



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.



# **Acknowledgements**

Vielen Dank ...

München, im Monat Jahr



# **Contents**



# 1 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.

## 1.1 Hardware Platform

### 1.1.1 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 ??) was developed by Yujin Robot<sup>1)</sup>, 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.



(a) Yujin's iClebo Kobuki robot platform.  
Source: [DesignK2013]



(b) Hardkernel Odroid XU4. Source:  
[https://wikimedia.org/wiki/File:Hardkernel\\_Odroid\\_XU4\\_Board.jpg](https://wikimedia.org/wiki/File:Hardkernel_Odroid_XU4_Board.jpg)

---

<sup>1)</sup> <http://kobuki.yujinrobot.com/>

### 1.1.2 ROS integration

The Robot Operating System (ROS)<sup>2)</sup> 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 package<sup>3)</sup> is widely used and tested. With `rosbags`, the ROS system also provides a convenient way to record data for later playback and analysis.

### 1.1.3 Odroid

In the Kobuki platform, only an embeddded 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 XU4<sup>4)</sup> (see ??) single-board computer provides plenty of power with its Samsung Exynos Cortex-A15/Cortex-A7 2GHz Octa core CPUs paired with 2GB ram. It runs Ubuntu Linux 14.04 with ROS Indigo Igloo.

### 1.1.4 Lidar Slam

A RoboPeak / Slamtec RPLidar A1<sup>5)</sup> 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 [Hess2016] with the Cartopgrapher ROS Integration<sup>6)</sup>. Cartographer ROS for TurtleBots<sup>7)</sup> conveniently provides configuration files that work well with the Kobuki.

### 1.1.5 Radar sensor

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

#### 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

---

<sup>2)</sup> <http://www.ros.org/>

<sup>3)</sup> <http://wiki.ros.org/kobuki>

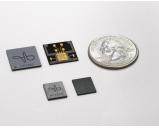
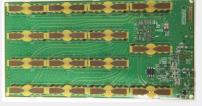
<sup>4)</sup> [http://www.hardkernel.com/main/products/prdt\\_info.php?g\\_code=G143452239825](http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825)

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

<sup>6)</sup> [https://github.com/googlecartographer/cartographer\\_ros](https://github.com/googlecartographer/cartographer_ros)

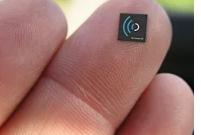
<sup>7)</sup> [https://github.com/googlecartographer/cartographer\\_turtlebot](https://github.com/googlecartographer/cartographer_turtlebot)

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  $\Delta f$  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 <sup>8)</sup>	High bandwidth. Presentation at SoC 2015[Brouwer2015]	60 GHz, 7 GHz	On-chip, 1 Tx, 2 Rx	\$4000	
Google / Infineon Soli <sup>9)</sup>	Expected 2018. Sub-millimeter accuracy, running at over 10,000 frames per second [Lien2016]	60 GHz, 7 GHz	In-package, ? 2 Tx, 4 Rx		
Walabot Pro <sup>10)</sup>	3D configuration. Slow update rate	6.8 GHz, 7 GHz	On-board, 9 Tx, 9 Rx	\$599	
Bosch Prototype	Prototype for In-wall pipe detection	5.15 GHz, 6.7 GHz	External, 2 Tx/Rx	-	
Silicon Radar SiRad Simple <sup>11)</sup>	Has WiFi	122 GHz, 6.4 GHz	On-chip, 1 Tx, 1 Rx	?	
Anokiwave AWMF-0117 <sup>12)</sup>		12.5 GHz, 4.5 GHz	On-chip, 1 Tx/Rx	?	
NXP Cocoon Radar <sup>13)</sup>	Relatively small board. Presentation at FTF 2016[Reuter2016]	77 GHz, 4 GHz	On-board, 3 Tx, 4 Rx	?	
TimeDomain P440 <sup>14)</sup>	Can operate as multistatic radar or UWB communication node	4 GHz, 1.7 GHz	External, 2 Tx/Rx	\$5000	

Continued on next page

## 1 Implementation

Product	Note	$f_C$ , $\Delta f$	Antennas	DK Price	Picture
Novelda Xethru X4M03 <sup>15)</sup>		8 GHz, 1.5 GHz	On-board, 1 Tx/Rx	\$399	
RFbeam MR2001_RD <sup>16)</sup>		77 GHz, 1 GHz	On-board, 4 Tx, 6 Rx	?	
Inras 77Ghz Radarbook <sup>17)</sup>	Configurable FPGA processing chain	77 GHz, 1 GHz	On-board, 4 Tx, 8 Rx	\$7300	
Acconeer A1 <sup>18)</sup>	Sub-mm accuracy	60 GHz, ?	On-chip, ?	?	
Inras 24Ghz Radarbook <sup>19)</sup>		24 GHz, 250 MHz	On-board, 4 Tx, 4 Rx	\$7300	
Infineon BGT24-RFB2412-EVAL <sup>20)</sup>		24 GHz, 250 MHz	On-board, 1 Tx, 2 Rx	\$1333	
IMST DK-sR-1200e <sup>21)</sup>		24 GHz, 250 MHz	On-board, 1 Tx, 2 Rx	\$3333	
InnoSenT IVS-565 <sup>22)</sup>		24 GHz, 250 MHz	On-board, 1 Tx, 2 Rx	?	

Continued on next page

Product	Note	$f_C, \Delta f$	Antennas	DK Price	Picture
ST EVB- STradA431 <sup>23)</sup>	SMA connectors for internal signals	24 GHz, 250 MHz	External, 1 Tx, 3 Rx	?	
OmniPreSense OPS241-A <sup>24)</sup>	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.

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

<sup>8)</sup> <https://www.omnidaradar.com/products/>

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

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

<sup>11)</sup> [http://www.siliconradar.de/evalkits\\_e.html](http://www.siliconradar.de/evalkits_e.html)

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

<sup>13)</sup> Reuter2016

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

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

<sup>16)</sup> [https://www.rfbeam.ch/files/products/26/downloads/ProductBrief\\_MR2001\\_RD.pdf](https://www.rfbeam.ch/files/products/26/downloads/ProductBrief_MR2001_RD.pdf)

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

<sup>18)</sup> <http://www.acconeer.com/>

<sup>19)</sup> dkradarbook

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

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

<sup>22)</sup> [http://www.innosent.de/fileadmin/media/dokumente/DATASHEETS\\_2016/Datenblatt\\_IVS-565.pdf](http://www.innosent.de/fileadmin/media/dokumente/DATASHEETS_2016/Datenblatt_IVS-565.pdf)

<sup>23)</sup> [http://www.st.com/content/st\\_com/en/products/evaluation-tools/product-evaluation-tools/automotive-ic-eval-boards/evb-strada431.html](http://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/automotive-ic-eval-boards/evb-strada431.html)

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

## 1 Implementation

### Bosch Radar

The Bosch radar presented some challenges under a Linux environment. 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 under Windows, but by the time the board arrived, the decision to focus on Omnidaradar's chip was made and the newer driver was not cross-platform patched.

### Vayyar Walabot

Vayyar<sup>25)</sup> is an Israeli startup<sup>26)</sup> 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 documentation<sup>27)</sup> they showcase the available sensor operation modes: 3D imaging, 2D imaging, object tracking, pipe detection and raw data (see ??).

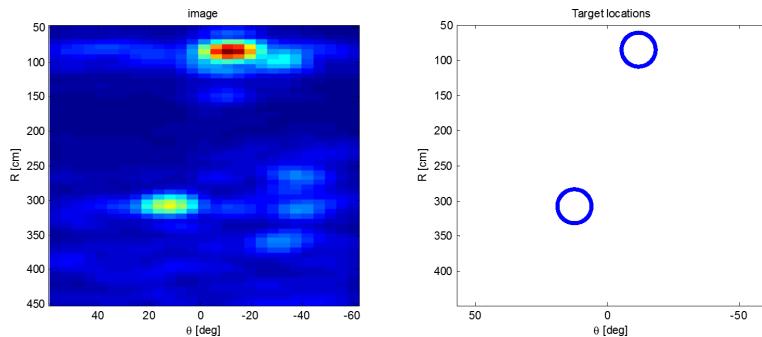


Figure 1.2: Vayyar's Walabot Pro sensor is claimed to have Target localization and tracking. Source: [https://api.walabot.com/\\_features.html#\\_examples](https://api.walabot.com/_features.html#_examples)

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 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.

<sup>25)</sup> <https://www.vayyar.com/>

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

<sup>27)</sup> <https://api.walabot.com>

At the time of writing, the only interesting published project is a fall detection scenario [**Haider2017**] by Haider, in which people can be localized at intersections of vertically and horizontally oriented heatmaps.

**Static range test** ?? shows the signal from two Walabot antenna pairs as it records the scene in image ?? with metal can stacks at 0.5m 1m and 1.5m. 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 ?? 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 7GHz. 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 ??) 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 ??.

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 ??.

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 [**Valens2016**] 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 that the number of objects to be detected must be configured as a fixed parameter first.

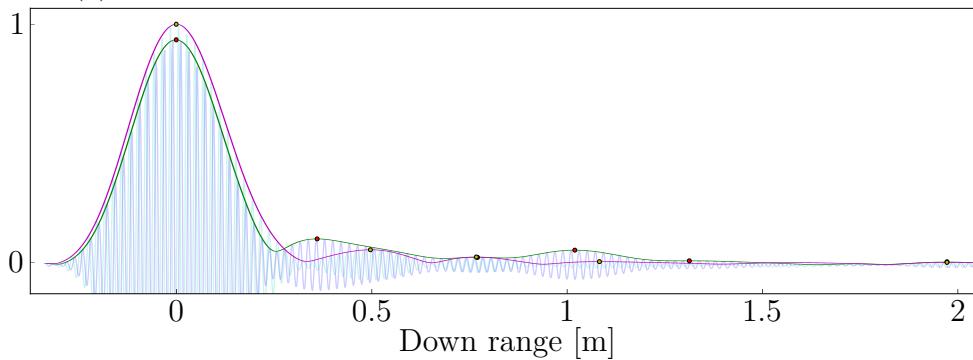
## Omniradar RIC60-A

With its good range resolution and a good idea of its capabilities from [**Ernst2016**], this sensor promised good results. It will provide the basis for the proof of concept implemen-

## 1 Implementation



(a) Setup with three towers of cans in front of the Walabot sensor

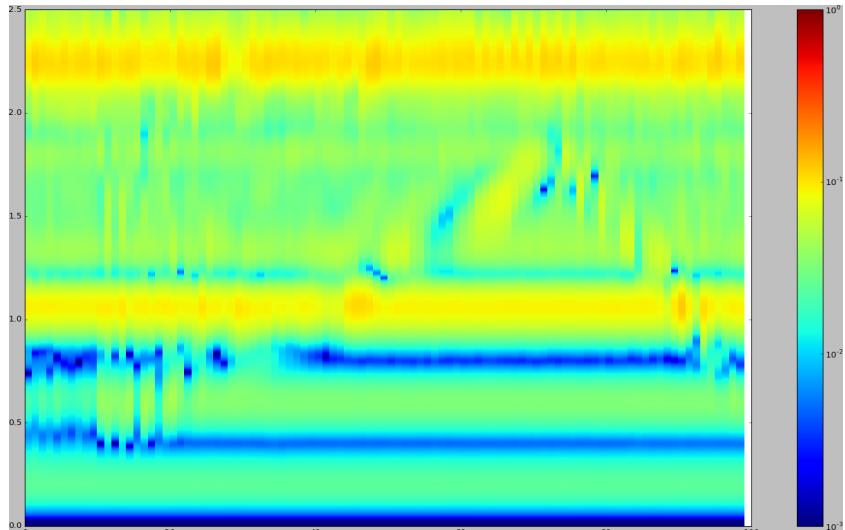


(b) Normalized echo intensity of range profile measurement with two antenna pairs. After the transmit peak at  $t = 0$ , three peaks corresponding to the three can targets are vaguely perceptible.

Figure 1.3: 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 1.4: Dynamic range test with the Walabot sensor

tation, so it receives a more detailed look.

Founded in 2010, Omnidaradar<sup>28)</sup> is a Dutch startup<sup>29)</sup> 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 7GHz of bandwidth on two receiving antennas. An Altera Cyclone IV FPGA handles the signal acquisition and communication. ?? shows the radar IC and how the three antennas are integrated in silicon.

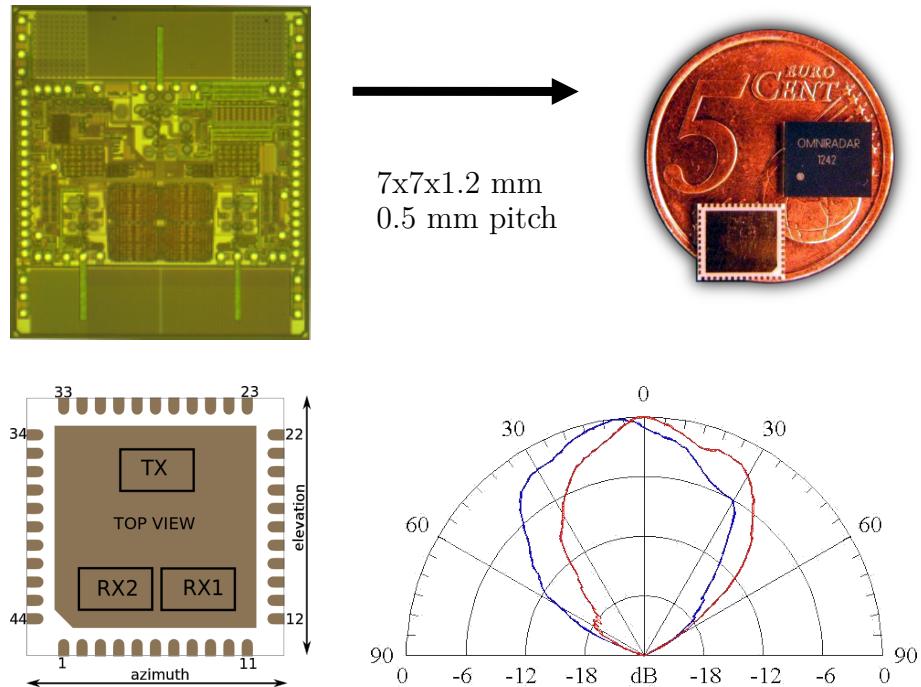


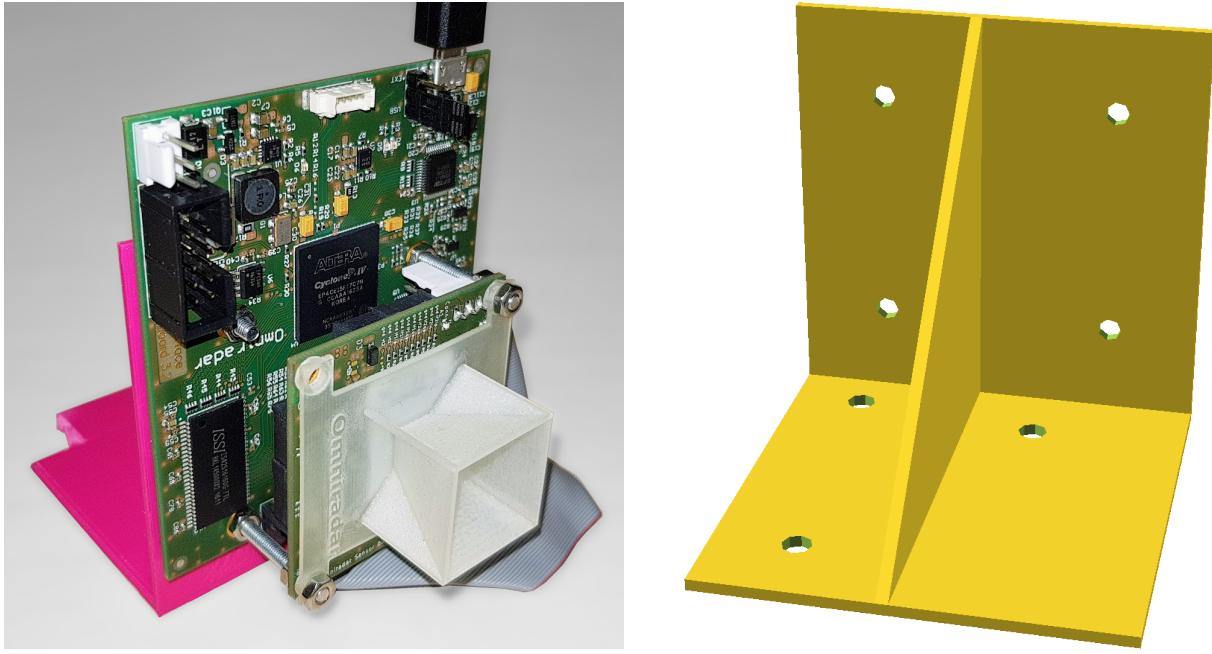
Figure 1.5: Decapped Omnidaradar IC (top left); five cent coin as size reference (top right); schematic with antenna locations (botto left); and antenna directivity pattern of Omnidaradar's RIC60-A (bottom right). Adapted from [Brouwer2015] p.9

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 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.

<sup>28)</sup> <https://www.omnidaradar.com/>

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

## 1 Implementation



(a) Omnidar's RIC60-A sensor with horn antenna extension, attached to the 3D-printed Kobuki mount

(b) 3D model of the radar sensor mount, designed with OpenSCAD.

Figure 1.6: 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 OpenSCAD<sup>30)</sup> and printed on a Dremel 3D printer<sup>31)</sup>. The bottom mount hole positions were extracted from the mechanical drawings of the Kobuki Base [YujinRobot2012]; the side holes from the Altium layer document of the version 3.2 Omnidar interface board [Omnidar2014].

When rotated to face to the left side of the robot, the RPLidar mount was in the way, so parts of the print had to be clipped off.

Two screws hold the Omnidar interface board to the mount. The other two mounting holes in the interface board hold the Omnidar 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 degrees, thanks to the symmetric layout.

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

The RIC60A has a sensitivity of  $400 \frac{Hz}{m/s}$ . 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.

<sup>30)</sup> <http://www.openscad.org/>

<sup>31)</sup> <https://3dprinter.dremel.com/>

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).

Sampling frequency  $F_s = 25MHz$  and RIC60A Doppler sensitivity,  $S_D = 400 \frac{Hz}{m/s}$  are constant values of the Omnidaradar RDK.

Given a chirp duration of  $T_{chirp} = 2.5ms$ , we get  $N_s = t_{chirp}F_s = 62500$  Samples,  $N_r = \lfloor \frac{N_s}{2} \rfloor = 31250$  Samples per up/downsweep,  $dF = \frac{F_s}{N_r-1} = 800Hz$  FFT frequency bin width and hence a Doppler resolution of  $\frac{dF}{S_D} = 2m/s$ .

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.

TODO wording?

### 1.1.6 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.

**Very short durations** ( $< 2ms$ ) incur a considerable processing overhead to acquisition time ratio and have a very low SNR.

**Short durations** ( $< 5ms$ ) have acceptable SNR, and are more efficient with respect to overhead.

**Long durations** ( $> 15ms$ ) have good SNR, and don't create a lot of overhead. However at higher robot speeds, target peaks get blurred over several range bins as they move in range. Less intense target echos are more difficult to detect then.

**Very long durations** ( $> 20ms$ ) required the Omnidaradar driver to be patched on Linux so as not to freeze when chirps longer than  $20ms$  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.

The optimal range was empirically found to lie between  $2ms$  and  $10ms$ .

TODO wording here

?? shows the chirp efficiency  $\eta = \frac{n_{chirp} t_{chirp}}{t_{msg}}$ , with number of consecutive sweeps  $n_{chirp}$  (two in the graph's data source), chirp length  $t_{chirp}$ , and  $t_{msg}$  the time since last radar message.

The chirp length has an effect on accuracy and resolution. ?? shows the range profiles of one scene, recorded over  $10s$ . Short chirp times lead to a higher noise floor, but exhibit less variance, i.e. a slimmer band of frequencies. Longer measurement times have a lower

## 1 Implementation

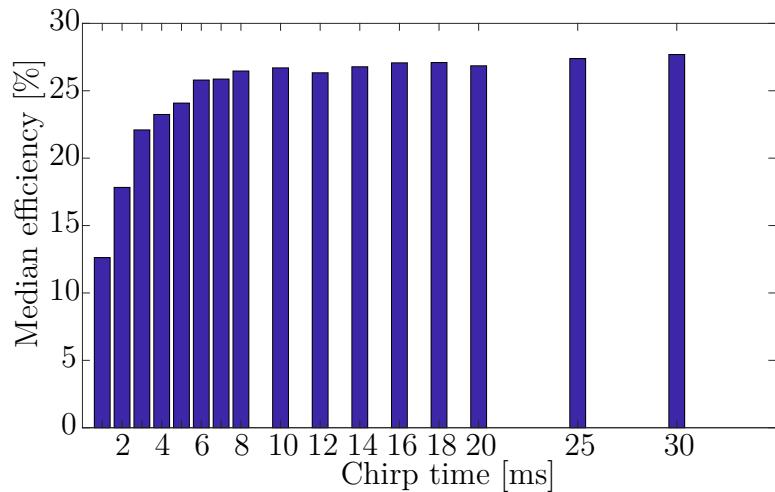


Figure 1.7: Chirp efficiency  $\eta$  (measurement time per time spent) for various chirp lengths

noise floor but have more outliers. Especially with the horn extension attached longer measurement time also improves SNR.

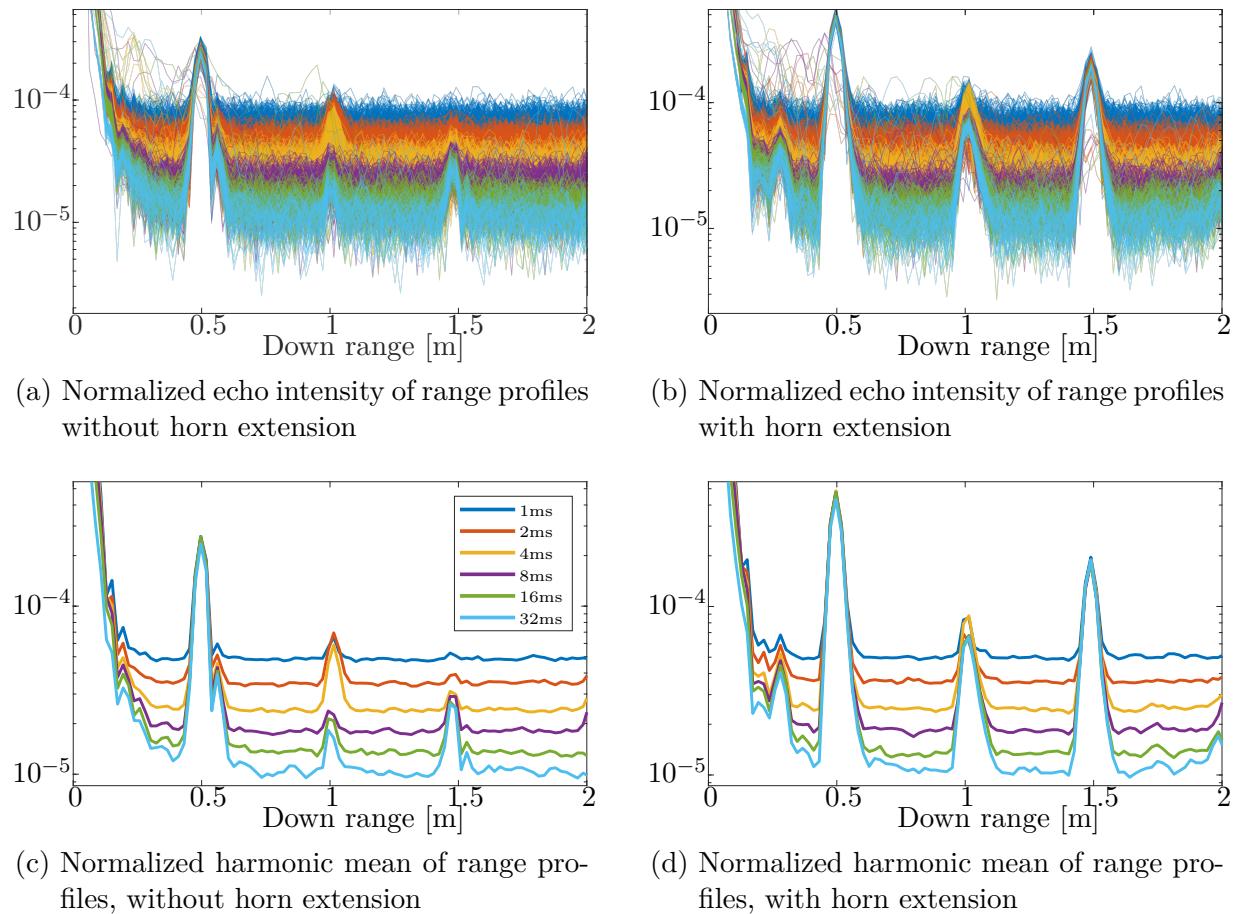


Figure 1.8: Effect of chirp length on SNR; with and without Horn extension



# A List of Scans

Table A.1: Scan parameters

Name	Sweep	Orient.	Squint	Horn	Map	TF
Attic	5ms	H	90°	✓	X	X
Basement	5ms	H	90°	✓	X	X
Cafeteria	5ms	H	90°	✓	X	X
Dungeon	5ms	V	20°	✓	X	X
Entryway	5ms	V	20°	✓	X	X
Fallout shelter	5ms	V	20°	✓	X	X
Garden	5ms	V	20°	✓	X	X
Home cinema	5ms	V	20°	✓	X	X
Indoor swimming pool	5ms	V	20°	✓	X	X
Jail cell	5ms	H	20°	✓	X	X
Kitchen	5ms	H	20°	X	X	X
Lobby	5ms	H/I	0°	X	X	X
Mancave	5ms	H/I	0°	X	X	X
Nirvana	5ms	H/I	0°	X	✓	X
Orbit	5ms	H/I	0°	X	✓	X
Public Restroom	5ms	H/I	0°	X	✓	✓
Queue	20ms	H/I	0°	X	✓	✓
Racetrack	2ms	H/I	0°	X	✓	✓
Sauna	2.5ms	H/I	0°	X	✓	✓
Torture Chamber	2.5ms	H/I	45°	X	✓	✓
Underground	2.5ms	H/I	45°	X	✓	✓
Virtual Reality	2.5ms	H/I	45°	X	✓	✓
Washroom	2.5ms	V	45°	X	✓	✓
Xray-Room	2.5ms	V	0°	X	✓	✓
Y is there a cable on the floor	5ms	V	0°	X	✓	✓
Z						

TODO

## A List of Scans

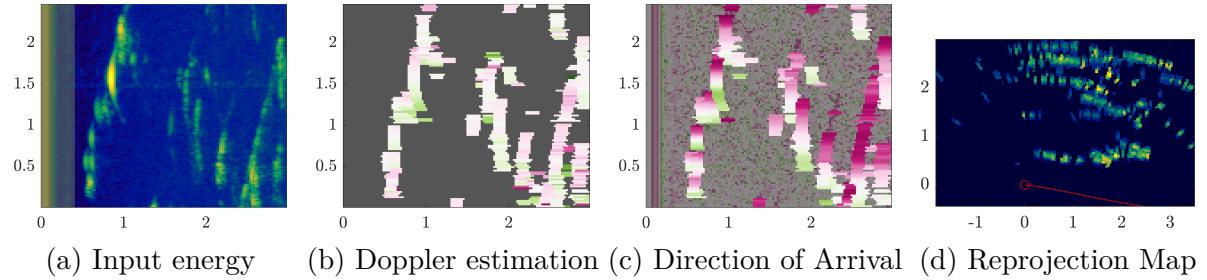


Figure A.1: Attic scan

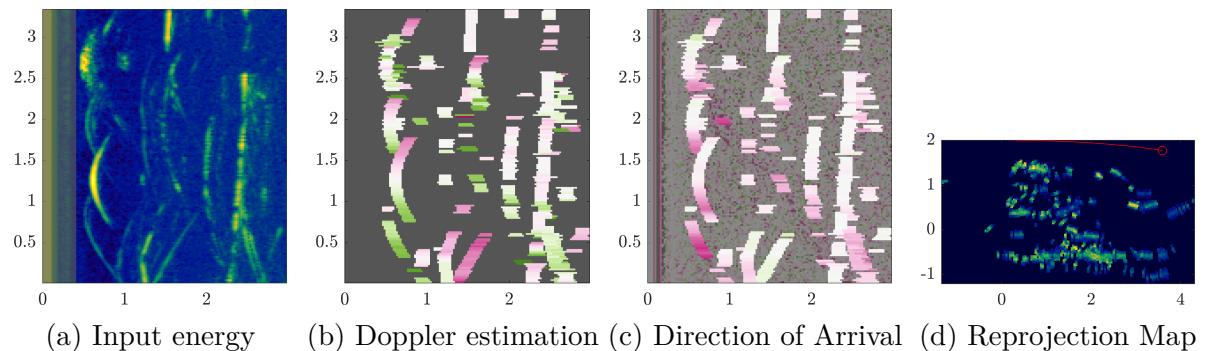


Figure A.2: Basement scan

# **List of Figures**



## **List of Tables**