

---

OTTO-VON-GUERICKE-UNIVERSITY MAGDEBURG

FACULTY OF COMPUTERSCIENCE  
Institute for Intelligent Cooperative Systems



---

## Master's Thesis

# Autonomous Driving in Urban Centers - Roundabout Monitoring

Julian-B. Scholle

March 21, 2017

Otto-von-Guericke-Universität Magdeburg  
Fakultät für Informatik  
Universitätsplatz 2  
39106 Magdeburg  
Germany

---

---

# Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne unerlaubte fremde Hilfe angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Stellen sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keinem anderen Prüfungsaamt vorgelegt und auch nicht veröffentlicht.

Göteborg, den March 21, 2017

---

Julian-B. Scholle

---

## Acknowledgements

I would especially like to thank Associate professor Christian Berger and J. Prof Sebastian Zug for their organizational and technical support during and especially in the run-up to this work and The “Deutschen Akademischen Austauschdienst” - DAAD for their financial support during my stay in Gothenburg. I would also like to thank all the other colleagues from Gothenburg, who have supported me professionally and morally.

---

# Index of Abbreviations

**DBSCAN** Density-Based Spatial Clustering of Applications with Noise

**SVD** Singular Value Decomposition

**LLSQ** Linear least Squares

**RANSAC** Random Sample Consensus

**MKS** multi-body simulation

**ACC** Adaptive Cruise Control

**DARPA** Defense Advanced Research Projects Agency

**CTRV** Constant Turn Rate and Velocity

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Ausgangssituation . . . . .	1
1.1.1	Test Platform . . . . .	1
1.2	Zielsetzung . . . . .	3
<b>2</b>	<b>Basic Knowledge</b>	<b>4</b>
2.1	Roundabouts . . . . .	4
2.1.1	Roundabouts in Law . . . . .	5
2.1.2	Elements of a Roundabout . . . . .	5
2.1.3	Types of Roundabouts . . . . .	6
2.2	Random Sample Consensus . . . . .	8
2.3	Density-Based Spatial Clustering of Applications with Noise .	9
2.4	Middleware OpenDAVINCI . . . . .	9
<b>3</b>	<b>State of the Art</b>	<b>11</b>
<b>4</b>	<b>Methodology</b>	<b>13</b>
<b>5</b>	<b>Sensor Analysis</b>	<b>15</b>
5.1	Theoretical Analysis . . . . .	15
5.2	Practical Analysis . . . . .	16
<b>6</b>	<b>Objekt Detection</b>	<b>18</b>
6.1	Ground Removal . . . . .	18
6.1.1	Planefitting . . . . .	20
6.2	Clustering . . . . .	22
6.3	Tracking . . . . .	23

6.3.1	Cluster Tracking . . . . .	23
6.3.2	Object Tracking . . . . .	24
6.3.3	Object Confidence . . . . .	27
6.4	Classification . . . . .	27
6.5	State Estimation . . . . .	28
6.5.1	Constant Turn Rate and Velocity Model . . . . .	28
6.5.2	Prediction . . . . .	29
<b>7</b>	<b>Simuation</b>	<b>30</b>
7.1	Simuation Scenario . . . . .	30
7.2	Simulation Logic . . . . .	31
7.2.1	Sensor Connection . . . . .	31
7.2.2	Mapping . . . . .	32
7.2.3	Statemachine . . . . .	34
<b>8</b>	<b>Evaluation</b>	<b>36</b>
8.1	Simuation . . . . .	36
8.2	Real Measurements . . . . .	36
<b>9</b>	<b>Conclusions</b>	<b>37</b>
<b>10</b>	<b>Future Work</b>	<b>38</b>

# 1

# Introduction

bezug auf kreisverkehre fehlt

Autonomous driving and the networking of vehicles with their environment are, together with electromobility, the most frequently discussed topics in the automotive sector. Rightly so: Autonomous driving has the potential to create completely new structures in the mobility market.<sup>1</sup>

So also, the Chalmers University of Technology, which has also initiated the project "CampusShuttle" in addition to Volvos' "DriveMe" project, is an interdisciplinary and cooperative research project at the Chalmers University of Technology and the University of Gothenburg. The project is located in the ReVeRe (Chalmers Research Vehicle Resource). The vision is a self-driving car between the two campuses of Chalmers.

Within the scope of the project, the vehicle is to be examined in various traffic scenarios. The focus is on urban transport and the vehicle must not only be able to interact with other cars, but also be safe with trams, buses, bicycles and all other traffic users.

---

## 1.1 Ausgangssituation

### 1.1.1 Test Platform

The test platform used in this work is a Volvo XC90 (2015) SUV, named Snowfox (see fig. 1.1). This test platform is equipped with many sensors for environmental monitoring. This includes five radar sensors, all around the vehicle, where front radar has a wider range. As well as a stereo camera and a Velodyne VLP-16 LiDAR. The arrangement of the sensors can be taken from fig. 1.2.

A Applanix POS LV is installed in the vehicle in addition to the environmental sensor system and the standard vehicle sensors (for example odometer, inertial sensors). At the time of writing this work, it was unfortunately not yet possible to access the radar sensors and the stereo camera. Therefore, only the Velodyne Lidar and the Applanix system will be described in detail.

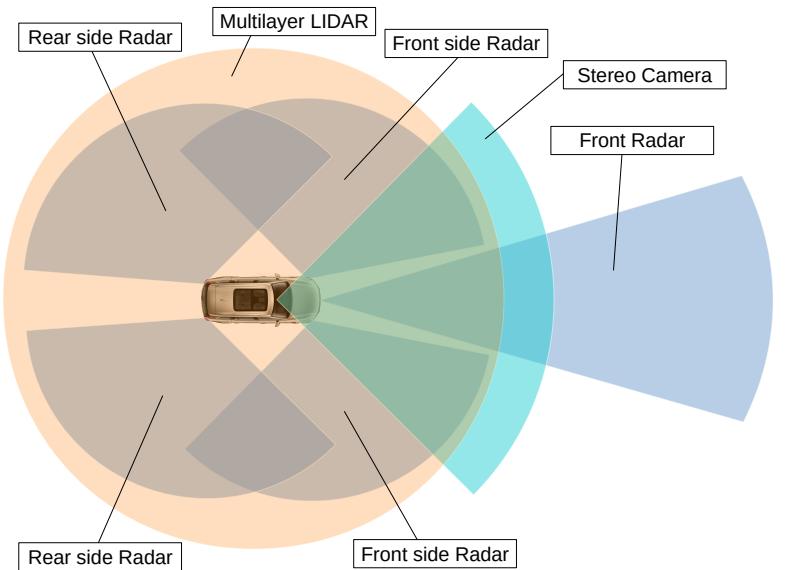
---

1. <https://www2.deloitte.com/de/de/pages/consumer-industrial-products/articles/autonomes-fahren-in-deutschland.html>  
(03/09/2017)

Figure 1.1: Test Platform  
Snowfox



Figure 1.2: Snowfox Sensors



### Velodyne VLP-16 LiDAR

The Velodyne VLP-16 is a 360 degree 3D laser scanner with a rotational speed of 5 to 20 revolutions per second cite manVEL. It provides a vertical FOV of 30 degrees, at 2 degrees resolution. With a range of 100m it can cover a circumference of 200m diameter. Furthermore, the VLP-16 can be synchronized with the Applanix POS LV, which allows a low-jitter signal. A further function of the Velodyne sensor is that it can react to different measuring pulses. By evaluating the last pulse instead of the strongest pulse it is possible to see through transparent objects. This allows us to determine the width of the vehicle in a later part of this work, as the Velodyne can look through the glass windows of the vehicle. At a set speed of 10Hz, the VLP-16 provides a resolution of 0.2 degrees with a variance of +- 3cm. The VLP-16 is centered on the roof of the XC90 in order to achieve the highest possible positioning to achieve a panoramic view of the vehicle. It is important to note that this alignment is unacceptable to the sensor because the sensor has a vertical field of

view of -15 to +15 degrees. As a result, all measurements over zero degrees are practically useless. The view of the manufacturer side<sup>2</sup> Reveals that the VLP-16 has been constituted for use with drones, while the larger HDL64E<sup>3</sup> is advertised explicitly for the urban automotive sector, and has a field of view of +2 to -24.9 degrees and thus for use in the Automotive sector appears to be more appropriate. The resulting problems will be discussed later.

### **Applanix POS LV**

The POS LV is a compact position and orientation system. It offers stable, reliable and reproducible positioning solutions for land-based vehicle applications. The POS LV provides an inertial sensor and odometry based position measurement with an accuracy of up to 0.3m (up to 0.035m when using the RTK correction). Furthermore, the heading delivered by the POS LV is also used, which provides an accuracy of 0.2 degrees. Even after losing the GPS signal, the POS-LV can provide a position through its odometer and the inertial sensor. However, this will deteriorate over time so that an accuracy of 2.51m can be expected 60 seconds after the GPS signal is lost. [8]

---

## **1.2 Zielsetzung**

Since autonomous driving is a very wide, interdisciplinary subject, it is obvious. That not everything in this work can be dealt with. Within the scope of darpaH Challenge many papers were published on this subject. What has not yet been explicitly discussed in these publications is the handling of roundabouts, with autonomous vehicles. The aim of this thesis is therefore to analyze what sensor equipment is necessary for the observation of roundabouts, or whether the existing sensor equipment of the ReVeRe test vehicle Snowfox can be regarded as sufficient.

---

2. <http://velodynelidar.com/vlp-16.html> (03/09/2017)  
3. <http://velodynelidar.com/hdl-64e.html> (03/09/2017)

# 2

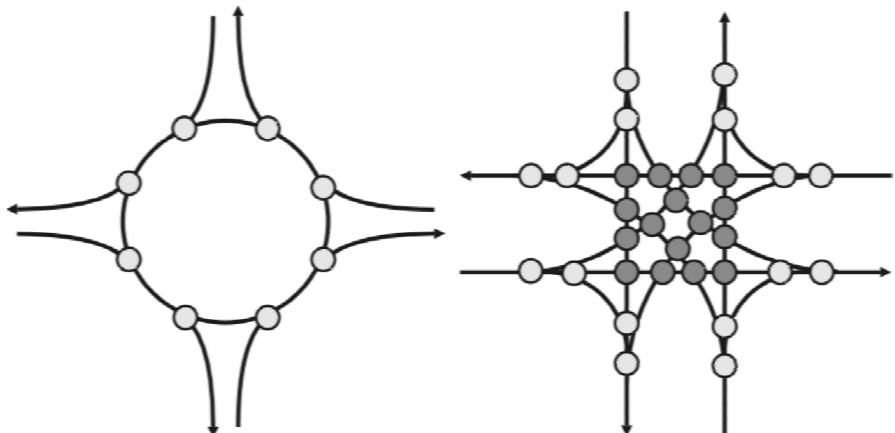
## Basic Knowledge

### 2.1 Roundabouts

reference

Roundabouts are growing in popularity in Germany. As seen in fig. 2.1, they have a smaller number of points of conflict (8) as opposed to crossroads (32), thus contributing greatly to road safety, which reduces the rate of accidents.

Figure 2.1: Roundabout Conflict Points [23]

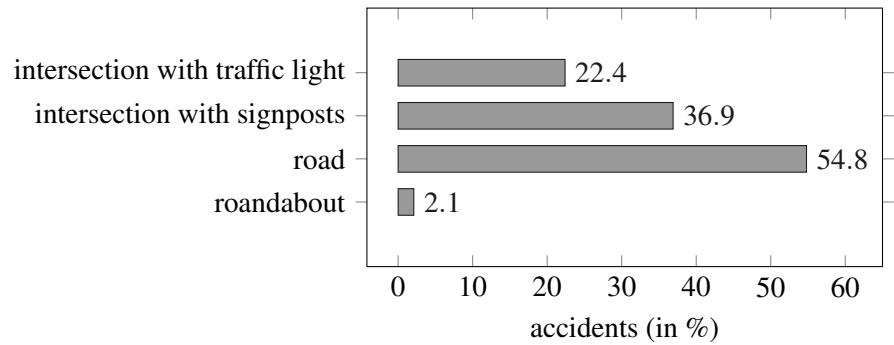


439/5000 For example, in 2009 a survey of 54426 accidents in the inner city area was made. The number of accidents in roundabouts was 1150 (2.1 %) [3]. You can see the statistics in fig. 2.2. Furthermore, circular traffic also allows nodes to be constituted with more than 4 roads, which gives them a further advantage. Unfortunately, there are no publications known to the author in the context of autonomous driving and roundabouts.

wieviele Kreisverkehre und Kreuzungen wurden verglichen?

Even if roundabouts for the human driver seem to be a relief, it is necessary to investigate the challenges in regard to autonomous driving. In the following, we present an overview of the legal aspects in Germany.

Figure 2.2: Accidents in City Limits [3]

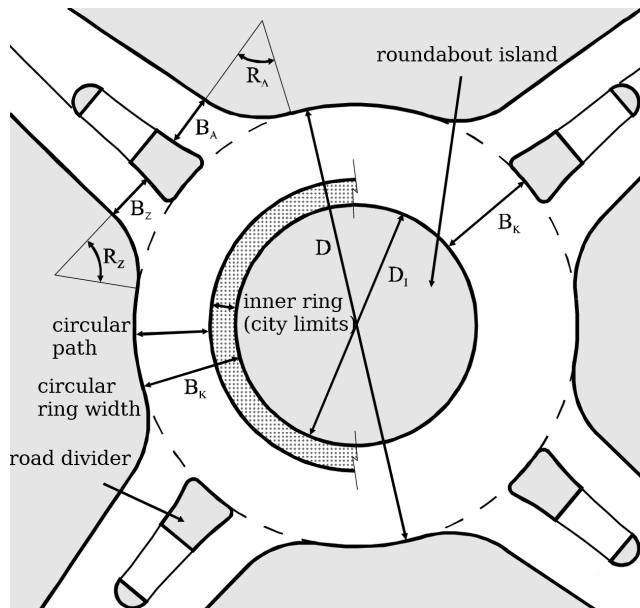


### 2.1.1 Roundabouts in Law

In Germany, there is no law stipulating the exact construction of roundabouts. Instead, the elements of the rural roads and city streets are dealt with in “Directives for the Design of rural roads” [24] and the “Directives for the Design of Urban Roads” [25]. These guidelines are also relevant to the choice of a convenient junction type when linking roads. The considerations discussed there are based on traffic variables, area-related characteristics, economic criteria and spatial planning or urban planning requirements. The guidelines also regulate the basic design and operational formation of roundabouts. The Directives for the Design of Urban Roads [25] are relevant for this dispute. Since the access the RAST ist limited, most of the information is coming from [23] whereupon RAST is based on.

### 2.1.2 Elements of a Roundabout

Figure 2.3: Definition of individual design elements and dimensions of a roundabout [23]



**Definition 2.1 (roundabout island)** *The roundabout island is the constructional area in the middle of the roundabout, which is surrounded by vehicles. For miniature roundabouts, the roundabout island is crossable. [23]*

**Definition 2.2 (circular path)** *The circular path is the road that serves to drive the roundabout island. An inner ring, if present, is not part of the circular path (VwV-StVO zu §9a V., Rn. 5). [23]*

**Definition 2.3 (circular ring with ( $B_K$ ))** *The structural width includes the circular track and a paved inner ring, if any. It is dependent on the outer diameter and the desired traffic routeing (one or two lanes). The edge strip width is oriented on the relevant continuous roadway. [23]*

**Definition 2.4 (outer diameter ( $D$ ))** *The outer diameter is measured at the outer edge of the circular ring. It is the essential measure for describing the size of the roundabout. [23]*

**Definition 2.5 (inner diameter ( $D_I$ ))** *The inner diameter is the diameter of the roundabout island. [23]*

**Definition 2.6 (road divider)** *The road divider is the structurally designed island between the circular exit and circular driveway. It serves to separate the circular exit and circular driveway, the management of the traffic, as well as the pedestrians and cyclists as cross-bordering aid. [23]*

**Definition 2.7 (lane width of the circular driveway ( $B_Z$ ) and circular exit ( $B_A$ ))** *The width of the circular driveway and exit is measured at the beginning of the corner. [23]*

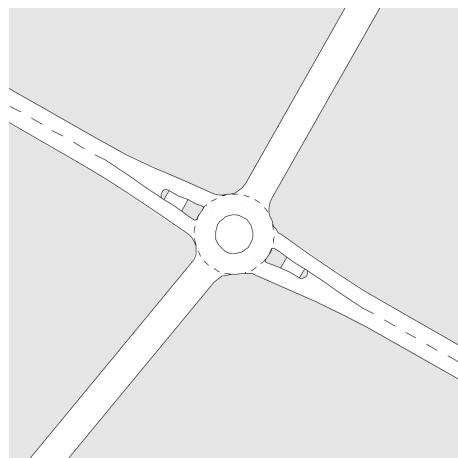
**Definition 2.8 (Corner rounding radius ( $R_Z$  and  $R_A$ ))** *This is the radius of the rounding at the right edge of the road between the circular driveway and the circular path. For a elliptical arch with a radius sequence of three different radii,  $R_Z$  is the radius  $R_2$  of the central arc. When the road edge is formed as a tractrix,  $R_Z$  is the smallest radius of the road edge. [23]*

### 2.1.3 Types of Roundabouts

There are several types of roundabouts, which are differentiated by the different application criteria and the partly different design principles according to the situation inside and outside built areas. Furthermore, a division is made as a function of its size. [23]

#### Mini Roundabout

Figure 2.4: Mini Roundabout  
[23]

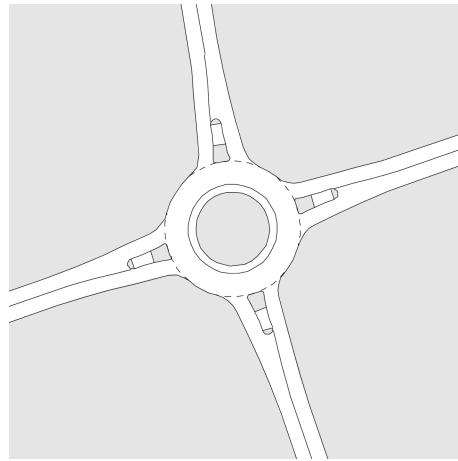


Within built-up areas, smaller outer diameters are possible under certain conditions. These roundabouts are called mini roundabout. The roundabout island

must then be capable of being passed over. The outer diameter should be at least 13 m, so that the circular island does not become too small. Larger outer diameters make driving easier. Outer diameters of more than 22m, however, do not offer any transport advantages. From an outside diameter of about 22 m, therefore, the installation of a small roundabout with 26 m is generally more convenient. Bypasses are generally not required in the areas where mini roundabout can be used.

### Small Roundabout

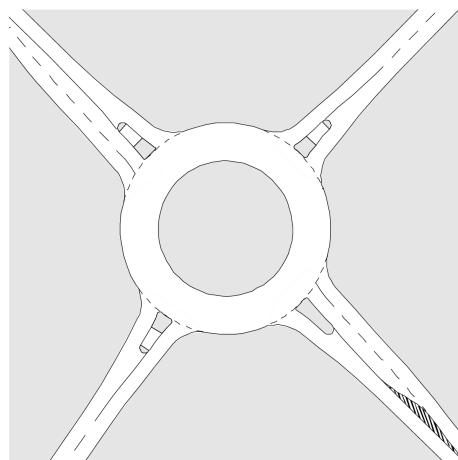
Figure 2.5: Small Roundabout [23]



The small roundabout has a single lane circular path and single lane circular driveways and exits. The roundabout island is not passable. The outer diameter must be at least 26 m. Bypasses can be set up for driving geometric reasons or to increase performance.

### Two-lane Passable Roundabout

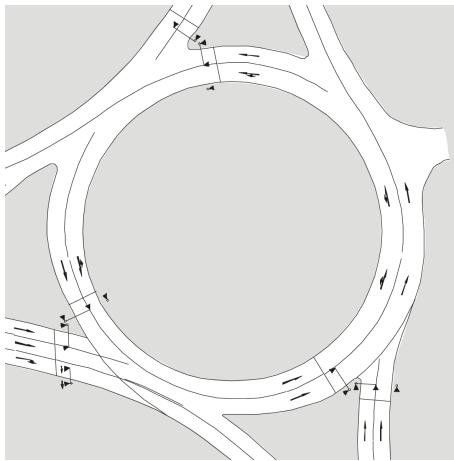
Figure 2.6: Two-lane Passable Roundabout [23]



If the capacity of the small roundabout is not sufficient and can not be ensured by the installation of bypasses, the circular path of a small roundabout can be designed to be two-lane driveable. At such a roundabout, the circular path is so wide that cars can travel side by side in a circle. If a further increase in the capacity is required, individual circular driveway can also be carried out in two lanes, if pedestrians and cyclists are not to be considered regularly. For safety reasons, circular exits are always carried out in single lanes. For geometrical reasons, the outer diameter must be at least 40 m for two-laned accessibility.

## Large Roundabout

Figure 2.7: Large Roundabout  
[23]



Large Roundabouts with two or more lanes marked by markers on the circular path should be operated with a light signaling system only, if the nodal point design and traffic control are closely coordinated.

---

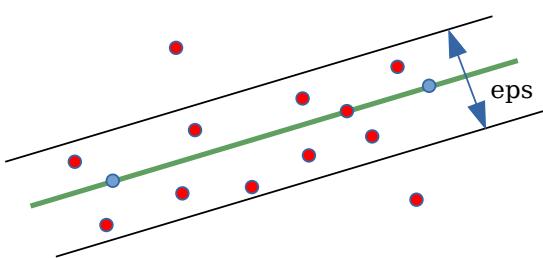
## 2.2 Random Sample Consensus

The Random Sample Consensus (RANSAC) [10] is an algorithm for estimating a model within a series of readings with outliers and rough errors. Due to its robustness, it is mainly used in the evaluation of automatic measurements primarily in the field of image processing. In this case, RANSAC is supporting traditional compensation methods such as the least squares method, which usually fail with a larger number of outliers, by calculating an outliers free data set, the so-called consensus set.

The RANSAC requires more data points than is required for the unambiguous determination of the model. From this set of data points randomly as many data points are selected, as necessary to the model unambiguously. From the remaining data, those which have a spacing which is smaller than a certain limit value are then selected. This set now represents the “consensus set”. If it contains a certain minimum number of values, a good model was probably found and the consensus set is stored. These steps are repeated several times. Then the subset that contains the most points is selected. Using this subset, the model parameters are calculated using one of the usual compensation methods. The RANSAC therefore has three parameters to determine, which influence the result.

missing reference to pics in dbscan and ransac

Figure 2.8: RANSAC [10]



- number of iterations

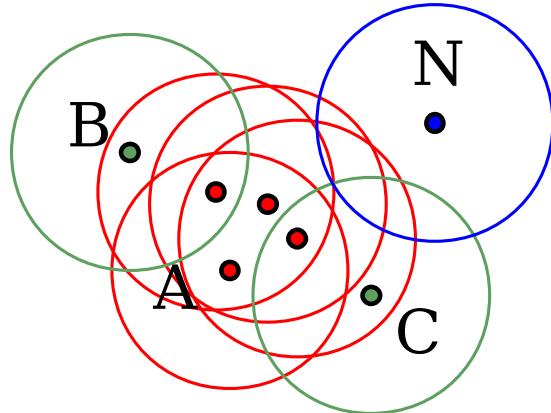
- minimum size of the “Consensus Set”
- distance threshold value (eps)

Note that the RANSAC is not a deterministic algorythm, beacuse of the random selection of the data points.

### 2.3 Density-Based Spatial Clustering of Applications with Noise

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [9] is a deterministic data mining algorithm for cluster analysis. The algorithm is based on density connectivity, that means, it binds points based on its distance to clusters, DBSCAN, iterates over all data points that have not yet been processed, each processed data point is marked as processed. A range query request is then made for each of these points. If the size of the returned neighborhood is lower then a certain limit, the point is marked as noise. Otherwise, a new clusrer is created by performing a new range request for each point in the neighborhood, if the point is not classified as noise, the Point is added to the cluster and marked as processed. This is repeated until all the points in the cluster are marked as processed, so no further points can be reached in the neighborhood.

*Figure 2.9: DBSCAN [9]*



The DBSCAN therefore has two parameters to determine, which influence the result.

- maximum distance of the neighborhood points
- minimum number of points required to form a dense region

### 2.4 Middleware OpenDAVINCI

Autonomous software is typically a distributed system, on today’s vehicles this system is based on ECUs and bus systems such as CAN, LIN. Distributed software makes it easier to integrate complex components within the system. In the area of autonomous driving, however, the historical structure of vehicles with ECUs and CAN is not optimal. [4] In order to handle the many components required, it is also advantageous to decouple components within an ECU or a computing unit. For this purpose, there are several middelwares which handle

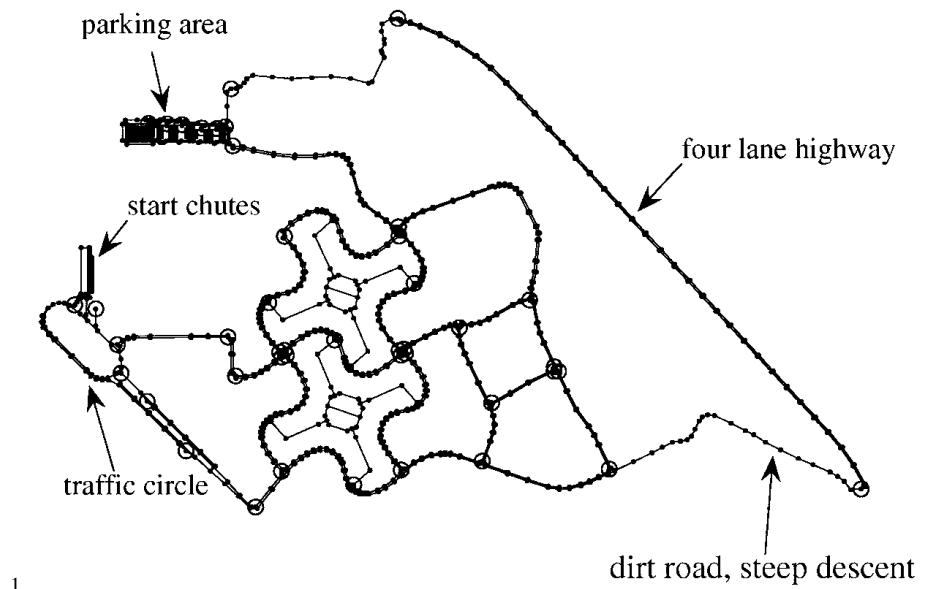
and abstract communication within the components. As part of the Copplar project, the OpenDaVINCI middleware is used here. OpenDaVINCI is a real-time runtime environment designed for autonomous vehicles. OpenDaVINCI is based on Hesperia. [2] Communication between the components in OpenDaVINCI is based on UDP Multicast, which enables real-time communication between components and computing units. [15] For the communication, OpenDaVINCI offers time-triggered transmitters and data-triggered receivers, from which the data-triggered receiver is used to connect our software. Furthermore, OpenDaVINCI offers many further functionalities, which can be used, for example the handling of World Geodetic System 1984 (WGS84), for the conversion of GPS coordinates into local Cartesian coordinates. For this purpose, a reference GPS position is required, which should not be too far away to keep the calculation error small.

# 3

## State of the Art

Auch wenn Autonomes Fahren ein aktuelles Forschungsthema ist, gibt es bisher keine mit bekannten Veröffentlichungen welche sich explizit mit Kreisverkehren beschäftigen. Um trotzdem einen Vergleich zur Sensorausstattung und Verkehrsbeobachtung zu haben werden an dieser Stelle einige Paper der DARPA Urban Challenge herangezogen. Die DARPA Urban Challenge ist ein Rennen zwischen autonomen Fahrzeugen, welches im Jahre 2007 von der Defense Advanced Research Projects Agency (DARPA) ausgetragen wurde. [5] Das Rennen fand in bebautem Gebiet einer verlassenen Kaserne des ehemaligen Air Force-Stützpunktes George Air Force Base statt, die Karte kann ist in fig. 3.1 zu sehen.

Figure 3.1: DARPA Urban Challenge Map



1

Die Fahrbahnen waren teilweise durch Linien markiert, teilweise durch Betonwände begrenzt. Alle selbstfahrenden Autos befanden sich gleichzeitig auf der Strecke. Auf ein-, zwei- und vierspurigen Straßen mussten Kreisverkehr, 4-Wege-Kreuzungen mit Stoppschildern, blockierte Fahrbahnen und Einfädel-situationen erfolgreich bewältigt werden.

Bei der Auswertung möchte ich mich vor allem auf die Finalteams des Wettbewerbes beschränken. Ausgewertet werden dabei welcher Typ Sensoren und welche Anzahl die Teams verwendet haben.

<b>Team</b>	<b>2D Lidar</b>	<b>3D Lidar (low res)</b>	<b>3D Lidar (high res)</b>	<b>Radar</b>	<b>Camera</b>
Cornell [6]	6	0	0	3	4
TerraMax [7]	3	1 (4)	0	0	3
Stanford Racing Team [17]	7	0	1 (64)	3	1
Tartan Racing [26]	3	0	1 (64)	2	1
Victor Tango [1]	1	1 (4)	0	2	1

# 4

## Methodology

siehe jens BA, das hier ist zu kurz

In the previous chapter we looked at the types of roundabouts and their components. We also reviewed the available test platform and its sensor technology. We have found that the detection of objects in other projects often combines several and more expensive sensors in order to ensure a reliable detection of other traffic users.

beleg

In der Research Question eins haben wir festgehalten, das wir die verwendung eines günstigen VLP-16 Sensors in einem Komplexen Verkehrszenario, die Verkehrsbeobachtung eines Kreisverkehrs evaluieren wollen.

For this purpose, an algorithm for recognizing and tracking objects with the help of the Velodyne VLP-16 is proposed and implemented. The difficulty lies in the use of a single and in a comparatively low priced environmental sensor, which obviously has not been developed as a standalone solution for this purpose.

beleg

In its current application, this sensor offers a comparatively low resolution in the area which is relevant for this work. Therefore, with many gradient-based algorythms used in similar projects Segmentation will often fail. For this reason, a ground plane based algorithm is implemented for the segmentation.

In addition, it is necessary to follow vehicles beyond their measuring horizon in roundabouts with built-up central islands and multi-lane roundabouts, in order to ensure a safe entry into the roundabout. For this purpose, a tracking and state estimation algorithm is developed in section , which should grant that.

section reference

In order to evaluate these algorythms, were collected several data collections at the Swedish AstaZero test area nearby Sandhult [fig. 4.1] For all experiments not carried out there, are carried out in a simulation made for this purpose. In this simulation a roundabout in a urban area with pavement and bikeway is designed, which the roundabout on AstaZero can not offer.

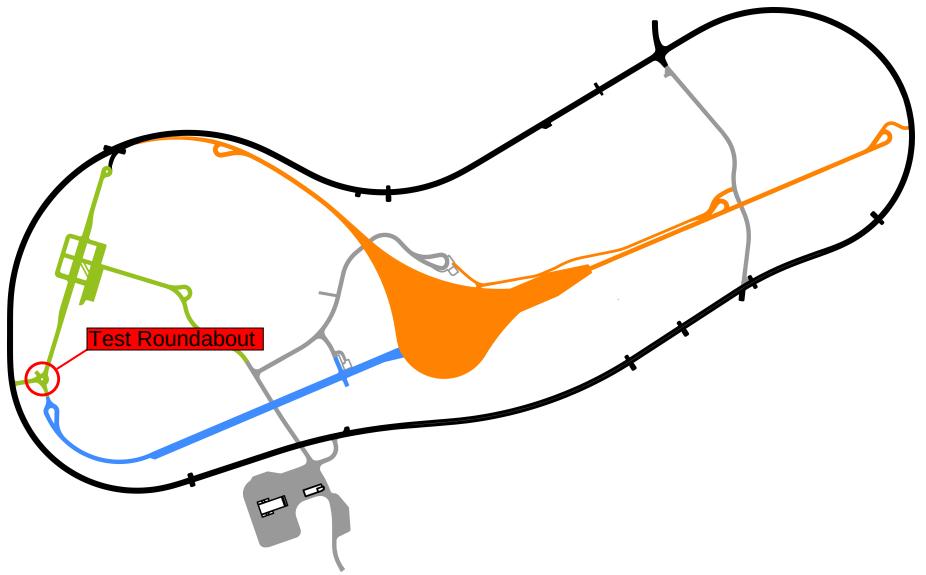
The evaluation is carried out manually by using of the graphically prepared measurement data. While the evaluation especially False-Negative and False-Positive detected obstacles will be discussed. Coarse outliers in the position or orientation of objects are also noted.

In order to evaluate the operability of the roundabout, a statemachine is imple-

mented which is intended to move the vehicle safely and accident-free through the roundabout. For this purpose, the simulation is monitored over a longer period of time, and the number of possible collisions is noted.

Figure 4.1: AstaZero Proving  
Ground

[http://www.astazero.com/wp-content/uploads/2016/09/%C3%96versiktsskiss\\_mod.pdf](http://www.astazero.com/wp-content/uploads/2016/09/%C3%96versiktsskiss_mod.pdf)



# 5

# Sensor Analysis

As already discussed, the selection of sensors will be limited to the Applanix POS-LV and the Velodyne VLP-16. Since the Velodyne is the more important sensor for the recognition of other traffic users, it is now investigated how far the sensor is suitable for the detection.

## 5.1 Theoretical Analysis

Under section 2.1.3 we have established four different types of roundabouts. In order to evaluate the practical use of the Velodyne VLP-16 is now analyzed, how far the Velodyne is able to overlook the situation. For this purpose we will look at the size ratios.

Roundabout Type	min Size	max Size
Mini Roundabout	13m	22m
Small Roundabout	26m	40m
Two-lane Passable Roundabout	40m	-
Large Roundabout	>40m	-

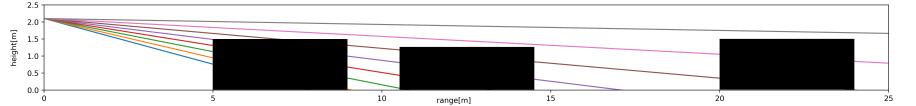
For a comparison to the Velodyne VLP-16, we will look at the visual range of the sensor in the 2D side view. The sensor provides 16 measurements at the same azimuth angle, at a distance of 2 degrees. Since for the evaluation, and unfortunately also for the recognition, the angles greater than 0 degrees are not relevant, we only see the eight angles in the range of -1 to -15 degrees.

In the evaluation we assume that we try to recognize a small car with a length of 4 meters and 1.5 meters height. In doing so, we start from the ideal situation that the vehicle drives to us. Furthermore, we assume a mounting height of the sensor of 2.1 meters as this is finally the mounting height of the sensor on the test vehicle.

**Range 0-25m** For the range from 0m to 25m, we see in fig. 5.1 that in the worst case we still get at least one measurement from the sensor. Is the vehicle very much at the Sensor, we get even a large part of the available resolution, whereby close to behind lying flat vehicles can be hidden. Therefore, we can

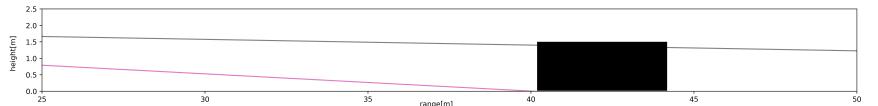
see that the mini roundabout can be covered with the sensor, but we have to think about hidden obstacles.

Figure 5.1: Velodyne VLP-16 -  
Range 0-25m



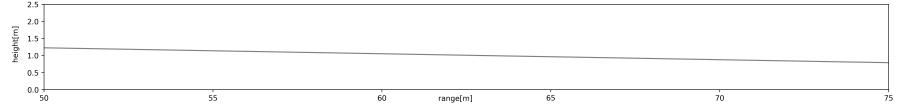
**Range 25-50m** For the range of 25m to 50m we see in fig. 5.2 that we can only get two measurements. For our assumed small car with a height of 1.5 meters, however, we do not keep a dead zone. However, for most measurements, we get only one measurement point and thus a very low resolution. Therefore the detection of vehicles in a small roundabout is already a challenge for the segmentation of the vehicles

Figure 5.2: Velodyne VLP-16 -  
Range 25-50m



**Range >50m** For the area greater than 50m, we see in fig. 5.3 that the resolution remains the same in the same way as for closer objects. However, there is a higher likelihood that vehicles have already been hidden in the previous area.

Figure 5.3: Velodyne VLP-16 -  
Range 50-75m



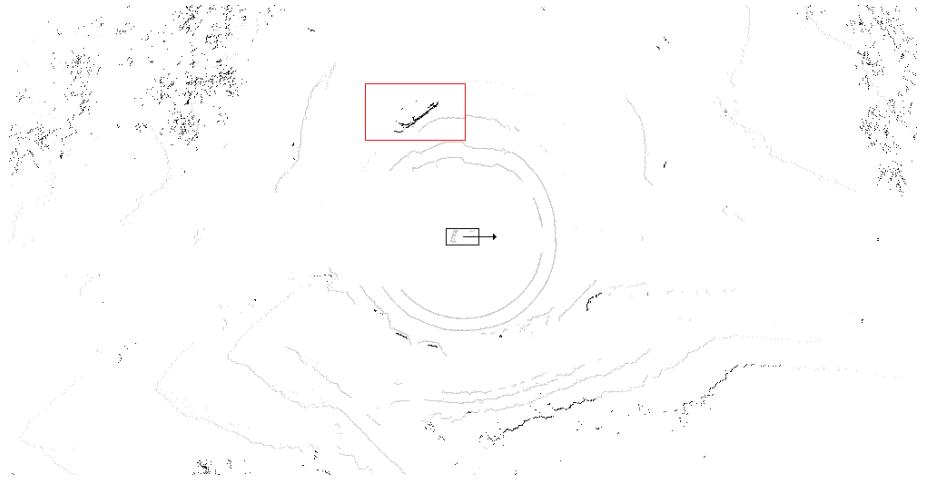
As we can see, we can excellent cover the area up to 25 meters and thus observe the mini roundabout completely, for larger circular traffic the sensor is only conditionally suitable. For all the following considerations, we shall consider only miniature circuits or smaller small roundabouts. For all other roundabouts, evaluate whether it is at all necessary to observe all the vehicles in the roundabout as the distance from driveway to driveway increases accordingly. It is possible to consider the roundabout as a simple yield sign situation.

## 5.2 Practical Analysis

Since we have already looked at the theoretical considerations, we look at the practical measurements of the sensor. To do this, we can see a measurement on the test roundabout on AstaZero in the bird's eye view in fig. 5.4.

It can be seen here that the measurements in the front and especially in the rear region have gaps. Unfortunately, this was not prevented during installation of the Velodyne since the holding pre-treatment of the sensor does not provide a further height adjustment and no invasive changes should be made to the test vehicle. Therefore, a compromise had to be concluded for the assembly, either to lose measurements in the front area, or to lose in the rear. The missing

Figure 5.4: Velodyne VLP-16 -  
Hidden Layers

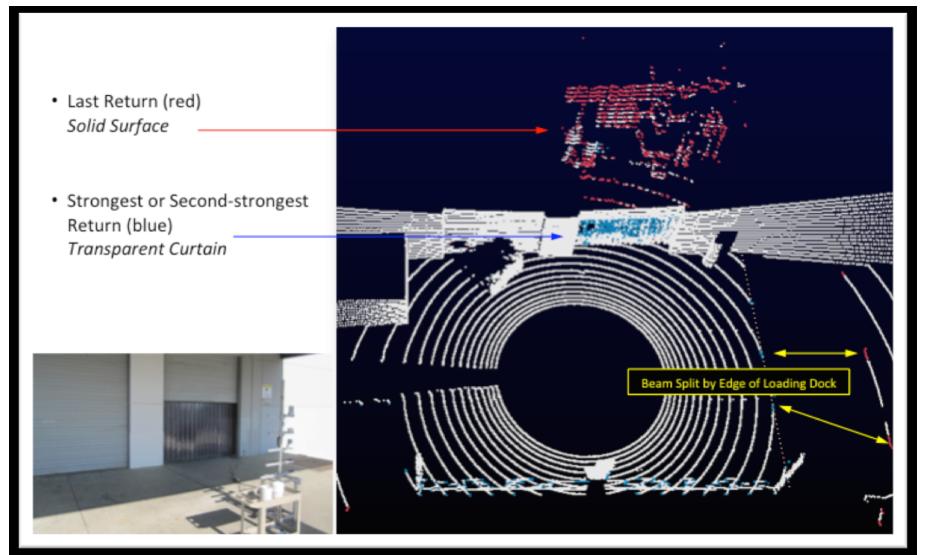


measurements are obscured by the vehicle roof. The smaller gaps in the front area come from the GPS antenna of the Applanix system and unfortunately could not be prevented.

Since the monitoring of the rear area is not important for the roundabout observation, the missing measurements in the rear area are negligible. Since we have the highest resolution in the front sensor range of the sensor, the missing measurements in the front range are also not important, because these are only in the first measurement layer.

Further in fig. 5.4 to see it is another test vehicle (red box). It can be seen from this that we also get measurements in the rear of the vehicle, which is actually hidden. This is explained by the fact that the sensor has several return modes, which allows it to see through transparent objects. This can be seen in the example in fig. 5.5

Figure 5.5: Velodyne VLP-16 -  
Return Modes [27]



The sensor can be configured to return both measurements (Strongest Return and Last Return). Since the vehicles to be observed are not completely transparent and thus always provide sufficient measurements in the front area, in addition to keeping the data rate small, VLP-16 was configured to provide only the Last Return.

---

# 6

# Objekt Detection

In the following chapter, we will present the Velodyne VLP-16 with the recognition of traffic users. We begin with the segmentation, that is, in the first step all measured values which do not represent potential traffic are removed. In the next step, the remaining measured values are combined with the aid of a clustering algorithm. In the third step the objects are abstracted and tracked, that is, we try to establish a relationship between the objects between different time steps. This is necessary to determine time-dependent variables such as the speed and rotation rate of the detected objects. In the last step, the objects are classified into types using the predetermined parameters.

---

## 6.1 Ground Removal

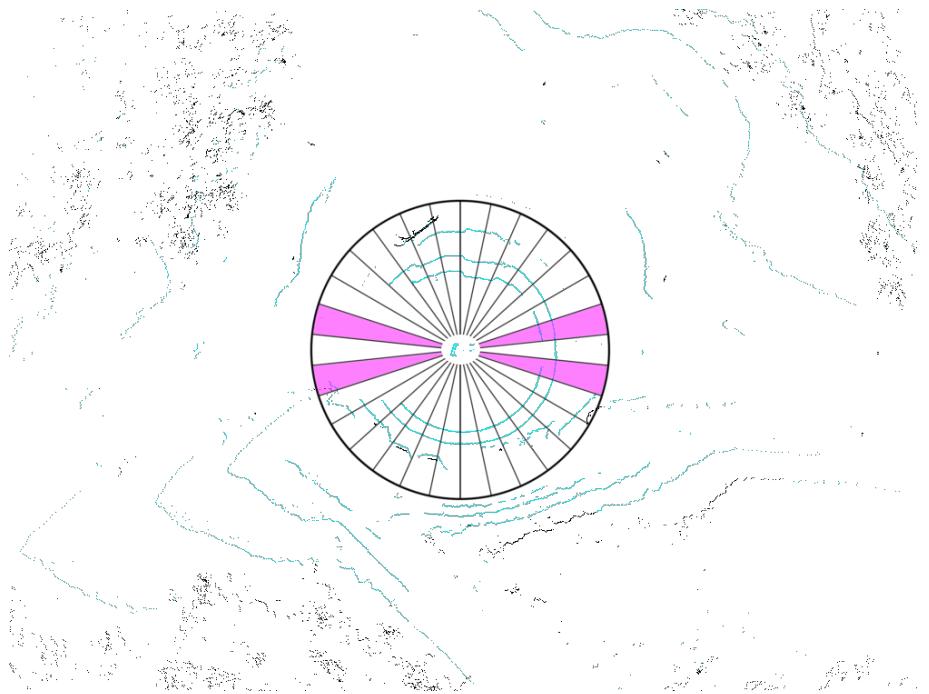
In order to recognize objects in a PointCloud, it is necessary to know which measurements belong to the ground and which belong to objects. There are many ways to achieve this. The most naive method is to remove the bottom plate by its Z coordinate. However, this method has many drawbacks, on the one hand, the LIDAR sensor has to be mounted exactly straight on a vehicle, on the other hand the vehicle must have a very stiff chassis in order to prevent a possible tilt of the sensor. Furthermore, this only allows the removal of planar surfaces, also flat non-hilly grounds. Another common method is the removal of the base plate on the basis of a statistical mean value [29]. However, this method also requires a calibration of the sensor distance to the ground. And the determination of further threshold values which are dependent on the environment. The advantages of both methods are their low computational performance and runtime  $\mathcal{O}(n)$ . Better methods such as gradient-based explanations algorithms need a starting point that can be identified as a bottom plate. A further possibility is the description of objects as convex objects [18], which can also be described on the basis of the gradients. The advantage of this method is that no initial position is required for the floor plate, but this is very processing power intensive.

For our application with the Velodyne VLP-16, the problem is that the resolution of the sensor is very low in height. Depending on the distance of the vehicle within the required range, only two layers fall on the test vehicles, which means that gradient-based methods fail reliably, because the gradients

are too small and the depletion of the necessary thresholds leads to frequent false positives. The method of the statistical mean and the method based on the Z coordinate, suffer from the chassis of the Volvo XC90 SUV. The height of the vehicle varies by several centimeters by changing the driving profile (Sport / Eco, etc.). Even slightly increased speed in the roundabout (about 30 km/h) lead to a clear lateral inclination of the vehicle. Therefore, another method is proposed. The detection of a base area in the measurement data.

For the recognition of the ground floor we assume the following assumptions: the road can be presented approximately as a plane in  $\mathbb{R}^3$ . Further, the ground floor is the lowest area in the searched area. Therefore, in the first step, the data set in polar coordinates is divided into segments segmentated into 30 of cake piece formed pieces. From this pieces, two segments [fig. 6.1] are then selected in front and rear, which are not neighbored. The selection of the segments follows from the assumption that the road is in front or behind the vehicle. In future, the selection of the segments could also be optimized with the help of the vehicle steering angle, or the valid range can be provided by a lane detection.

Figure 6.1: LiDAR Segments



Within these segments a search for the 10 measurements with the lowest Z value is made. The search is limited to the three lowest layers (-15, -13 and -11 degrees), since the measurements of all higher layers are too far away. The division into segments is therefore necessary in order to prevent all measured values running into a single local minima.

From this pre-filtered readings three are now randomly selected for a RANSAC, which will run iteratively.

From these three points a plane is now formed in the Hesse normal form, which allows an efficient distance calculation to other points. After that, we collect all other points from our set of minima, using a distance criterion. If we can find more than 10 other points in our minima set, a new plane and their error is calculated from the plane and the new Collected Points by a Planefitting Algorithms [section 6.1.1]. The error is calculated over the sum of the squared

welcher threshold wurde gewählt, und warum?

distances of all points to the plane.

However, before we add the plane as a possible solution candidate, it is checked whether the layer is within a plausible parameter range. This means that the distance of the plane should move between 1.9m and 2.2m, which corresponds approximately to the mounting height of the Velodyne sensor.

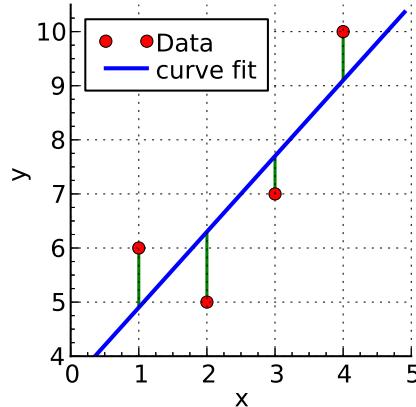
The number of iterations of the RANSAC is limited to 50. After the run of the RANSAC, the plane with the lowest error is taken and all points in the pointcloud are marked by their distance to the plane as ground. As distance threshold, an optimum value of 0.5 m was determined experimentally.

### 6.1.1 Planefitting

Macht das überhaupt sinn? Laut ransac sollte das Planefitting nicht iterativ ausgeführt werden???? Fuck!

For plane fitting usually a Singular Value Decomposition (SVD) is used [19, 20, 22]. SVD has a complexity of  $\mathcal{O}(\min\{mn^2, m^2n\})$  [13], since the planefitting inside of the RANSAC is very often called with a large number of points, running to the SVD within the RANSAC causes a very high running time. For this reason, a Linear least Squares (LLSQ) algorithm with some optimization is used here. When using the LLSQ, it is important to note that the LLSQ is not optimizing the distance of the points to the plane but the distance of the points along an axis (in our case the z axis), see fig. 6.2. This can lead to problems, if the points are scattered far apart from the optimal plane. However, since we are using our preselected points from our the RANSAC distance criterion, this poses no problem.

Figure 6.2: Linear least Squares (LLSQ) [28]



The representation of a plane in coordinate form is as follows:  $a\vec{x} + b\vec{y} + c\vec{z} + d = 0$ . Since we consider a plane in  $\mathbb{R}^3$ , this system of equations is over-determined. Since we want to optimize our plane in the direction of the Z-axis, we set parameter c to 1 and can now simply solve our equation system by z:  $a\vec{x} + b\vec{y} + d = -\vec{z}$ . The vectors  $\vec{x}, \vec{y}, \vec{z}$  represent the points to be fitted. In matrix notation:

$$X\vec{\beta} = \vec{z}$$

$$\begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ \dots & \dots & \dots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ d \end{bmatrix} = \begin{bmatrix} -z_0 \\ -z_1 \\ \dots \\ -z_n \end{bmatrix}$$

This system usually doesn't have a solution, but our real goal isn't finding an exact solution for  $\vec{\beta}$ , we want to find a good approximation  $\hat{\vec{\beta}}$  for this:

$$\hat{\beta} = \min (||\vec{z} - X\vec{\beta}||^2)$$

We can do this by multiplying our equation by the transpose of our point matrix  $X$ :

$$(X^T X) \hat{\beta} = X^T \vec{z}$$

$$\begin{bmatrix} x_0 & x_1 & \dots & x_n \\ y_0 & y_1 & \dots & y_n \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ \dots & \dots & \dots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ d \end{bmatrix} = \begin{bmatrix} x_0 & x_1 & \dots & x_n \\ y_0 & y_1 & \dots & y_n \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} -z_0 \\ -z_1 \\ \dots \\ -z_n \end{bmatrix}$$

This equation system can now be solved with the inverse of  $(X^T X)$ . Since the calculation of inverse matrices with  $\mathcal{O}(n^3)$  is also expensive, now another trick to save computing power. After multiplying with the transpose we get:

$$\begin{bmatrix} \sum x_i x_i & \sum x_i y_i & \sum x_i \\ \sum y_i x_i & \sum y_i y_i & \sum y_i \\ \sum x_i & \sum y_i & N \end{bmatrix} \begin{bmatrix} a \\ b \\ d \end{bmatrix} = \begin{bmatrix} \sum x_i z_i \\ \sum y_i z_i \\ \sum z_i \end{bmatrix}$$

The sums in the boundary areas of the matrix  $X$  and the vector  $\vec{z}$  are good to see. We can set this to zero if we define all points relative to the mean point of all points, ie  $P_i = P_i - \bar{P}$ . Now we get:

$$\begin{bmatrix} \sum x_i x_i & \sum x_i y_i & 0 \\ \sum y_i x_i & \sum y_i y_i & 0 \\ 0 & 0 & N \end{bmatrix} \begin{bmatrix} a \\ b \\ d \end{bmatrix} = \begin{bmatrix} \sum x_i z_i \\ \sum y_i z_i \\ 0 \end{bmatrix}$$

Now we can also set  $d$  to zero, because if all our points are relative to the mean point, then our plane always runs through this point. Therefore we now can get rid of a complete dimension:

$$\begin{bmatrix} \sum x_i x_i & \sum x_i y_i \\ \sum y_i x_i & \sum y_i y_i \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum x_i z_i \\ \sum y_i z_i \end{bmatrix}$$

The equation system can now be solved with the Cramer's rule

$$D = \sum x_i x_i \cdot \sum y_i y_i - \sum x_i y_i \cdot \sum x_i y_i$$

$$a = \frac{\sum y_i z_i \cdot \sum x_i y_i - \sum x_i z_i \cdot \sum y_i y_i}{D}$$

$$b = \frac{\sum x_i y_i \cdot \sum x_i z_i - \sum x_i x_i \cdot \sum y_i z_i}{D}$$

$$\vec{n} = [a, b, 1]^T$$

It should be noted that the determinant can not be zero or near zero. However, since the angle between the vehicle and the plane is always close to 90 degrees, the determinant is typically very large. If the determinant is nevertheless close to zero (not equal to zero), the calculation is carried out in spite of the fact that

this also leads to a large error in the fitting. This is desirable at this point since the RANSAC sorts out invalid layers based on the error. If the determinant is extinct zero, the calculation is continued with a small value for D.

From the normal vector  $\vec{n}$  and the mean point  $\bar{P}$ , we can again determine the Hessian normal vector.

In the end, we have been able to break down the algorithm from  $\mathcal{O}(m^2n)$  to  $\mathcal{O}(n)$ .

## 6.2 Clustering

In current work with 3D-LIDAR data, the data is often first projected into a heightmap. Then directly adjacent measurements are combined with similar measured values. Alternatively, the measurements are also summarized by means of a distance criterion. The former method has the disadvantage that individual outliers cause the object to disintegrate in several clusters. The latter is usually combined with a KD tree or a similar data structure, which typically entails high costs for the creation. Since the tree has to be rebuilt after each 360 degree measurement is the problem

Here a method is suggested which combines the advantages of both methods. It is therefore necessary to know how the data from the OpenDAVINCI Middleware is delivered. Because the OpenDAVINCI Middleware relies on the transmission of the data with UDP Multicast, the data is transferred in a compact form that fits into a single UDP frame.

CompactPointCloud
startAzimuth : float
endAzimuth : float
entriesPerAzimuth : uint32
distances : byte[]
getStartAzimuth : float
...

In this case, we assume a constant rotational rate of the sensor output, which results in an equidistant distance of the measured values. The number of measurements per azimuth is recorded in entriesPerAzimuth and corresponds to the Velodyne VLP-16. In order to arrive at the actual measured values, two distance values must be merged and converted to an unsigned 16-bit integer, which then contains the measurement in cm. In each case, 16 of these values result in a measuring frame in which the polar angle must be mapped to a range between -15 and +15 degrees. After the spherical data of each measurement point have been reconstructed, they are converted into Cartesian coordinates and stored in a point data structure.

iterativen ablauf des ganzen Algorythmus  
klar machen

Point
azimuth : float
measurement : float
visited : bool
isGround : bool
point : vector3f
getAzimuth : float
...

This is stored again in a static two-dimensional array: Points [2000] [16]. The order of the data is kept thereby to allow efficient access to the values based on their azimuth angle. This data structure now provides the base for the subsequent clustering alorythm.

On this basis, a DBSCAN [9] is now executed. The DBSCAN algorithm has the following advantages. In contrast to the K-Means algorithm, for example, it is not necessary to know how many clusters exist. The algorithm can recognize clusters of any shape (e.g., not only spherical). Furthermore, DBSCAN can deal with noise and recognize as this. This makes the DBSCAN optimal candidates for us, because our objects can be have any shape and measurement from the sensor can have errors within a real scenario, which can be assumed as noise effect. In fact, DBSCAN is itself of linear complexity. However, most computing times are caused by the neighborhood calculation. But instead of the range request via a tree structure, we here profit from the behavior, that measurements in a small neighborhood have a similar azimuth angle. For this purpose, we examine two additional entries for each measured value to the left and right in our array. Therefore, we need to check  $5 \cdot 16 = 80$  values. The run-time of the range request can therefore also be performed in linear complexity. All measured values previously classified as ground are skipped during the calculation. In addition, the construction of a KD-tree is omitted, which also leaves us a run-time advantage. All clusters are stored by the algorithm as a list of references to the original array to avoid unnecessary copying.

## 6.3 Tracking

The task of tracking is to establish a relationship between the measured values over time. The tracking is divided into two sections. Tracking of the clusters from the DBSCAN and the creation and tracking of the objects.

datenstruktur Objekte hinzufügen, weil positionen und attribute verwirrend

### 6.3.1 Cluster Tracking

For cluster tracking, we assume that objects move only slowly from time step to time step, and the shape of the cluster also changes only slightly. This is important because the position of a cluster is defined by a mean value point. Tracking is performed in  $\mathbb{R}^2$ . In the initial step, an ascending ID is assigned to each cluster. In each further step, each new cluster is assigned the ID of the old cluster, which has the smallest distance over time. For this distance, there is a generous upper bound of 3m, which was determined experimental, clusters that are not within this boundary are given a new ID. This results in the fact that multiple clusters can be assigned the same ID, which is important because objects sometimes break down into multiple clusters.

### 6.3.2 Object Tracking

The basis for the object tracking are the previously tracked clusters. In the initial step, objects are created from all clusters with the same IDs. Which apart from the measured values contain further parameters for the description of the object. This includes values such as the potential size and position of the object as a rectangle, the speed of the object and its direction, the ID and type of the object, and its confidence. The ID is derived from the ID of the initial used cluster

Der vorerst wichtigste Schritt beim Updatevorgang ist die berechnung der Bewegungsrichtung eines Objektes, da folgende Berechnungen auf dieser basieren. Bei der Berechnung der Bewegungsrichtung ist zu beachten, dass die Bewegung des eigenen Fahrzeuges herausgerechnet werden muss. Dazu werden die Positionsdaten des Applanix POS-LV genutzt. Da sowohl die Positionsdaten des Applanix Systems, als auch die Erkannte Postition des Fahrzeuges Fehlerbehaftet sind, wird die Bewegungrichtung nur ab einer minimalen Bewegung von 2m geupdatet.

In each further step, all clusters with the previously identical ID are used to update the objects. New objects are created from clusters with new IDs.

The first important step in the update process is the calculation of the movement direction of an object, because the following calculations are based on this. When calculating the direction of movement, it must be noted that the movement of our own vehicle must be separated out. The position data of the Applanix POS-LV are used for this purpose. Since both the position data of the Applanix system and the detected position of the vehicle have errors, the direction of movement is updated only with a minimum movement of 2m.

$$\begin{aligned}\Delta x &= P_x(t) - P_x(t-2m) + \Delta C_x \\ \Delta y &= P_y(t) - P_y(t-2m) + \Delta C_y \\ \theta &= \text{atan}2(\Delta y, \Delta x)\end{aligned}$$

with  $P$  - position of the object,  $C$  - position of our own vehicle

The result can be viewed in fig. 6.3. It is easy to see that the direction of movement (arrow) does not conform with the orientation of the object (black), but the bounding box is correctly oriented. How this calculation comes from is clarified in the following.

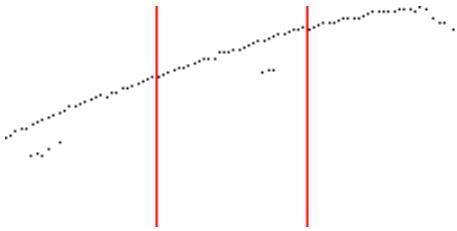
Figure 6.3: Obstacle Movement



Based on the direction of movement, the alignment of the vehicle is now calculated. To do this, all clusters assigned to the object are grouped together and rotated by  $-\theta$ . Then the object is divided into 3 equal segments (fig. 6.4).

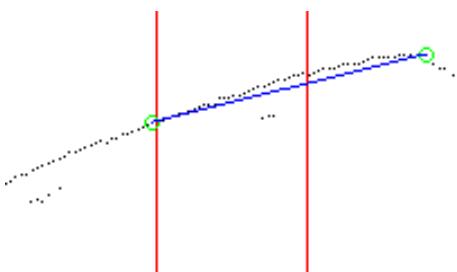
It is also determined whether the object is above or below the x-axis. This is important because we need to know which side of the object we are measuring.

Figure 6.4: Obstacle Cutting



If the object is below the x-axis, then the y-value is maximized in the next step; if it is above, it is minimized. In the following, we assume that the object is under the x-axis. Therefore we maximize the y-values in the left and right segment of the divided obstacle. The division into 3 segments is necessary to prevent the two maxima from running into the same point and the distance of the points doesn't become too small, which would cause a great inaccuracy. With these points  $(\vec{R}; \vec{L})$  a correction of the rotation of the object is now calculated:

Figure 6.5:  $\theta$  - Correction



$$\Delta x = R_x - L_x$$

$$\Delta y = R_y - L_y$$

$$\theta_{\text{correction}} = \text{atan2}(\Delta y, \Delta x)$$

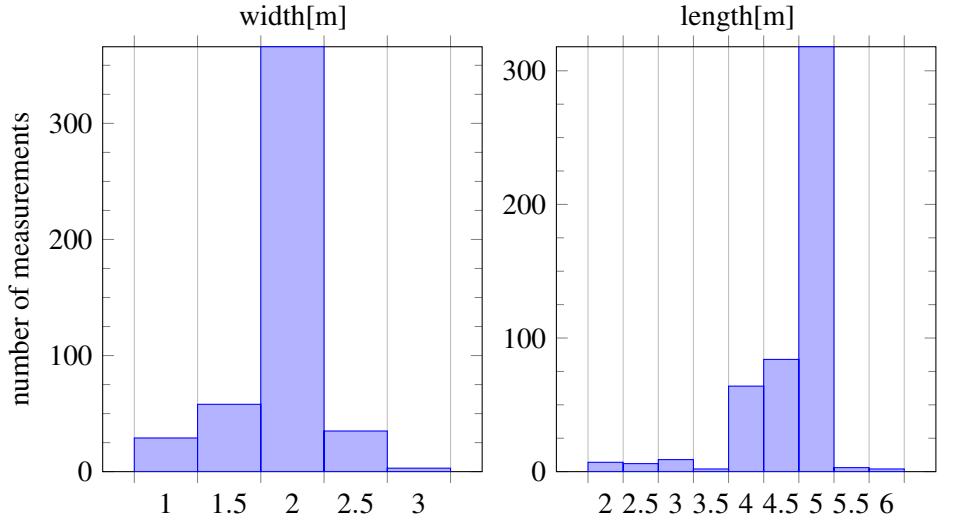
After applying the correction, the size of the obstacle is calculated. For this purpose, the maximum and minimum x and y values of rotated objects are used. With these values, a histogram for the length and width of the obstacle, rounded to 0.5 m, is now established over time. The most probable value is then selected. This causes the size of the obstacle to change more often before the size converges to a stable value. Since the size of the object can begin to be very small, there is a lower limit of 0.9m for both values. Measured values for a sample object can be seen in fig. 6.6.

skalierung angleichen, achsenbeschriftung

Easy to see, a width of 2m and a length of 5m is calculated for the object. In this case the object is a Volvo S60, which has outer dimensions of approximately 1.9m and 4.6m, whereby the deviations are correctly within, the rounding of the values. For the subsequent filtering of the measured values with the help of an extended Kalman filter, the position of the vehicle from the center of the bounding box is now determined, using the mean value of the four corner points

However, the position used for the calculation of the direction of movement is different, since the so calculated position is not yet available at this time and the position is very unstable shortly after the initial recognition due to the frequent dimensional changes. Therefore, the maximum x coordinate of the rotated clusters is always used as the position. This position can also be

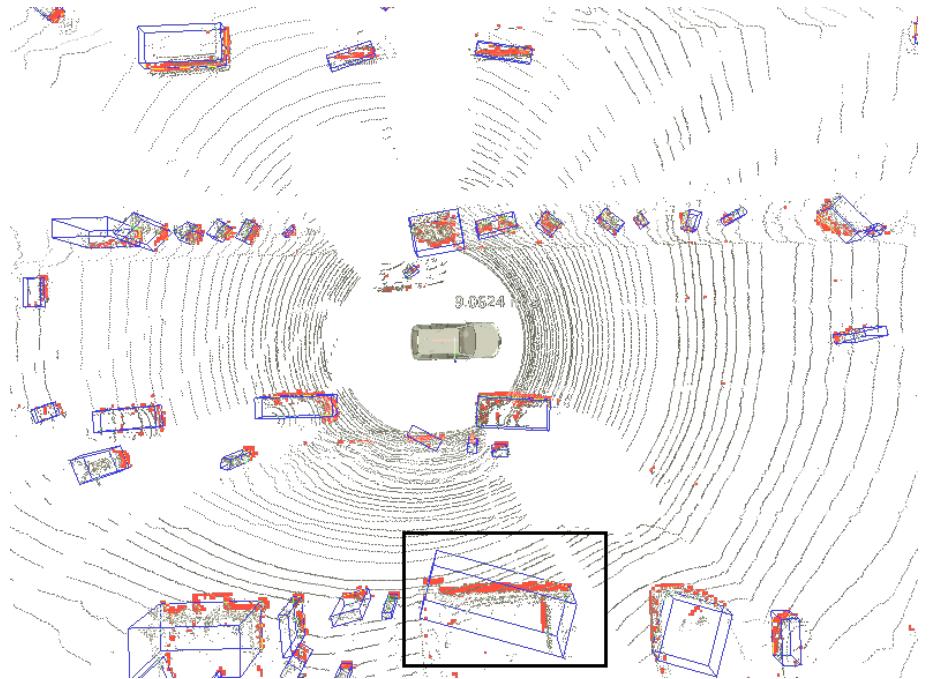
*Figure 6.6: Object Size Histogram*



examined in fig. 6.3, as a small green circle. Since  $\theta$  in the initial time step is zero, this corresponds to the global maximum x-coordinate of the cluster. This leads to the assumption that the object moves in the positive x direction in the initial time step. For objects where this is not the case, this leads to a short-term oscillation of the orientation, which however quickly stabilizes over time.

One thing or another may be wondering why the Boundingbox was calculated so elaborately. A simple way to calculate a bounding box for the objects would be the calculation of the minimum bounding box over the convex hull, as it is done in many other works [29, 12]. The minimum Bounding box, however, under certain circumstances does not provide the desired result, on the one hand you have only the size of the current measurement and on the other hand it can provide a wrong orientation as seen in fig. 6.7.

*Figure 6.7: Error with minimum Boundingbox [12]*



However, since the orientation and position of the objects are used as input

for the subsequent Kalman filter, which reacts very sensitively to false orientations, the algorithm was developed.

### 6.3.3 Object Confidence

implementierung Konfidenz wert auf 1 setzen und entfernen, wenn wert unter 1, oder objekte auf basis ihrer Position entfernen

In order to prevent the detection of false positives immediately recognized as an object and thus to influence the subsequent logic, a confidence value is now introduced. Before a recognized object is considered to be valid, this must achieve a certain confidence value. The initial confidence value of an object is zero. The confidence value is increased by one if the object can be tracked in two consecutive time steps and satisfies the following conditions:

wie ergibt sich Höhe und Breite! (aus der Orientierung.... erklären)

- The width of the object must be less than the length of the obstacle plus 1.5m
- The length of the obstacle must be less than 10m
- The width of the obstacle must be less than 4m

If one of these constraints is not met, the Confidence value is halved. For an object to be considered kind, it must reach a confidence value of 3 so that an object can only be recognized after at least 3 three iterations. This value was empirically determined and represents a tradeoff between fast detection and filtering.

---

## 6.4 Classification

text zu listing anpassen

implementierung Plausibilitätstest

filterung in code entfernen, weil Größe bereits gefiltert wird

Now a classification of the objects is made by their size, there is no classification according to moved and immobile objects. Only pedestrians, cyclists, vehicles, and others are differentiated. The size of the objects is used as a classification criterion. The classification is as follows:

**pedestrian:** length < 1.5 and width < 1.5

**cyclist:** length < 2 and width < 1.5

**car:** length < 10 and width < 4

**undefined:** length >= 10 and width >= 4

Furthermore, a plausibility test is carried out using the speed. Thus, a pedestrian must not exceed a speed of 10km/h and the maximum speed for a cyclist is 30km/h. Since no sensible speed limit can be assumed for vehicles, we use the change of orientation in their place. The maximum rotation rate is assumed to be 0.3 rads/sec from [14]. Since the value is an upper limit, we assume a slightly higher value of 0.4 rads/sec. If one of these values is exceeded, the confidence is also halved.

## 6.5 State Estimation

Following the tracking, a state estimation is performed on the detected objects. This is necessary because objects can be hidden during the movement, be it by other moving objects or buildings. An estimation of the state beyond the recognition horizon allows us to make a statement about the position of objects which are not visible at the moment. Furthermore, it allows you to easily rerecognize objects that have not been recognized at a time, ie to assign the same ID to the object as before.

*kalman nicht nur für Prediction auch für filtering*

From the previous tracking, we can get the current position, speed, rotation and rotation speed. For an estimation of the state with the for vehicle common the bicycle model, we miss the wheelbase and the weight of the vehicle. Therefore, we must limit ourselves to a relatively simple “Constant Turn Rate and Velocity” model. This allows us to use the same model for all classes of objects. Because this model can also be used on pedestrians and cyclists.

*reference*

### 6.5.1 Constant Turn Rate and Velocity Model

The state vector [21] of the CTRV model looks as follows:

$$\vec{x}(t) = [x \ y \ \theta \ v \ \omega]^T$$

x - Y Axis  
y - X Axis  
 $\theta$  - Object Yaw Angle  
 $v$  - Object Velocity  
 $\omega$  - Yaw Rate

The dynamic matrix is obtained by a non-linear state transition:

$$\vec{x}(t+T) = \begin{bmatrix} \frac{v}{\omega}(-\sin(\theta)) + \sin(T\omega + \theta) + x(t) \\ \frac{v}{\omega}(\cos(\theta)) - \cos(T\omega + \theta) + y(t) \\ \omega T + \theta \\ v \\ \omega \end{bmatrix}$$

Da es sich bei der Dynamikmatrix um einen nichtlinearen Zustandsübergang handelt benötigen wir noch die Jacobi Matrix. Die Matrix wurde mit Sympy generiert, und auf Grund ihrer Größe eingekürzt.

$$J(\vec{x}) = \begin{bmatrix} 1 & 0 & \frac{v}{\omega}(-\cos(\theta)) + \cos(T\omega + \theta) & \dots & \dots \\ 0 & 1 & \frac{v}{\omega}(-\sin(\theta)) + \sin(T\omega + \theta) & \dots & \dots \\ 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Im Update Schritt des Filters muss beachtet werden, dass wir mir Rotationen arbeiten die im Bereich von  $[-180 \leq \theta \leq 180]$  definiert sind und desshalb Singularitäten auftreten können. Da unsere Messdaten verbraucht sind, kann es

vorkommen, dass unser State bei 179 Grad ist und mit einem Wert von  $-179$  geupdated werden soll. Nehmen wir der einfacheit an, dass beide werte gleich gewichtet sind, ergibt sich nach dem update ein Wert von 0 Grad, dies ist offensichtlich falsch. Deshalb, wird vor jedem updateschritt bei bedarf 360 Grad addiert oder suptrahiert, do das der Abstand zwischen State und Messwert minimiert wird. Nach dem Updae wird der interne State wieder auf den bereich  $[-180 \leq \theta \leq 180]$  normiert. Betrachten wir nun wieder unser Beispiel mit den korrigierten Werten, State:  $179 - 360 = -181$  Grad und Messwert:  $-179$  Grad erhalten wir den korrekten Wert von  $-180$  Grad.

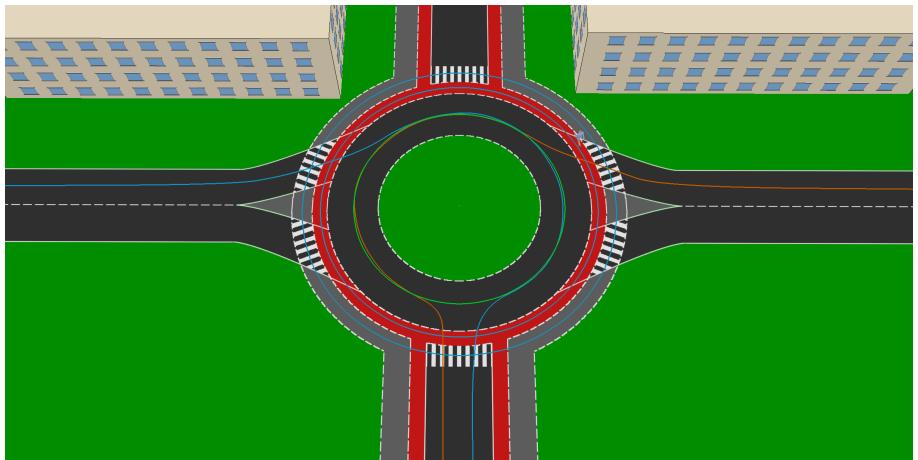
### 6.5.2 Prediction

an object is within the range of the Velodyne sensor and is not detected in the subsequent time step, the prediction step of the Kalman filter is still executed. This is done as long as the uncertainty of the position passes a certain threshold. If one of these conditions does not apply, the object is removed from the list of known objects. As soon as the clustertracking detects a new object which can not be assigned a previously known ID, the position is matched with all objects in the prediction phase. If the new object is close to the predicted position, the cluster is assigned to the object and the correction step of the EKF is performed.

*threshold threshold*

The simulation environment being used is VREP<sup>1</sup>. VREP has been developed for various robotic applications. VREP allows you to construct any multi-body simulation (MKS) based on various physics engines within a graphic editor. Furthermore, VREP already has many ready-to-use sensor models, such as the Velodyne VLP-16. The whole simulation can communicate via a RemoteAPI interface with nearly every programming language.

Figure 7.1: VREP



## 7.1 Simuation Scenario

referenzen zu "roundabout in law" mit formelsymbolen und so

For the simulation scenario, a simple small roundabout with an outside diameter of 26m and a miniature circuit was designed, as this is the most interesting object due to its size in combination with the VLP-16, which still has an acceptable resolution in this area. And in urban areas due to their size, they are frequent. To test the object detection, the small roundabout offers good possibilities to test the tracking due to its buildable roundabout island. Furthermore, due to its size, it is possible to survey the entire roundabout. The whole sce-

bebaubare mittelinsel in smal roundabout er-  
genzen (im ersten Kapitel)

1. <http://www.coppeliarobotics.com/>

nario was down-scaled by a factor of 10 due to limitations in Vrep.

Inside the scenario there is a bicycle, a pedestrian and two other vehicles driving around the roundabout. The objects are moving on fixed paths. The speed of all traffic is adapted to the type. The pedestrian moves with the 5km/h usually assumed for pedestrians. The bike with 15km/h. The two cars move at different speeds between 25 and 35 km/h. In order to avoid a collision of the vehicles, both vehicles are equipped with distance sensors at the front. If this sensor detects a vehicle, the vehicle enters an Adaptive Cruise Control mode and adapts to the front vehicle. The Adaptive Cruise Control is implemented as a simple proportional controller.

The autonomous vehicle also moves along a fixed path, which leads from the right into the roundabout and leaves the roundabout at the third exit. The task of the vehicle is to safely enter the roundabout and safely leave the roundabout. In doing so, the vehicle must pay attention both to the pedestrian, the cyclist and the other vehicles within the roundabout. For this, the vehicle is equipped with a virtual Velodyne VLP-16.

---

## 7.2 Simulation Logic

In order to carry out the previously described scenario, a logic had to be developed, which includes not only the statemachine for passing through the roundabout, but also the localization of the vehicle in the scenario as well as the connection of all sensors to the object detection. Since all subsequent calculations require little resources, the corresponding software was developed in Python.

### 7.2.1 Sensor Connection

The object detection requires the data of the Velodyne VLP-16 and the data of the Applanix POS-LV as sensor input. However, only the current position in the WGS84 format and the current heading are required by the object detection. Since the Applanix sensor can not be easily reconstructed in VREP, the necessary data is simply generated from the position and rotation data readable in Vrep. Since the position in VREP is specified in Cartesian coordinates, they are transformed via the transformation contained in OpenDAVINCI, using a reference coordinate in WGS84 format.

The connection of the Velodyne VLP-16 is more difficult because the structure of the measured data differs significantly from the original Velodyne. Although the data is also output in polar coordinates, only measured values are output when they hit an object. Since a large part of the scenario is empty space, the measurement data have many holes on them. But the object detection needs continuous measurements, so they must be transformed into a suitable form.

It is important to note that object detection requires a complete 360 degree measurement. To do this, you must wait until all necessary measurement data are available. The virtual Velodyne runs a measuring rate of 10Hz, and is divided into 4 segments, which are read one after the other. The time step used for the simulation is 50 ms, so that we can read out 2 segments in one step. Once all data are collected, they are assembled into a suitable measuring frame.

The measured data are provided as a list of spherical coordinates (radius  $r$ ,

polar angle  $\theta$ , azimuth angle  $\Phi$ ). Since the azimuth and polar angles are not at equidistant intervals but have a much higher resolution than the original sensor, the measured values are rounded down to the original resolution of 0.2 and 2 degrees and the measured values are entered into a corresponding two-dimensional fixed-size array. Thus, areas that are not detected have the value zero.

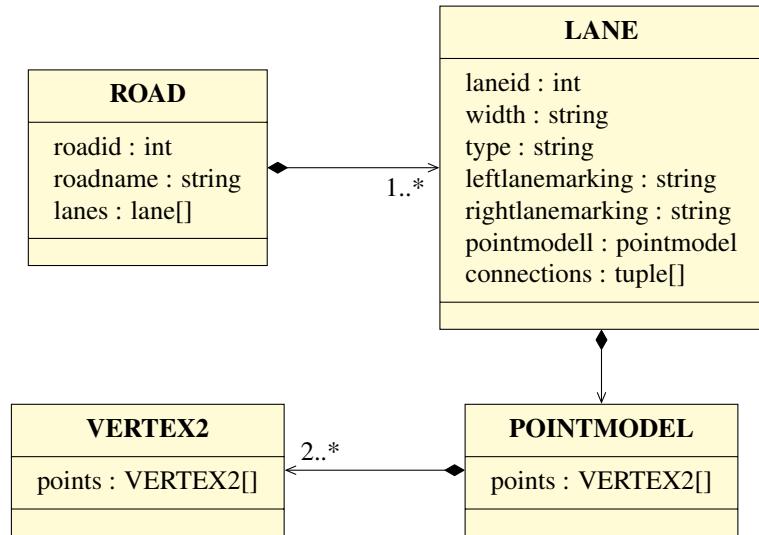
After all necessary measurement data have been collected and converted, they are sent to the object detection via a specially developed OpenDAVINCI-Python interface. Unnecessary data for the Applanix is simply set to zero.

### 7.2.2 Mapping

The OpenDAVINCI internal Compressed Scenario Data Format (SCNX) [2] is used for mapping. This allows scenarios to be defined by describing and combining stationary and dynamic elements to formulate different traffic situations. The SCNX format offers, among other things, the classes ROAD, LANE for the modeling of roads. One ROAD can consist of several LANEs. A lane consists of a set of points which describe the course of the lane. The lane can be assigned with the attributes lane mark and width. Individual lanes can be connected to each other. What makes it possible to build a complicated road network with these simple models. Since OpenDAVINCI does not provide a Python interface for parsing the Scenariofiles, a special Lexer was implemented for the processing of the Scenariofiles. At this time, only the classes required for the simulation are implemented. Therefore, a lane can only be described by a point model, while OpenDAVINCI offers additional models.

Furthermore, the model included in OpenDAVINCI had to be extended, since OpenDAVINCI unfortunately does not support other types of paths. Therefore, the class lane has been extended by one type attribute, which makes it possible to declare this as a bicycle or footpath. The structure of the extended Python classes can be examined in fig. 7.2.

Figure 7.2: Parser Objects



For the handling of the roundabout, however, the map format had to be further extended. According to RAST [25], a roundabout should be as circular as possible, for the sake of simplicity, we assume that the roundabout in simulation is a perfect circle. This means that both roads, cycle paths and footpaths

are perfectly circular and can be described by an inner and outer radius. For this reason, a class ROUNDABOUT [fig. 7.3] was added to the description of the roundabout, which contains the center point of the roundabout, references to all lanes, and inner and outer radii. Furthermore, the connections to the connecting lanes are defined therein.

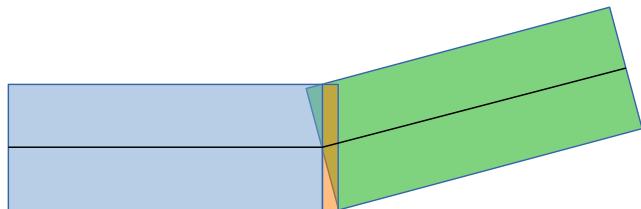
Figure 7.3: Roundabout Class

ROUNDABOUT
roundaboutid : int lanes : lane[] junctions : tuple[] inner_lane_radius : float[] outer_lane_radius : float[] centrum : VERTEX2

## Localization

Da wir uns innerhalb der Simulation bewegen, können wir für die Lokalisierung eine perfekte Position annehmen. Lokalisierung in Rahmen der Karte bedeutet, dass wir die aktuelle bekannte Position auf ein Lanesegment mappen wollen. Da die Lanes nur in Form von Punkten und Breite gegeben sind, werden für jeden Laneabschnitt Rechtecke berechnet und für jedes Rechteck geprüft, ob sich die aktuelle Position innerhalb des Rechtecks befindet. Dies wird gemacht, indem das Rechteck (und die Position) in eine Achsen aligned Position gedreht und in den Koordinatenursprung verschoben wird. Weiterhin muss das Rechteck vergrößert werden, sofern wir nicht das letzte Lanesegment bearbeiten, da sonst eine Lücke zwischen den Rechtecken verbleiben würde. Dazu wird der Winkel zum nächsten Linienabschnitt genutzt. Das Ganze ist in fig. 7.4 symbolisiert. Dabei wird das blaue Rechteck das aktuell zu bearbeitende darstellen, das grüne Rechteck das nachfolgende und das orangefarbene die nötige Verlängerung.

Figure 7.4: Mapping Rectangles



Wie in fig. 7.4 zu sehen, stehen die einzelnen Segmente üblicherweise nicht in einem 180 Grad Winkel zueinander. Daher können sich die Rechtecke überschneiden und für eine Position können mehrere Rechtecke zurückgeliefert werden. In diesem Fall wird die Bewegungsrichtung des Objektes herangezogen und die Aussicht der Bewegungsrichtung neuste Segment ausgewählt. Sollten sich mehrere Lanes überschneiden, z.B. Radwege welche die Kreisverkehr einfahrt kreuzen, wird das richtige Segment anhand des Objekttyps ausgewählt, also: auto -> Straße oder Radfahrer -> Radweg. Weiterhin wird die X-Koordinate des Objektes im gedrehten und verschobenen Koordinatensystem zurückgeliefert, was für die Distanzberechnungen von Nötien ist. Dabei nehmen wir immer an, dass sich das Objekt mittig auf der Spur befindet. Alternativ bietet sich für das Problem auch der "Ray casting algorithm" [11] an. Da wir für alle Distanzberechnungen annehmen, dass sich das Objekt mittig auf der Spur

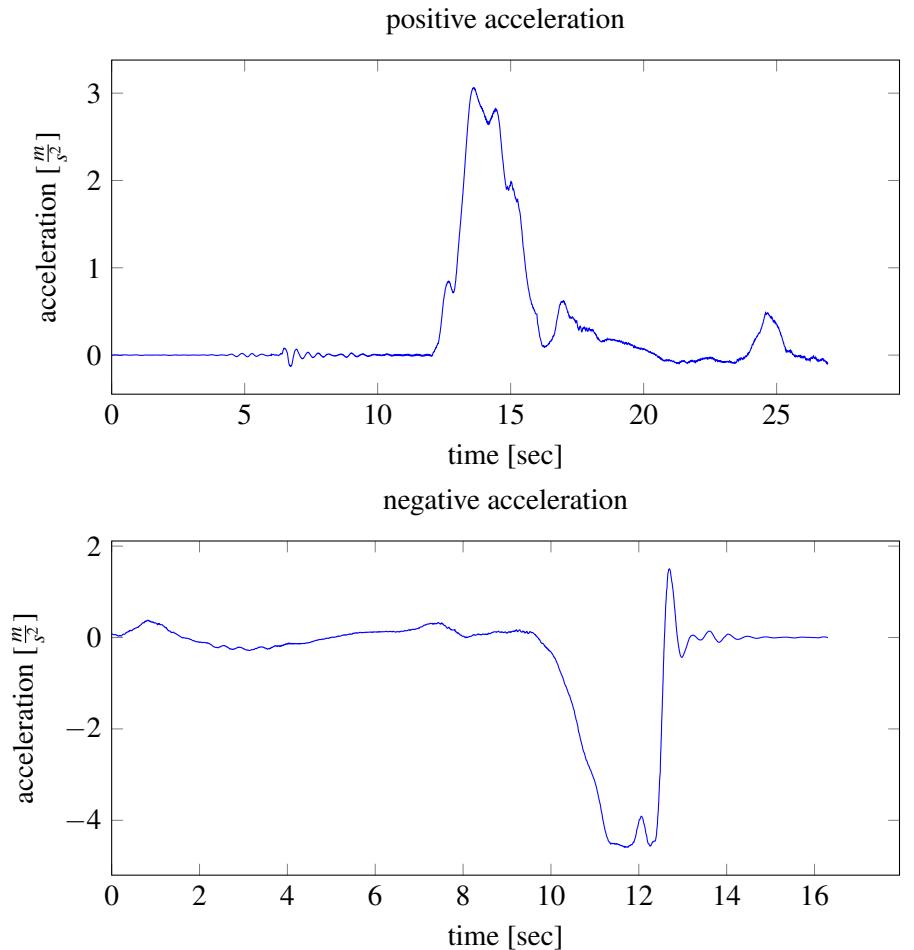
befindet, ergibt sich aus dieser Berechnung jedoch nur ein sehr kleiner Fehler und die Entfernung ergibt sich praktischer weise aus den Vorherigen Berechnungen, sodass dieser nicht umgesetzt wurde.

### 7.2.3 State machine

simulation mit synthetischen daten (aus vrep) und daten aus object tracking vergleiche  
Annahme constant acceleration model  
nummer anpassen

The statemachine is implemented according to Gang of Four [16]. For this, X States were implemented. A constant acceleration model is assumed for all vehicle computations. In order to determine the maximum positive and negative acceleration, test runs with the real vehicle were recorded and the maximum subjectively pleasant values being determined. Die genutzten Aufnahmen sind in fig. 7.5 zu sehen, dabei wurde das Fahrzeug von null auf 25 km/h beschleunigt bzw. von 25 auf null km/h gebremst. Es wurden jeweils die maximalen Werte abgelesen, so dass wir für die maximale positive Beschleunigung  $3m/s^2$  und für die maximale negativen Beschleunigung  $-4.5m/s^2$  annehmen.

Figure 7.5: Acceleration Plot



plot acceleration positiv/negativ, und auswertung  
bild mit intersection position

Before each call of a state, all objects delivered by the object detection are picked up and mapped to the lanes held in the map. Objects that were not on a track or path are rejected. Furthermore, the expected intersection points of your own vehicle with all paths in the roundabout is calculated. This is necessary because the vehicle is not driving straight into the roundabout and the position in the Map can be slightly corrupted. For this purpose, the intersection points of trajectory and lanes are calculated from the expected vehicle trajectory and

the simplified roundabout model.

**Start - State** The vehicle is in the start state until it reaches a distance of 20 m to the outside lane of the roundabout, afterwards the state is changed to the ToRoundabout state.

**ToRoundabout - State** We are in this State, if the vehicle is not yet located in the roundabout. At each call, the remaining distance, of all obstacles which are moving in the direction of the intersection, to the intersection point is calculated. Based on this distances and the object speeds the estimated time window is calculated, in which the obstacles reaches the intersection point. A time window is also calculated from the maximum acceleration and speed to define max speed of our vehicle and the calculated distance to the intersection point. If this time window is smaller than one of the obstacles plus a safety interval of two seconds, the vehicle enters the roundabout and changes to the InRoundabout state. Therefore, the necessary acceleration is reduced to such an value that the condition is still fulfilled. If this condition does not apply, the vehicle enters the Brake state.

**Brake - State** In this state, the distance of the our vehicle to the next intersection point is calculated. With this distance, the necessary Negative acceleration is now calculated in order to get the vehicle standing up to this point, and sent to the vehicle. After the vehicle has been brought to a standstill, we changes back into the ToRoundabout state.

**InRoundabout - State** When the vehicle is inside the roundabout, it carries out ac ACC. That is, the vehicle travels at a constant speed until it encounters an obstacle. As soon as an obstacle is reached, the speed is reduced proportionally to the distance of the obstacle, so that a safety distance is established. In addition, the time window is calculated for each call which the vehicle needs to reach the target exit.

If the vehicle feeds the target exit at 10m, the state will be changed to ExitRoundabout.

### ExitRoundabout - State

---

# 8

## Evaluation

---

### 8.1 Simulation

---

### 8.2 Real Measurements

---

# 9

## Conclusions

- Kann in mehrere Unterkapitel gegliedert werden
- Greift Thesen oder Fragestellungen aus der Einleitung wieder auf
- Fasst die Arbeit knapp und prägnant zusammen
- Ordnet die Ergebnisse in Gesamtzusammenhänge ein
- Zieht Schlussfolgerungen aus den erarbeiteten Ergebnissen
- Kann auch eigene Bewertungen oder Meinungen enthalten
- Gibt eine Ausblick auf mögliche Konsequenzen oder notwendige weitere zu lösende Probleme

---

# 10

## Future Work

---

# Bibliography

- [1] A Atreya et al. “DARPA Urban Challenge – Virginia Tech Technical Paper”. In: (2007).
- [2] Christian Berger. “Automating Acceptance Tests for Sensor- and Actuator-based Systems on the Example of Autonomous Vehicles”. PhD thesis. 2010, p. 298. ISBN: 9783832293789.  
URL: <http://www.christianberger.net/Ber10.pdf>.
- [3] Lothar Bondzio. *Verkehrssicherheit innerörtlicher Kreisverkehre*. Techn. Ber. Berlin: Gesamtverband der Deutschen Versicherungswirtschaft e. V., 2012.
- [4] Manfred Broy. “Challenges in automotive software engineering”. In: *Proceeding of the 28th international conference on Software engineering - ICSE '06*. New York, New York, USA: ACM Press, 2006, p. 33. ISBN: 1595933751.  
DOI: 10.1145/1134285.1134292.  
URL: <http://portal.acm.org/citation.cfm?doid=1134285.1134292>.
- [5] Martin Buehler, Karl Iagnemma, and Sanjiv Singh. “The DARPA Urban Challenge”. In: *The DARPA Urban Challenge*. Vol. 49. 2010, p. 626. ISBN: 3642039901.  
DOI: 10.1007/978-3-642-03991-1.  
arXiv: arXiv:1011.1669v3.
- [6] M Campbell and E Garcia. “Team Cornell: technical review of the DARPA urban challenge vehicle”. In: *DARPA Urban Chall. ...* (2007), pp. 1–25.  
URL: [http://archive.darpa.mil/grandchallenge/TechPapers/Team%7B%5C\\_%7DCornell.pdf](http://archive.darpa.mil/grandchallenge/TechPapers/Team%7B%5C_%7DCornell.pdf).
- [7] Oshkosh Truck Corp. “Team Terramax Darpa Grand Challenge 2005”. In: (2005), pp. 1–14.
- [8] Applanix Corporation. *POSLV SPECIFICATIONS*. 2015.
- [9] Martin Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise”. In: AAAI Press, 1996, pp. 226–231.

- [10] Martin a Fischler and Robert C Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Communications of the ACM* 24.6 (1981), pp. 381–395. ISSN: 0001-0782.  
DOI: 10.1145/358669.358692.  
URL: <http://dx.doi.org/10.1145/358669.358692>.
- [11] Michael Galetzka and Patrick Glauner. “A Simple and Correct Even-Odd Algorithm for the Point-in-Polygon Problem for Complex Polygons”. In: *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, 2017, pp. 175–178. ISBN: 978-989-758-224-0.  
DOI: 10.5220/0006040801750178.  
URL: <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006040801750178>.
- [12] M Himmelsbach, T Luettel, and H.-J. Wuensche. “LIDAR-based 3D Object Perception”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2009), pp. 994–1000.  
DOI: 10.1109/IROS.2009.5354493.  
URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5354493>.
- [13] M Holmes, a Gray, and C Isbell. “Fast SVD for large-scale matrices”. In: *Workshop on Efficient Machine Learning* 1.1 (2007), pp. 2–3.
- [14] Alonzo Kelly. *A 3D state space formulation of a navigation Kalman filter for autonomous vehicles*. Pittsburgh: Carnegie Mellon University, 1994, p. 95.
- [15] James F. Kurose and Keith W. Ross. *Computer Networking A Top-Down Approach*. 5. 2013, p. 4. ISBN: 9780132856201.  
DOI: 10.1017/CBO9781107415324.004.  
arXiv: 8177588788.
- [16] P. Lester. *Gang of Four*. Omnibus, 2008. ISBN: 9781847722454.  
URL: <https://books.google.de/books?id=9Ap0M5GTpqOC>.
- [17] Michael Montemerlo et al. “Junior: The stanford entry in the urban challenge”. In: *Springer Tracts in Advanced Robotics* 56.October 2005 (2009), pp. 91–123. ISSN: 16107438.  
DOI: 10.1007/978-3-642-03991-1\_3.  
arXiv: 10.1.1.91.5767.
- [18] F. Moosmann, O. Pink, and C. Stiller. “Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion”. In: *2009 IEEE Intelligent Vehicles Symposium*. June 2009, pp. 215–220.  
DOI: 10.1109/IVS.2009.5164280.
- [19] Abdul Nurunnabi, David Belton, and Geoff West. “Diagnostic-Robust Statistical Analysis for Local Surface Fitting in 3D Point Cloud Data”. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* I-3.September (July 2012), pp. 269–274. ISSN: 2194-9050.  
DOI: 10.5194/isprsannals-I-3-269-2012.  
URL: <http://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/I-3/269/2012/isprsannals-I>

- 3-269-2012.pdf%20http://www.isprs-ann-photogra  
mm-remote-sens-spatial-inf-sci.net/I-3/269/2012  
/.
- [20] Meghashyam Panyam mohan Ram. “Least Squares Fitting of Analytic Primitives on a GPU”. PhD thesis. Clemson University, 2007, p. 103.  
URL: <http://tigerprints.clemson.edu/all%7B%5C%7Dtheses>.
- [21] Robin Schubert, Eric Richter, and Gerd Wanielik. “Comparison and evaluation of advanced motion models for vehicle tracking”. In: *Information Fusion, 2008 11th International Conference on* 1 (2008), pp. 1–6.  
DOI: 10.1109/ICIF.2008.4632283.  
URL: <http://ieeexplore.ieee.org/xpls/abs%7B%5C%7Dall.jsp?arnumber=4632283>.
- [22] Inge Söderkvist. *Using SVD for some fitting problems*. Tech. rep. 2. Sweden: Lulea University of Technology, 2009, pp. 2–5.  
URL: <https://www.ltu.se/cms%7B%5C%7Dfs/1.51590!/svd-fitting.pdf>.
- [23] Forschungsgesellschaft für Straßen- und Verkehrswesen. Arbeitsgruppe Straßenentwurf. *Merkblatt für die Anlage von Kreisverkehren*. 2006.
- [24] Forschungsgesellschaft für Straßen- und Verkehrswesen. Arbeitsgruppe Straßenentwurf. *Richtlinien für die Anlage von Landstrassen (RAL)*. FGSV (Series). FGSV-Verlag, 2013.
- [25] Forschungsgesellschaft für Straßen- und Verkehrswesen. Arbeitsgruppe Straßenentwurf. *Richtlinien für die Anlage von Stadtstraßen: RAST 06*. FGSV (Series). FGSV-Verlag, 2007. ISBN: 9783939715214.
- [26] Chris Urmson et al. “Tartan Racing: A Multi-Modal Approach to the DARPA Urban Challenge”. In: *Defense* 94.4 (2007), pp. 386–387. ISSN: 15564967.  
DOI: 10.1002/rob.20251.  
URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.8722%7B%5C%7Drep=rep1%7B%5C%7Dtype=pdf>.
- [27] Inc. Velodyne Acoustics. *Velodyne LiDAR Puck (VLP-16) Manual*. 2015.
- [28] Wikipedia. *Linear least squares (mathematics)*. 2017.  
URL: [https://en.wikipedia.org/wiki/Linear\\_least\\_squares\\_\(mathematics\)](https://en.wikipedia.org/wiki/Linear_least_squares_(mathematics)) (visited on [visited on 03/06/2017]).
- [29] Liang Zhang et al. “Multiple Vehicle-like Target Tracking Based on the Velodyne LiDAR\*”. In: *IFAC Proceedings Volumes* 46.10 (June 2013), pp. 126–131. ISSN: 14746670.  
DOI: 10.3182/20130626-3-AU-2035.00058.  
URL: <http://linkinghub.elsevier.com/retrieve/pii/S1474667015349211>.