technology, 2016, p. 56. URL: <http://hh.diva-portal.org/smash/record.jsf?pid=diva2%3A945057>.
- [75] Robert Ernst, Emil Nilsson, and Per-Arne Viberg. “60GHz vital sign radar using 3D-printed lens”. In: *2016 IEEE SENSORS*. IEEE, Oct. 2016, pp. 1–3. ISBN: 978-1-4799-8287-5. DOI: [10.1109/ICSENS.2016.7808774](https://doi.org/10.1109/ICSENS.2016.7808774). URL: <http://ieeexplore.ieee.org/document/7808774/>.
- [76] Anna Guerra et al. “Crowd-based personal radars for indoor mapping using UWB measurements”. In: *2016 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB)*. IEEE, Oct. 2016, pp. 1–4. ISBN: 978-1-5090-1317-3. DOI: [10.1109/ICUWB.2016.7790437](https://doi.org/10.1109/ICUWB.2016.7790437). URL: <http://ieeexplore.ieee.org/document/7790437/>.
- [77] Wolfgang Hess et al. “Real-Time Loop Closure in 2D LIDAR SLAM”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 1271–1278. DOI: [10.1109/ICRA.2016.7487258](https://doi.org/10.1109/ICRA.2016.7487258). URL: <https://research.google.com/pubs/pub45466.html>.
- [78] Youngwook Kim and Brian Toomajian. “Hand Gesture Recognition Using Micro-Doppler Signatures With Convolutional Neural Network”. In: *IEEE Access* 4 (2016), pp. 7125–7130. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2016.2617282](https://doi.org/10.1109/ACCESS.2016.2617282). URL: <http://ieeexplore.ieee.org/document/7590035/>.
- [79] Jaime Lien et al. “Soli: Ubiquitous Gesture Sensing with Millimeter Wave Radar”. In: *ACM Transactions on Graphics* 35.4 (July 2016), pp. 1–19. ISSN: 07300301. DOI: [10.1145/2897824.2925953](https://doi.org/10.1145/2897824.2925953). URL: [%20http://dl.acm.org/citation.cfm?doid=2897824.2925953](http://doi.acm.org/10.1145/2897824.2925953).
- [80] Karl Nordin and Sina Shamekhi. “24 GHz Patch Antenna Array Design for RADAR”. PhD thesis. Faculty of Engineering, LTH, Lund University, 2016, p. 102.
- [81] Matthias Rapp et al. “FSCD and BASD: Robust landmark detection and description on radar-based grids”. In: *2016 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*. IEEE, May 2016, pp. 1–4. ISBN: 978-1-5090-2367-7. DOI: [10.1109/ICMIM.2016.7533831](https://doi.org/10.1109/ICMIM.2016.7533831). URL: <http://ieeexplore.ieee.org/document/7533831/>.
- [82] Ralf Reuter. *Radar Innovations by NXP*. 2016. URL: <https://community.nxp.com/docs/DOC-331463>.
- [83] Branko Ristic et al. “A Random Finite Set Approach to Occupancy-Grid SLAM”. In: (2016), pp. 1–15.

- [84] F. Schuster et al. “Landmark based radar SLAM using graph optimization”. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Nov. 2016, pp. 2559–2564. ISBN: 978-1-5090-1889-5. DOI: [10.1109/ITSC.2016.7795967](https://doi.org/10.1109/ITSC.2016.7795967). URL: <http://ieeexplore.ieee.org/document/7795967/>.
- [85] Clemens Valens. “Review : Walabot – or how to transform your smartphone into a radar”. In: *Elektor* (June 2016). URL: <https://www.elektormagazine.com/news/review-walabot-or-how-to-transform-your-smartphone-into-a-radar>.
- [86] Filippo Valmori et al. “Indoor detection and tracking of human targets with UWB radar sensor networks”. In: *2016 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB)*. IEEE, Oct. 2016, pp. 1–4. ISBN: 978-1-5090-1317-3. DOI: [10.1109/ICUWB.2016.7790435](https://doi.org/10.1109/ICUWB.2016.7790435). URL: <http://ieeexplore.ieee.org/document/7790435/>.
- [87] Wipassorn Vinicchayakul, Sathaporn Promwong, and Pichaya Supanakoon. “Study of UWB indoor localization using fingerprinting technique with different number of antennas”. In: *2016 International Computer Science and Engineering Conference (ICSEC)*. IEEE, Dec. 2016, pp. 1–4. ISBN: 978-1-5090-4420-7. DOI: [10.1109/ICSEC.2016.7859934](https://doi.org/10.1109/ICSEC.2016.7859934). URL: <http://ieeexplore.ieee.org/document/7859934/>.
- [88] Claire M. Watts et al. “2D and 3D millimeter-wave synthetic aperture radar imaging on a PR2 platform”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2016, pp. 4304–4310. ISBN: 978-1-5090-3762-9. DOI: [10.1109/IROS.2016.7759633](https://doi.org/10.1109/IROS.2016.7759633). URL: <http://ieeexplore.ieee.org/document/7759633/>.
- [89] Yi Zhong et al. “Classification of animals and people based on radio-sensor network”. In: *2016 16th International Symposium on Communications and Information Technologies (ISCIT)*. IEEE, Sept. 2016, pp. 113–116. ISBN: 978-1-5090-4099-5. DOI: [10.1109/ISCIT.2016.7751603](https://doi.org/10.1109/ISCIT.2016.7751603). URL: <http://ieeexplore.ieee.org/document/7751603/>.
- [90] Guopeng Zhu, Ting Jiang, and Zheng Zhou. “A fusion method for object identification in rainy weather based on PSO-SFLA and UWB”. In: *2016 16th International Symposium on Communications and Information Technologies (ISCIT)*. IEEE, Sept. 2016, pp. 537–541. ISBN: 978-1-5090-4099-5. DOI: [10.1109/ISCIT.2016.7751689](https://doi.org/10.1109/ISCIT.2016.7751689). URL: <http://ieeexplore.ieee.org/document/7751689/>.
- [91] Seunghoon Cho et al. “A new direction-of-arrival estimation method using automotive radar sensor arrays”. In: *International Journal of Distributed Sensor Networks* 13.6 (June 2017), p. 155014771771362. ISSN: 1550-1477. DOI: [10.1177/1550147717713628](https://doi.org/10.1177/1550147717713628). URL: <http://journals.sagepub.com/doi/10.1177/1550147717713628>.
- [92] FEDERAL COMMUNICATIONS COMMISSION and OFFICE OF ENGINEERING AND TECHNOLOGY POLICY AND RULES DIVISION. *FCC ONLINE TABLE OF FREQUENCY ALLOCATIONS*. 2017. URL: <https://transition.fcc.gov/oet/spectrum/table/fccitable.pdf>.

Bibliography

- [93] Robin P. Guan et al. “Feature-based robot navigation using a Doppler-azimuth radar”. In: *International Journal of Control* 90.4 (Apr. 2017), pp. 888–900. ISSN: 0020-7179. DOI: [10.1080/00207179.2016.1244727](https://doi.org/10.1080/00207179.2016.1244727). URL: <https://www.tandfonline.com/doi/full/10.1080/00207179.2016.1244727>.
- [94] Fezza Haider and George Shaker. *People and Fall Detection with Walabot*. 2017. URL: <https://www.hackster.io/0/-8db4aa>.
- [95] Daniel Novak, Dusan Kocur, and Jakub Demcak. “Static person detection and localization with estimation of person’s breathing rate using single multistatic UWB radar”. In: *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, Jan. 2017, pp. 000143–000148. ISBN: 978-1-5090-5655-2. DOI: [10.1109/SAMI.2017.7880291](https://doi.org/10.1109/SAMI.2017.7880291). URL: <http://ieeexplore.ieee.org/document/7880291/>.