Universitetet
i Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study program/specialization: Information Technology – Automation and Signal Processing	Spring semester, 2017 Open / ~~Confidential~~
Author: Elisabeth Victoria Borg Baardseth	[illegible] .... (signature author)
Instructor: Kim Mathiassen Supervisors: Prof. Karl Skretting, Kim Mathiassen	
Title of Master's Thesis: Vehicle Detection and Pose Estimation in Autonomous Convoys Norwegian Title: Bildeteksjon og Estimering av Kjøretøyets Posisjon og Rotasjon i Autonome Konvoyer	
ECTS: 30	
Subject headings: Platoon, vehicle detection, pose estimation, POSIT, UGV, LiDAR,	Pages: 61 + attachments/other: 6 Stavanger, June 15 2017

Abstract

Autonomous convoys, also known as platooning, are defined as a group of vehicles driving after one another autonomously. Acting as one single unit the gap between them can be significant reduced and hence the fuel costs compared to conventional driving.

In this thesis a semi-automatic two-vehicle military convoy is studied. Assume the first vehicle remain in control by humans. The second vehicle should then follow it's track autonomously based on information about the relative distance, position and velocity of the first vehicle. The purpose of this thesis is to find and test methods for vehicle detection and pose estimation.

The methods are mainly based on 3D point cloud data gathered by a LiDAR, but information from cameras are also used. The LiDAR is chosen because of it's robustness in shifting light and weather conditions, but also because most of today's research on the field is based on camera vision.

Classifiers based on the leading vehicle's geometrical properties was found suitable. Also the POSIT method, which combines the imaging coordinates with the corresponding 3D properties provides good results. Distance error was measured to around 15% and orientation deviation to $\pm 4°$. For driving that not require millimetre precision the conclusion it that the methods used are well suited. They also have room for improvements as there was several sources of uncertainty in this project.

Acknowledgement

After almost six months of intense studying filled with joy and frustration through ups and downs, writing this note of thanks as the finishing touch on my dissertation feels like a big relief. I have learned so much, not only scientifically, but also on a personal level.

First I need to thank Prof. Karl Skretting, UiS, and Kim Mathiassen, FFI, for supervising me, answering my questions and always have an open mind to discuss my ideas and thoughts. Also a big thank you to the rest of you at FFI, especially Magnus Baksaas and Marius Thoresen, for an exciting and challenging problem statement and for spending your time helping me with my thesis project.

Then I would like to thank my friends and co-students. Not only for your support and valuable deliberations over our problems and findings, but also for taking your time to discuss and laugh about other things than just our papers.

And last, but not least, a big thank you to my fiance for your patient and support during this final stage of my studies.

Thank you very much, everyone!

Elisabeth V. Borg Baardseth
Stavanger, June 15th, 2017

Contents

List of Figures

List of Tables

List of Abbreviations

AHS	Automatic Highway System
ATV	All Terrain Vehicle
FOV	Field of View
GPS	Global Positioning System
HOG	Histogram of Oriented Gradients
IMU	Intertial Measurement Unit
LBP	Local Binary Pattern
LiDAR	Light Detection and Ranging
M-ELROB	Military European Land Robot Trial
POS	Pose from Orthography and Scaling
POSIT	POS with Iterations
RANSAC	Random Sample Consensus
ROS	Robot Operating System
SVM	Support Vector Machine
UGV	Unmanned Ground Vehicle

1. Introduction

1.1 Motivation

Platooning is defined as a group of vehicles driving autonomously in a convoy. The platoon can be considered as one unit where every vehicle follows the vehicle in front of it with fixed spacing. The idea of platooning is inspired by nature where groups of animals organise themselves into energy efficient formations take advantage of aero- and hydrodynamics. Well known examples are like migrating bird swarms and packs of dolphins in the sea [10].

By reducing the spacing between the vehicles the drag coefficient can be dramatically reduced and hence the fuel costs too [10]. As humans do not have the reaction time needed to maintain the safety factor at such short margins, fuel cost effective and safe platoons are realisable by means of autonomous vehicles only. Other benefits of platooning are the increase in road capacity. Also the traffic safety in general may be improved as accidents related to human errors and inattention can be neglected.

Figure 1.1: *Truck platoon test drive [1]*

In a military perspective platoons are of great interest. Not only based on the arguments mentioned above, but as a tool to reduce the risk of severe injuries or death on soldiers. Transportation of supplies and personnel, surveying, aid missions etc. are typical military tasks which often take place in risky environments. Also the transportation can become more time and cost efficient compared to convectional convoy driving.

Fully autonomous driving in rural off road environments requires a sophisticated and robust system. The vehicle system need to interpret the environment to avoid obstacles and dead ends, and adjust the speed according to the current ground conditions. Typical issues are related to differentiate between lakes and solid ground, evaluate ground friction (dry/icy) and estimate steepness. To reduce the complexity this thesis set focus on the following routine only. Assumed humans remain in control of the first vehicle of the convoy, e.g. by physically driving it or by remote control, the second vehicle and backwards "only" have to follow the leaders track perfectly and the mentioned issues can be neglected.

1.2 Thesis' background

This master thesis project is done in collaboration with the Norwegian Defence Research Establishment and the University of Stavanger. The thesis is part of an ongoing research project on autonomous vehicles and military off road convoys.

The Norwegian Defence Research Establishment, FFI[1], is the prime responsible institution for defence related research in Norway. It's principal mission is to develop state-of-the-art technical solutions and perform research that meets the requirements to the Norwegian Armed Forces. The agency was founded in April 1946 and is subordinated the Norwegian Ministry of Defence. [11].

1.3 Problem statement

The leading vehicle in a two-vehicle platoon is to be followed autonomously. The task of this thesis is to create and test methods to estimate the relative distance between the two vehicles and the relative position, orientation and velocity of the leading vehicle only. The estimates should be based on measurements gathered by the following vehicle. Sensors available are LiDAR, colour and gray scale cameras, stereo vision, odometer data and GPS position.

No internal communication within the platoon is allowed due to safety issues. Any challenges faced by the leading vehicle itself, e.g. path selection due to rough terrain, traffic, obstacles etc. are not to be considered.

Based on challenges found in previous studies using stereo vision for position estimation, see chapter 2.1, this thesis focuses on estimations based on LiDAR 3D point cloud data.

1.4 Assumptions and limitations

The convoy studied in this paper consists of two vehicles: the leading vehicle and the one following it, the *ego-vehicle*. The main approach is driving in rural environments, on and off roads. Issues related to obstacles, route planning etc. are neglected and it's assumed that the leading vehicle drives a route that is safe to follow. Due to safety reasons the methods should be based on information collected by the ego-vehicle only, that means no internal communication within the convoy.

Processing time should not be considered as an issue, see chapter 4.1.4. Duration and speed performance of the methods are therefore not considered in this thesis.

Calculations, method adjustments and results presented in this thesis are based on one

[1]Forsvarets forskningsinstitutt

data set recorded in February 2017. Details on the gathering procedure, duration and more are described in chapter 4.2.1.

1.5 Thesis outline

Chapter 1 - Introduction
The reader is introduced to the background and motivation of the thesis topic. Also the problem statements is defined, and assumptions and limitation described.

Chapter 2 - Background
A brief presentation of previous work on the field of platooning, both on and off road. Theory relevant for this thesis is also presented.

Chapter 3 - Implementation
Detailed description of the proposed methods and it's purpose.

Chapter 4 - Experiments
Vehicle and sensor configuration, the collection process of data used and calibration methods are described in detail. Experiments arranged to verify the performance of methods created are then presented.

Chapter 5 - Results
This chapter contains all results obtained from the experiments described in chapter 4.

Chapter 6 - Discussion
Solutions and results obtained throughout the thesis are discussed.

Chapter 7 - Conclusion and Future work A conclusion is set, and future work and challenges are discussed.

Appendix A
Sketches with measurements and dimensions of the ego-vehicle used.

Appendix B
Sketches with measurements and dimensions of the leading vehicle used.

Appendix C
Matlab source code is attached.

2. Background

This chapter presents background information this thesis is based upon. The first section gives an overview on previous work and studies focusing on autonomous driving and platooning. The main focus in this summary will be on platooning itself, not necessarily issues related to autonomous driving such as path selection and obstacle avoidance. Both on road and off road aspects are included.

2.1 Previous work

The first part of the thesis project was a literature study to gain knowledge on previous and ongoing projects of relevance. In this section the findings are presented.

History

The concept of platooning was first presented by General Motors at the World Fair in 1939. This resulted in several studies which led to the Automatic Highway System (AHS) as presented in San Diego in 1997 [10]. The research started in the 1960 and the first tests took place in the 1970's.

Initially the systems were build on a single free agent, i.e. a single vehicle platoon. This developed to include more vehicles and inter-platoon communication was introduced. The result was increased road safety and the fixed spacing between could be reduced. A study from 1995 [12] found that a reduction in spacing could cause a 55% reduction of the drag coefficient and hence the fuel costs.

Later a concept of infrastructure assisted platooning was released, which implemented communication between the highway and the platoon. Active or passive components, such as magnetic plates and emitting units, was suggested integrated in the road infrastructure. This could provide the platoon with information of interest, for example velocity, position in the lane, internal distance estimates in the platoon, road exits and entries, speed limits etc. [10] [13].

Due to disadvantages related with infrastructure modifications, like big investments and that the platoon need to be compatible with the current road system, has directed the research over to independent platoons again. The most resent research programme is SARTRE (Safe Road Trains For Environment), an EU founded study launched in 2009. They achieved their first milestone in 2011 demonstrating a truck following a leading truck autonomously. The leading truck however was not autonomic.

In April 2016 another milestone was reached. This time by an initiative between the European automobile industry named *EU Truck platooning challenge*. For the first time six on-road truck platoons successfully arrived Rotterdam after driving autonomously more than 20.000 km in total through Sweden, Denmark, Germany, Belgium and the Netherlands. Fuel costs was estimated reduced by 10%. The project is led by Scania, Daimler, MAN, IVECO, DAF and Volvo Group [14] [15].

Another aspect of platooning is off road platoons. This is more of interest for military departments, space agencies etc. Off road autonomous driving is associated with more complex interpretations compared to on road driving. Varying ground conditions, no roads to follow and route planning that avoids dead ends are some examples [16].

A leading research team on the field is located at the Universität der Bundeswehr (UniBW) in Munich, Germany. With more than 30 years of research, named the MuCar-project, they have successfully developed an autonomous two-vehicle platoon. As participants in M-ELROB[1] 2013 they managed to drive 99.76% of the route fully autonomously. Future challenges are related to improve the following accuracy, increase of speed and robustness and route planning [17] [18].

Technology

Several methods for vehicle detection have been submitted by researches through the years. Frequently used is either visual recognition using camera(s) or detection in 3D point clouds obtained by surveying sensors like LiDAR and radar. Both approaches has it's pros and cons. For example is visual recognition excellent for object detection and recognition providing information about texture, colour, shadows, shape and other unique features using neural networks and classifier methods like LPB, SVM, adaBOOST etc. [19]. However it's very light sensitive and the results are highly affected by weather and illumination conditions. 3D data on the other hand is not affected by illumination and provides the same information during both daylight and night. Also a LiDAR provides range data with high accuracy. But with classifiers based on geometrical properties only this approach has a association issue [20]. Recently studies therefore look at the possibilities of merging camera data with LiDAR to take advantage of both aspects.

2.2 POS/POSIT

The leading vehicle's orientation and position relative the ego-vehicle is to be estimated. Assuming a 3D model of the object is known, i.e. the relative geometry of a set of feature points, the translation and rotation matrices can be approximated by means of the POS/POSIT method from a single image.

The methods POS, *Pose from Orthography and Scaling*, and POSIT, *POS with Iterations*, was first presented by DeMenthon et al. in [21]. The methods require minimum four known pairs of 3D feature point coordinates and the corresponding 2D image coordinates.

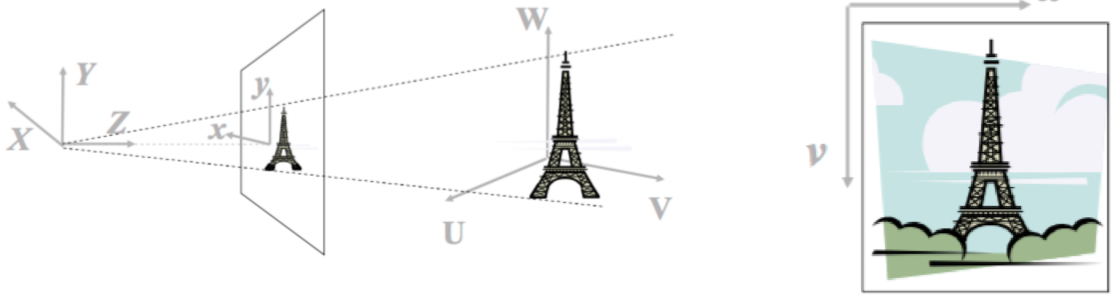
[1]Military European Land Robot Trial

Based on perspective projection the POS method generates a linear equation system based on the given feature point coordinate pairs. When solved an approximate of the current rotation and translation matrices of that object in relation to the camera position is estimated. [21] [22]

POSIT is an extended version of POS. By implementing POS in an iteration loop the results can be used to estimate an even more accurate approximation. The number of iteration can be specified according to available processing time. POSIT converges after only a few iterations, see section 16 in [21].

2.3 Imaging geometry

Imaging geometry and perspective projection is used to create a conversion method from LiDAR 3D-coordinates to camera image 2D-coordinates. For this purpose the theory of *weak forward projection* is presented. Forward projection is the process of converting 3D



(a) (b)

Figure 2.1: *The figure illustrates forward projection, i.e. how 3D world coordinates are transformed to 2D image coordinates [2]*

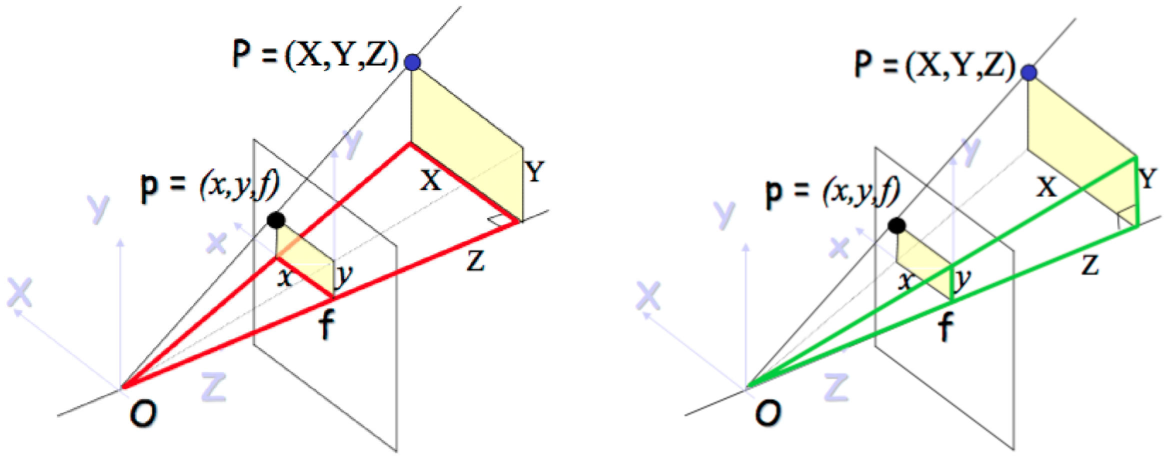
world coordinates into 2D image pixel coordinates. See figure 2.1. By weak projection a linear approximation is used instead of the full projection. The approximation is based on the similar triangles rules, see figure 2.2 and 2.3.

For a world point, $P = (X, Y, Z)$, the image point, $p = (x, y, f)$, this gives the following approximated:

$$x = f \cdot \frac{X}{Z} \tag{2.1}$$

$$y = f \cdot \frac{Y}{Z} \tag{2.2}$$

f is the focal length of the camera.



(a) (b)

Figure 2.2: *Geometrical properties of weak perspective projection. Z-axis is optical axis in both figures.* ***(a)*** *x-plane properties highlighted in red.* ***(b)*** *y-plane properties highlighted in red. [2]*

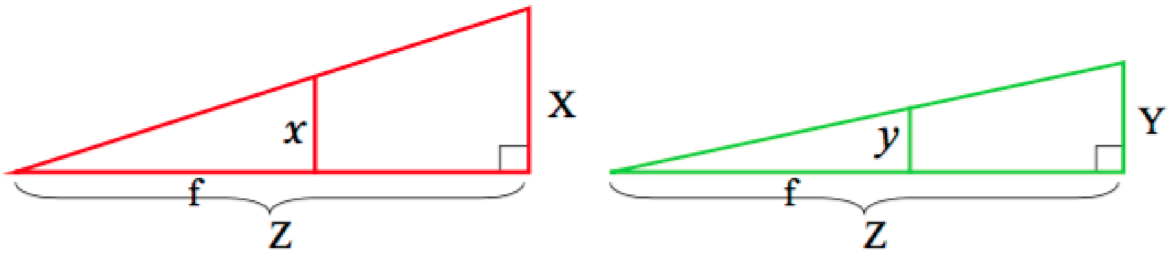


Figure 2.3: *Illustration of the similar triangles for the problem given in fig. 2.2 [2].*

2.4 RANSAC

Based on a vehicle's geometrical properties, see chapter 3.1.1, it's of interest to find surfaces with a specific angle around the ground plane. Plane matching and plane estimation from a set of LiDAR data points using RANSAC serves this purpose.

The Random Sample Consensus, shortened RANSAC, is an iterative method estimating the parameters of a mathematical model based on a given data set with outliers. In this context *inliers* referrers to the data points whose distribution can be described by the given mathematical model, though may be affected by noise, and *outliers* to the remaining data points that does not fit this model. RANSAC can successfully be applied to data sets with up to 50% outliers.

Based on a voting routine and the least square method RANSAC estimates the model parameters that has the most inliers. The outliers does not affect the results. For a given data set only one solution is available [3] [23].

In figure 2.4 an example of using RANSAC for line estimation is displayed. Outliers are marked in red and inliers in blue.

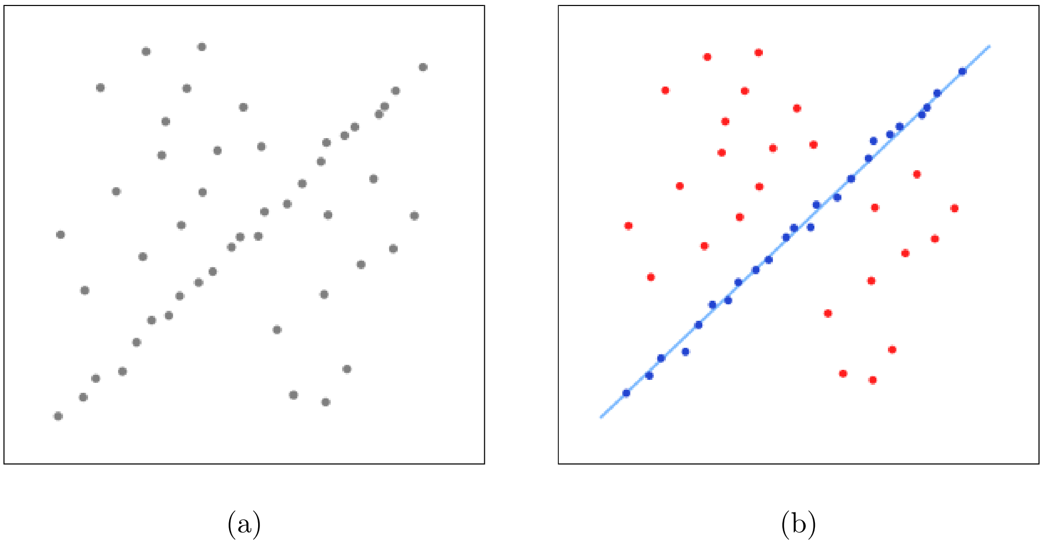


Figure 2.4: *Example on how RANSAC processes the data set (a) and returns the estimated line with inliers (blue) and outliers (red). The outliers does not affect the line parameters. [3]*

3. Methods

This chapter presents descriptions of methods developed as a solution to the problem statement. Some of them obtain the same information even though only one of them are to be implemented in the final system. In this way their results can be compared and evaluated and the method with the best performance selected.

The methods are implemented in Matlab for testing, and therefore make use of build-in Matlab functions where available. Their role and functionality is still described in this chapter.

For some methods it's assumed that the leading vehicle is known, e.g. it's geometrical properties etc. This should come out clear during the method description.

All axis and angle references are consistent. The reference coordinate system is right-handed with the x-axis pointing in the driving direction, the y-axis to the left and the z-axis upwards. If not specified, angles, distances etc. are all describing the leading vehicle in relation to the ego-vehicle.

3.1 Vehicle detection

This section describes the method used for vehicle detection. Before the method description an overview over the vehicle's features are presented.

3.1.1 Vehicle properties

This section will give a short description of features associated with any car, and also the vehicle to be followed, which create the foundation for how the detection methods are designed and which features they are searching for.

Visual properties

Visual features related to any object are those which are present in an image or by eye vision such as colour, shape and illumination. Based on the fact that the ego-vehicle in the convoy situation, which is the case in this paper, the vehicle to be followed will always be in front and observed from behind. The visual features of interest can therefore be limited to those on it's back and it's left and right

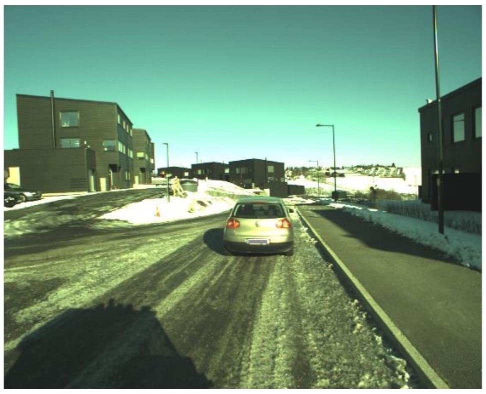


Figure 3.1: *Image captured by the roof top color camera during the test run.*

sides.

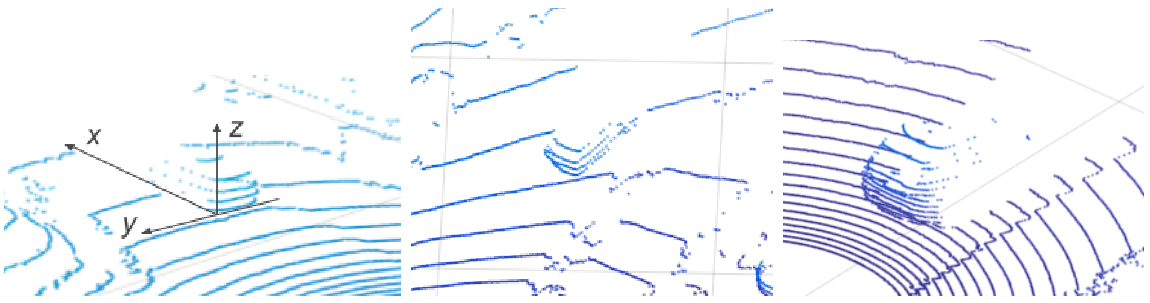
When it comes to illumination there are two main
features that can be used for vehicle detection. The first one is the difference in illumination between the car body and the windows [20]. Typically, also for the leading vehicle used, the windows reflect less light than the chassis and therefore appears darker than the body. This feature can be described using Haar-like features, see page **??**. The second one is related to the shadow underneath a vehicle. Statistically this area is darker than an unoccupied one [24].

The colour of the car body could ease the car detection problem, especially for colours that do not match the environment. In this study the leading vehicle used is silver metallic, and is therefore not a very useful colour. But, as for every normal car, the rear lights are red, which is a colour that is not usually found in the surroundings, especially in nature.

Other visual features of interest could be the license plate, which is unique for each car. If detected, and readable, it not only verifies that it's a car, but also the correct one.

Geometrical properties

Information about geometrical properties of the vehicle is helpful when it comes to detection in a 3D point cloud (as obtained from the LiDAR). Four general vehicle properties, which are applied to the detection algorithm presented in chapter 3.1.2, are described below [25]. Reference axes are displayed in figure 3.2a



(a) (b) (c)

Figure 3.2: *Screenshots of leading vehicle in the 3D point cloud. The* 90° *requirement and the other listed properties are well illustrated.*

1. **90 degree requirement** (along the z-axis)
   The surface of a vehicle's body has box like shape and the sides have an angle of 90° among the ground or road, i.e. the xy-plane in the point cloud.

2. **Smoothness**
   With some exceptions all vehicles has a smooth surface, i.e.

$$2\,f(x_1) = f(x_0) + f(x_2) \tag{3.1}$$

$$2\,f(z_1) = f(z_0) + f(z_2) \tag{3.2}$$

3. **Convexity** (xy-plane/ground plane)
   Use the fact that vehicles have a convex outline contour as given by Moosman et al. in [26]. Observed in the xy-plane the property can be described as:

$$f\left(\frac{x_0 + x_1}{2}\right) \geq \frac{f(x_0) + f(x_1)}{2} \tag{3.3}$$

4. **Negative gradients** (along the z-axis)
   From figure 3.2 it can be seen that the gradients in z-direction are either negative or zero:

$$\frac{f(z_1) - f(z_0)}{z_1 - z_0} \leq 0 \tag{3.4}$$

3.1.2 Point cloud detection

This method is a vehicle detection method in LiDAR point cloud data. It's based on the geometrical properties of a standard vehicle and the method is therefore general. However some verification statements are based on the properties of the used leading vehicle, but this can be adapted from one vehicle to another. The method consists of five steps:

(1) Normal estimation

First the point cloud data is rearranged from a $n \times 3$ matrix of spherical coordinates, $\mathbf{P}$, to a $32 \times \frac{n}{32}$ distance matrix, $\mathbf{D}$. The data is sorted with polar angles, ϕ, as rows, azimuth angles, θ as columns and radial distances, r, as cell values. See figure 3.3. The purpose is to estimate the normal of every data point based on it's neighbourhood.

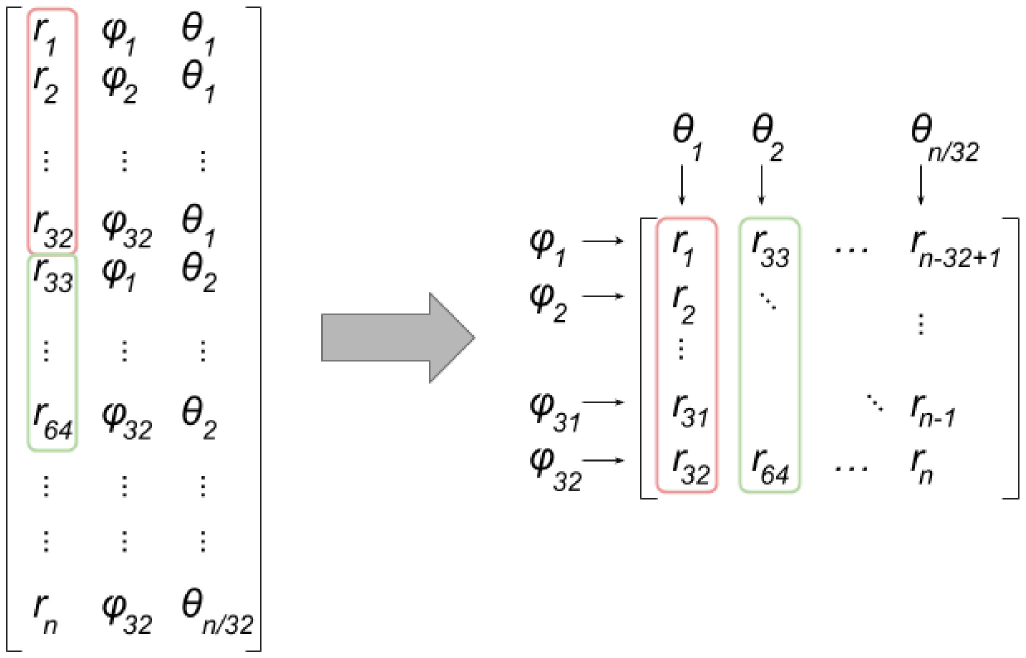


Figure 3.3: *Ilustration of rearrangement of spherical coordinates to a distance matrix.* $\boldsymbol{P}$ *to the left and* $\boldsymbol{D}$ *to the right.*

A 3×3 neighbourhood is selected. Then RANSAC is used to estimate a plane through the centre point of the mask. The plane's normal, $\vec{n}$, equals the estimated normal for that point.

To estimate the normals, $\vec{n}$, for every point in the point cloud the distance matrix, $\mathbf{D}$, is padded symmetrically. That meaning the first row is mirrored to the top, last row to the bottom, the leftmost column to the left and the rightmost to the right.

(2) Ground angle computation

With all the point normals known the angle between the ground, i.e. the xy-plane, and the object plane can be calculated. With the unit vector $\vec{e}_z = [0, 0, 1]$ representing the ground normal and $\vec{n}$ the point normal, the angle α between the two equals the angle α between the two planes:

$$\alpha = \cos^{-1}\left(\frac{\vec{n} \cdot \vec{e}_z}{|\vec{n}| \cdot |\vec{e}_z|}\right) \tag{3.5}$$

For angles $\alpha > 90°$ it's replaced with the supplementary angle[1] so that $|\alpha| \in [0, 90]°$.

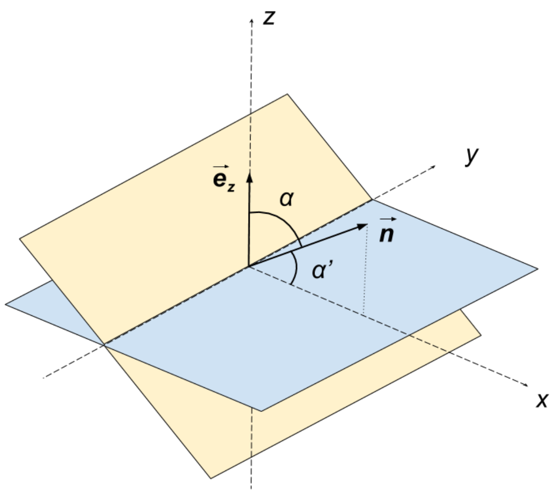


Figure 3.4: *Ground plane (blue), object plane (yellow), the plane normal, $\vec{n}$ and the ground normal $\vec{e}_z$.*

(3) Object extraction

Now the angles of every point surface is known. The purpose of this step is to remove all the points that are assumed part of the ground so that only the object points remain.

From chapter 3.1.1 the 90° requirement is stated. A desired threshold value, τ, is set and the data separated as follows:

$$\alpha \leq \tau \quad \leftrightarrow \quad \text{ground point} \tag{3.6}$$

$$\alpha > \tau \quad \leftrightarrow \quad \text{object point} \tag{3.7}$$

The points classified as object points have an angle between it's surface and the ground on the interval $[\tau, 90]°$.

In a new matrix $\mathbf{D'}$, with the same size as $\mathbf{D}$, the r-values are kept for the object points, but set to zero for the ground points.

(4) Clustering

The fourth step is to group the object points into k clusters based on the distances inside the cluster.

[1]Two supplementary angles sums up to 180°.

Again the 3×3 neighbourhood in is considered, now for $\mathbf{D}'$ and zero padded. To keep control of the clustering process a third dimension is added. The matrix size of $\mathbf{D}'$ now equals $(32 \times \frac{n}{32} \times 2)$. The purpose of this layer is to flag all points with their current cluster status. Table 3.1 gives an overview over the used flags. First the difference between the

Flag ID	Point status
-1	ground point
0	object point, not assigned
$[1, \rightarrow\rangle$	assigned to cluster i

Table 3.1: *Flag status overview.*

centre point and the neighbours in the 3×3 mask is computed. If the difference is below a given threshold that point is assigned with same cluster flag as the centre point. If not the flag remains zero. This process continues for the entire $\mathbf{D}'$ matrix until all object points are assigned. Each time a cluster is complete, i.e. it can't grow any more, the cluster ID i is increased by one and a new, unflagged object point ant it's neighbours are considered.

Finally the clusters with few points are removed by change the flag to -1. These points are not taken into further calculations.

(5) Vehicle verification

The final step is to verify which cluster that represents the vehicle. The routine described in the following are applied to one cluster at the time until the a match is found. The first positive verification is assumed to be the vehicle. If any step declares the cluster disqualified the method returns on to the next cluster.

First the outer dimension of the cluster is measured. If there's a significant difference in either the cluster's height, width or length compared the vehicle of interest the cluster is disqualified.

Secondly the gradients along the z-axis, i.e. column wise, are studied. From chapter 3.1.1 it's stated that the gradients should be negative or zero.

3.2 Pose estimation

With the term *pose* it's understood the position, orientation and rotation of an object. In this paper it referrers to the pose of the leading vehicle in relation to the ego-vehicle. The symbols used to describe the relative pose are shown in figure 3.5, with ρ as distance, γ as orientation angle and ω as rotation angle.

Methods that estimate these three parameters are described in the following. For some of them more than one approach is described. The results are then compared and discussed in chapter 5 and 6.

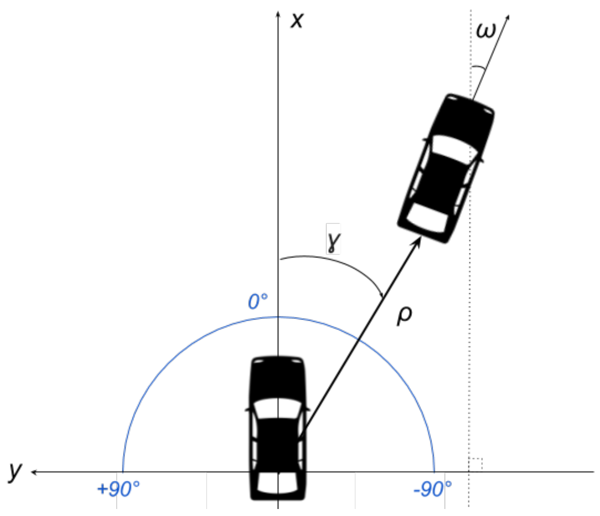


Figure 3.5: *Angles and axis references for pose variables.*

3.2.1 Implementation of POSIT

For implementation of the POSIT method the proposed source code by DeMenthon et al. is used. The method is found in appendix C and is named classicPosit.m [21]. The method takes the image and model feature point coordiante pairs as input and returns the approximated 3×3 rotation matrix, **R**, and the 3×1 translation vector, **T**.

3.2.2 Rotation, ω

The rotation estimate is based on the rotation matrix, $\boldsymbol{R}$, returned by the POSIT method. Of interest is the vehicles rotation in the ground plane, i.e. about the z-axis.

First the 3×3 rotation matrix is decomposed to Euler angles, θ:

$$\boldsymbol{R} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix} \tag{3.8}$$

$$\theta_x = \text{atan2}\left(\frac{r_{32}}{r_{33}}\right) \tag{3.9}$$

$$\theta_y = \text{atan2}\left(\frac{-r_{31}}{\sqrt{r_{32}^2 + r_{33}^2}}\right) \tag{3.10}$$

$$\theta_z = \text{atan2}\left(\frac{r_{21}}{r_{11}}\right) \tag{3.11}$$

$$\boldsymbol{\theta} = [\theta_x, \theta_y, \theta_z] \tag{3.12}$$

where the elements of $\boldsymbol{\theta}$ holds the rotation angles around the x-, y- and z-axis respectively. Thus the ω estimate is set to:

$$\hat{\omega} = \theta_z \tag{3.13}$$

3.2.3 Relative distance, ρ

Two methods estimate the relative distance. The results are presented and compared in chapter **??**. For compareability both methods estimates the distance between the reference points set for both vehicles.

Method 1

The first method is based on the POSIT method used for pose estimation. It has two outputs, a translation vector and a rotation matrix of the detected object. The translation vector $\boldsymbol{T} = [t_x, t_y, t_z]$ represents the position of the object's reference point, P, in relation to the current origin, O. The length of this vector equals the distance between these two reference points:

$$d = ||\boldsymbol{T}|| = \sqrt{t_x^2 + t_y^2 + t_z^2} \tag{3.14}$$

The distance relative the ego-vehicle reference point, R, is obtained using vector addition. With $\vec{d_{OR}}$ representing the position of R in relation to the origin, O, and $\vec{d_{ref}}$ point P relative the reference point:

$$\vec{d_{ref}} = \boldsymbol{T} - \vec{d_{OR}} \tag{3.15}$$

and the relative distance

$$\rho_1 = |\vec{d_{ref}}| \tag{3.16}$$

The relationship of the vectors and points mentioned above are visualized in figure 5.8.

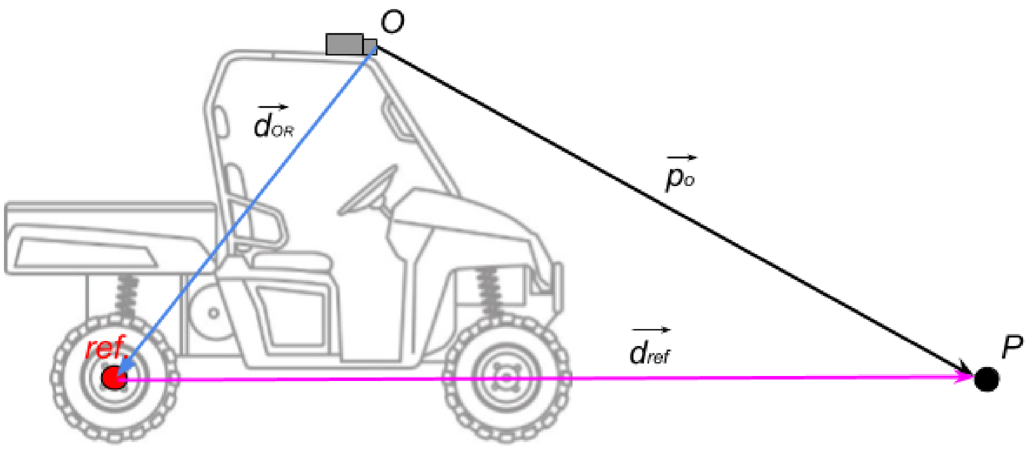


Figure 3.6: *The relationship between the position vectors and the origin, O, the object point, P, and the ego-vehicle reference point, R.*

Method 2

The second method is based on the LiDAR vehicle detection method. When the vehicle is successfully detected the distance is simply defined by the closest point in that cluster. Then, as for method 1, the distance is converted so it is relative the ego-vehicles reference point too.

A vector, $\vec{c}$, containing all detected point coordinates on the vehicle's surface, is returned by the detection method. First all the euclidean distance for each point is calculated. This distance corresponds to the vehicle's point closest to the LiDAR. Then the shortest distance is selected as point of interest. At last the same procedure as above is performed:

$$\vec{d_{ref}} = \vec{p_{closest}} - \vec{d_{OR}} \tag{3.17}$$

$$\rho_2 = |\vec{d_{ref}}| \tag{3.18}$$

3.2.4 Orientation, γ

The orientation can be calculated using information in the translation vector, $\boldsymbol{T}$, as found in chapter 3.2.1. The orientation is defined as the angle in the xy-plane between the x-axis and $\boldsymbol{T}$. The geometrical properties are illustrated in figure 3.7.

If $\boldsymbol{T} = [\Delta x, \Delta y, \Delta z]$, γ is defined a

$$\gamma = \tan^{-1}\left(\frac{\Delta y}{\Delta x}\right) = \sin^{-1}\left(\frac{\Delta y}{\rho}\right) \tag{3.19}$$

Negative γ values corresponds to a vehicle orientation on the right hand side and positive values to the left (seen from the ego-vehicle in the driving direction).

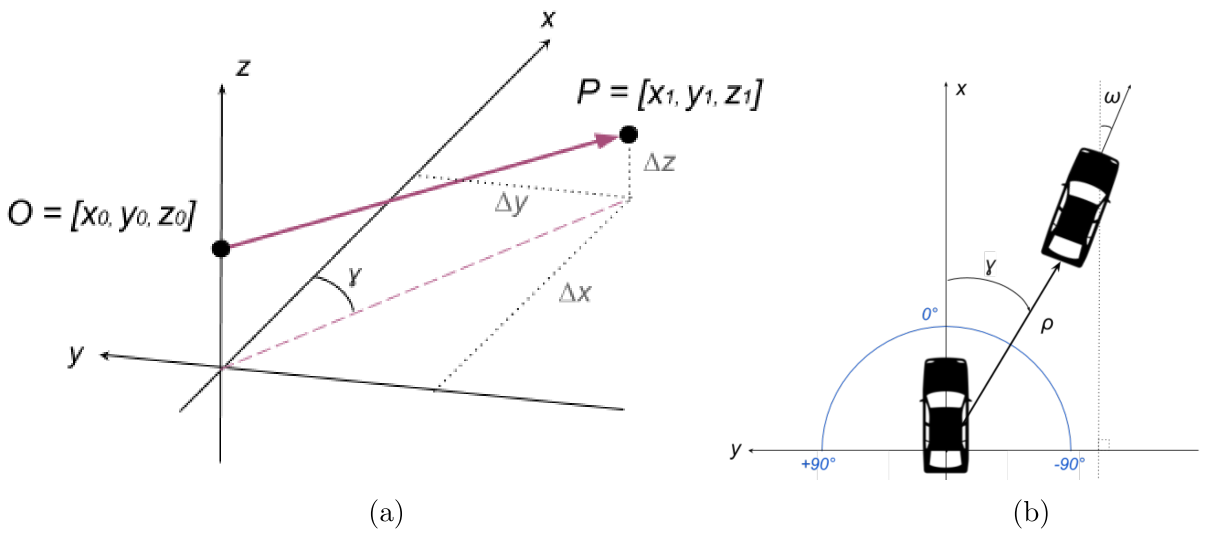


Figure 3.7: ***(a)*** *Geometrical properties of the translation vector,* $\boldsymbol{T}$, *(purple) between the origin, O, and a point, P.*
(b)

3.3 Estimation of relative velocity

The relative velocity, v, between the two vehicles is to be estimated. This parameter is of interest for future implementation as an input to the engine control system regulating the speed of the autonomous vehicle.

Known parameters and estimates obtained from other methods are the relative distance, time stamps of gathered sensor data and the ego vehicle speed, acceleration and position.

From physics the equation of motion for non-accelerated objects says:

$$\bar{v} = \frac{d}{\Delta t} \tag{3.20}$$

where d is the travelled distance, Δt the duration and $\bar{v}$ the average velocity. Inserting the relative distance between the two vehicles, ρ, and the time difference between the last two measurements, $t(n) - t(n-1)$, the average velocity of that time interval can be approximated. The smaller the Δt, the better the estimate.

Another approach is using the Backward Euler method:

$$\hat{v}(n) = \dot{d}(n) \approx \frac{1}{\Delta t}\Big(d(n) - d(n-1)\Big) \tag{3.21}$$

This estimate is based on the previous and current measured distance and the time interval of the two observations.

To reduce the effect of potential bad distance estimates, a weighted mean including the last three velocity estimates is implemented:

$$\hat{v}(n) = 0.85 \cdot \bar{v}(n) + 0.10 \cdot \bar{v}(n-1) + 0.05 \cdot \bar{v}(n-2) \tag{3.22}$$

4. Experiments

This chapter contains descriptions of all tests and experiments performed and their results. To ensure reproducibility the first section describes the vehicle and sensor configuration used, including relevant specifications. Sensor range and limitations are also discussed. Section 4.2 is a presentation of the data material used - how it's gathered, what data types it contains and some calculations on the sample rate used during the test run. Test and experiment descriptions are presented from section 4.3. Every test starts with a short explanation of it's purpose.

4.1 Vehicle configuration

The ego-vehicle is a custom modified Polaris Ranger ATV as the one displayed in figure 4.1. It's equipped with several sensors, see the list below, and all necessary actuators and sensors needed for autonomous driving.

- 2 Flir Grasshopper3 8.9 MP monochrome USB cameras
- 1 Flir Grasshopper3 8.9 MP color USB camera
- 1 Velodyne HDL-32E LiDAR
- 1 IMU
- 1 GPS antenna

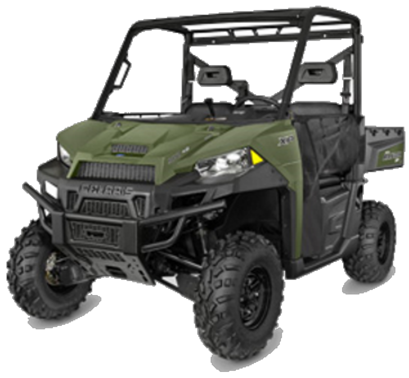


Figure 4.1: *An original unmodified Polaris Ranger ATV [4]*

The listed sensors are located on the roof top. Exact positions are found in appendix A on page 49. Additional hardware for data processing and communication are located in the boot lid, physically protected from rain, snow, potential harmful objects, direct sunlight etc. It also contains an external cooling to prevent over heating.

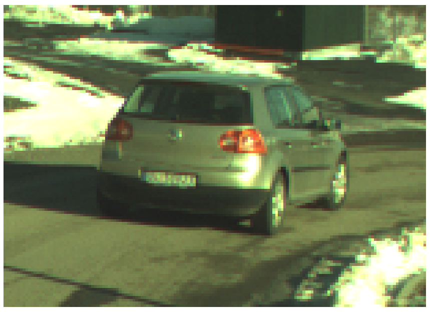


Figure 4.2: *Screenshot of the Golf during a test run.*

4.1.1 Vehicle specifications

Ego-vehicle

The ego-vehicle, also refered to as UGV, is a four wheel all terrain vehicle. Maximum speed is 40 km/h. An unmodified model is shown in figure 4.1

Leading vehicle

The leading vehicle used in this paper during the test runs is a silver metallic 5-doors VW Golf V (mod. 2008), see figure 4.2. Technical documentation is found in appendix B on page 51.

4.1.2 Camera specifications

The ego-vehicle has three cameras in total, two monochrome cameras for stereo vision and one colour. Specification are given as found in the data sheets [27] [28]:

	Color vision	Stereo vision
Type	Color	Grayscale
Model no.	GS3-U3-89S6C-C	GS3-U3-89S6M-C
Max. resolution	4096 x 2160	4096 x 2160
Resolution used	1688 x 1352	1688 x 1352
Max. frame rate	43 FPS	43 FPS
Frame rate used	6 FPS	6 FPS
Pixel size	3.45 µm	3.45 µm
Focal length	3.80 mm	3.80 mm
FOV	74.9° x 63.1°	74.9° x 63.1°

Table 4.1: *Camera specifications*

4.1.3 LiDAR specifications

Model no.	Velodyne HDL-32E
Channels	32
FOV, horizontal	360° horizontal
Angular resolution, horizontal	0.1° - 0.4°
FOV, vertical	[+10°, −30°]
Angular resolution, vertical	1.33°
Max. range	80 - 100 m
Range accuracy	±2 cm
Rotation rate	5-20 Hz
Rotation rate used	10 Hz

Table 4.2: *LiDAR specifications [9]*

The LiDAR returns the positions of surrounding objects reflecting the emitted laser pulses. The information is stored as spherical coordinates (r, ϕ, θ), with r representing radial distance, ϕ polar angle and θ azimuth angle as illustrated in figure 4.3. The data is converted so that zero azimuth points in the positive x direction with positive rotation counter clockwise (in the xy-plane) and with polar angle relative the z-axis. The LiDAR has coordinates (0,0,0). The LiDAR is located on the rooftop of the ego-vehicle. Due to the vehicle geometry this causes some potential occlusion issues as described in the following.

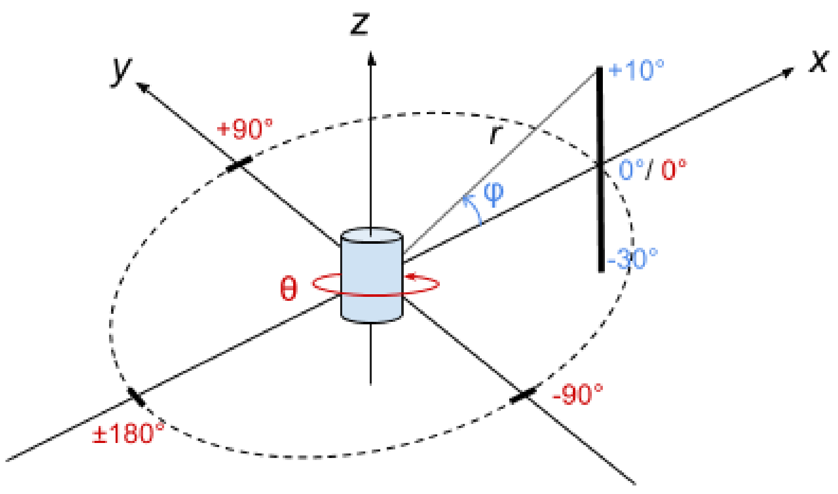


Figure 4.3: *Illustration of LiDAR configuration. The X-axis is pointing in the driving direction, Y to the left and Z upwards. Azimuth angle, θ, in red, polar angle, ϕ, in blue and radial distance, r, in black. [5]*

Forward occlusion

There are no limitations in the LiDAR viewing field in front of the car caused by it's location, but with a vertical FOV of $-30°$ the minimum horizontal detection range, d, is limited:

$$d = \frac{H + h}{\cos\phi} = \frac{2.079\text{ m} + 0.150\text{ m}}{\cos 30°} = 2.574\text{ m} \tag{4.1}$$

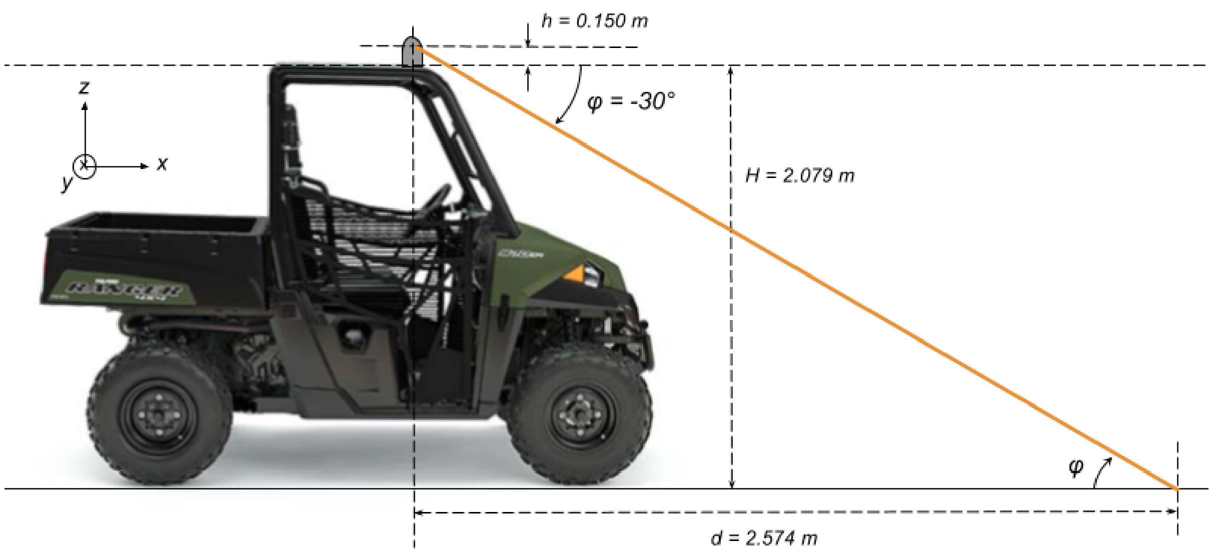


Figure 4.4: *LiDAR configuration, right view. No roof top occlusion, [6] [5]*

Sideways occlusion

Sideways the rooftop occludes the emitted laser beams and causes major limitation in the minimum horizontal detection range. As the LiDAR is not perfectly centred there are some differences between right and left side, represented by d_1 and d_2 respectively. The angles α_1 and α_2 represents the new vertical negative angle for the actual FOV. The occlusion problem is illustrated in figure 4.5.

$$\alpha_1 = \tan^{-1}\left(\frac{0.711\text{ m}}{0.150\text{ m}}\right) = 78.1^\circ \tag{4.2}$$

$$\alpha_2 = \tan^{-1}\left(\frac{0.717\text{ m}}{0.150\text{ m}}\right) = 78.2^\circ \tag{4.3}$$

$$d_1 = \frac{H+h}{\cos\alpha_1} = \frac{2.079\text{ m} + 0.150\text{ m}}{\cos 78.1^\circ} = 10.810\text{ m} \tag{4.4}$$

$$d_2 = \frac{H+h}{\cos\alpha_2} = \frac{2.079\text{ m} + 0.150\text{ m}}{\cos 78.2^\circ} = 10.890\text{ m} \tag{4.5}$$

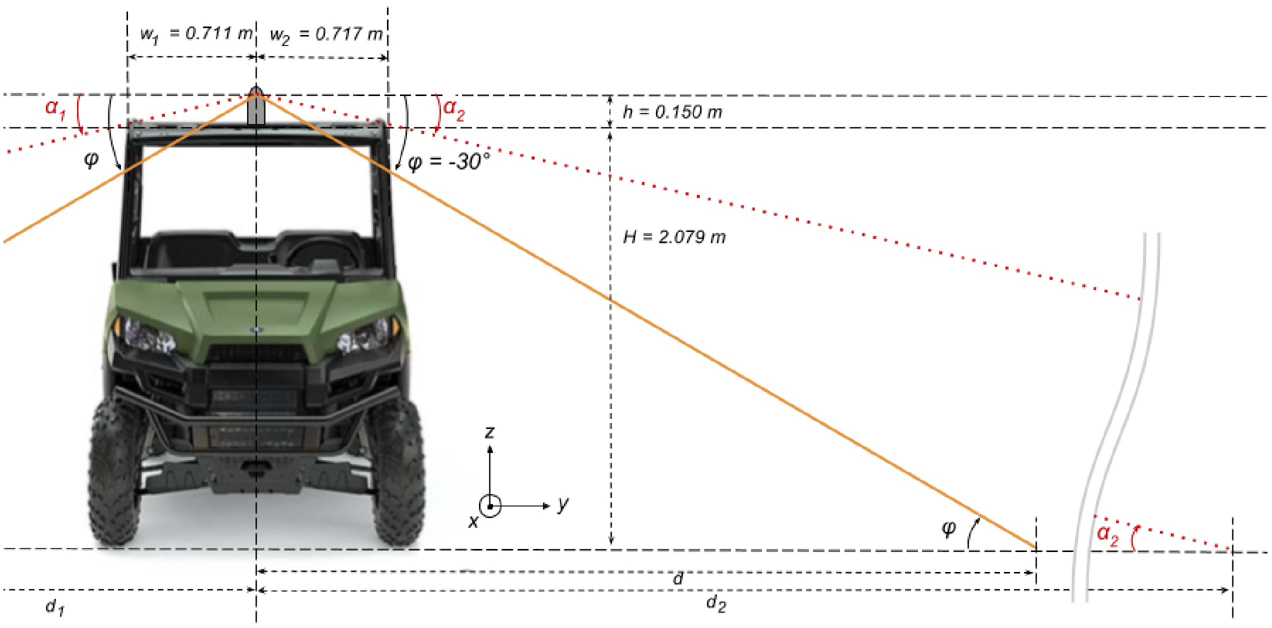


Figure 4.5: *Left and right occlusion. Viewing field set by LiDAR specs. marked with orange lines. Maximum practical downward viewing field due to rooftop occlusion marked with dotted red lines. [6] [5]*

Backward occlusion

This is considered not relevant for this paper, as the leading vehicle is assumed to be positioned within the azimuth sector 90.

4.1.4 Hardware

The hardware of the ego-vehicle, except the already listed sensors, are located in the boot lid. This gives protection from shifting weather conditions and external potential harmful objects. An external cooling system is also installed to prevent over heating. A high-performance computer is installed as processing unit with improved graphics card and processor.

4.2 Data material

This section is meant to give an overview of the data used for testing in this paper. First a description on how it's collected, then a short presentation of the data types and last some calculations on the recording frequency.

4.2.1 Data gathering

The data set used in this paper was gathered during one test run in Kjeller, Norway, mid-February 2017. In total 77.42 GB of driving data, lasting 21 minutes and 37 seconds, and 9.2 GB of still stand recordings for calibration purposes was gathered. The data was recorded as a ROS bag-file consisting of the following data:

- color images (6 fps)
- stereo grayscale images (6 fps)
- 360° LiDAR point cloud data (rotation speed: 10 Hz)
- ego-vehilce odometer data
- GPS data of ego-vehicle and leading vehicle

The route was driven on asphalt roads, with and without traffic (i.e. pedestrians, driving and parked cars etc.) in both urban and rural environments. Both (convoy) cars were driven manually during the test run. The route was chosen to include both flat, uphill and downhill roads, left and right curves, as well as varying distance between the two cars to ensure to obtain all possible perspectives of the leading vehicle from the ego-vehicle point of view. Weather conditions was -8 °C, clear blue sky, sunny and light winds.

4.2.2 Presentation of data

The onboard software is based on ROS, also known as *Robot Operating System*, a framework especially designed for robot applications [29]. All data collected during test runs, regarding their type, are organised and stored in the ROS specific .bag format. Bag-files are made readable in Matlab by the *Robotics System Toolbox.*

Different data types are stored under different topics, and each topic holds additional information relevant for this specific data type. The ROS-environment provides synchroized time stamps for all data consecutively when recorded. An overview of the data types is given in the next three subsections.

LiDAR data

The LiDAR provides 3D point cloud data of the surroundings. The LiDAR data is organised under the bag topic *lidar_sweep* and is stored as spherical coordinates (r, ϕ, θ). For every rotation, or sweep, $n \times 32$ laser beams are emitted - n in the horizontal plane and 32 in the vertical plane. A full 360° rotation generates approx. 60.000 data points.

The data points are organised in a tree structure. Each rotation generates one message cell, *lidarSweep2*. Each of these holds n *lidarScan2* messages which again contains

32 *laserReturn* submessages. This is where the information about the reflected beam and the estimated radial distance to the object is found. An illustrating this structure is given in figure 4.6 The message cells does not hold the azimuth and polar angles explicit, only

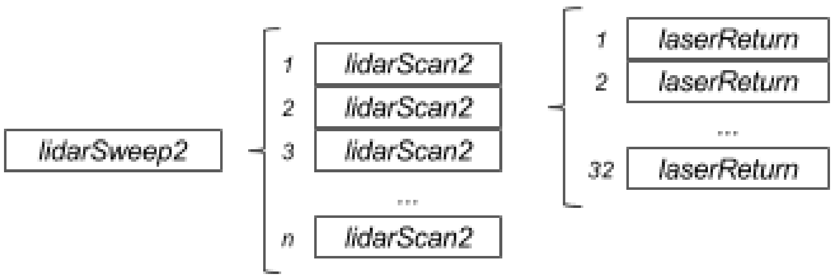


Figure 4.6

the radial distance. However the angular range and number of emitted beams are known: 32 in the vertical plane and n in the horizontal plane. According to message numbering, r_θ and r_ϕ, the angles can be calculated as follows:

$$\begin{aligned}\theta &= (r_\theta - 1)\cdot\Delta\,\theta \\ &= (r_\theta - 1)\cdot\frac{360^\circ}{n}\,, \qquad r_\theta \in [1, n]\end{aligned} \tag{4.6}$$

$$\begin{aligned}\phi &= (r_\phi - 1)\cdot\Delta\,\phi \\ &= (r_\phi - 1)\cdot\frac{40^\circ}{32}\,, \qquad r_\phi \in [1, 32]\end{aligned} \tag{4.7}$$

Comment: $\Delta\,\phi$ and $\Delta\,\theta$ refers to the angular resolution in the vertical and horizontal plane respectively.

Based on the assumption that the leading car normally is located in the front half sector relative the ego-vehicle the data is reduced to only contain the reflection point in the sector $\theta = \pm 60^\circ$. See illustration below. This reduces the amount of LiDAR data to be processed by a third.

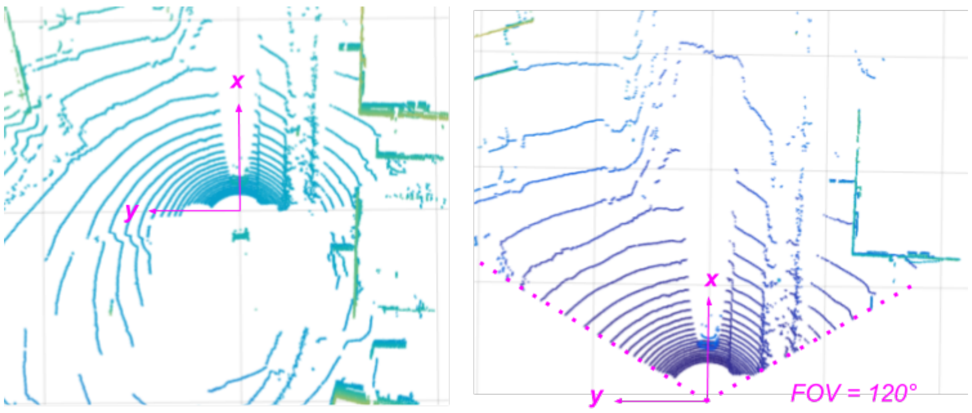


Figure 4.7: *Top-down view on two LiDAR point clouds. To the left a complete view of* 360° *and to the right one reduced to a horizontal FOV of* 120°.

To ensure equal axis references the point cloud is rotated 180° around the z-axis with the x-axis is pointing in the driving direction of the ego vehicle. This is handled by the function *lidar2polar.m* which also converts the LiDAR ROS-message into a 3xn mat-file with polar coordinates. Axis and angle references are displayed in figure 4.3 on page 20.

Images

Three cameras provides images independently, two gray scale stereo cameras and one colour camera. The images are organised in three individual bag topics, *camera/stereo_left*, *camera/stereo_right* and *camera/center*. Each topic holds k image messages with information about image size, time stamp, encoding and the image data. All images are recorded with resolution 1352×1688 pixels.

The gray scale images are stored directly as intensity images, ranging 0 to 255. The colour images are encoded as bgr8 Bayer vectors. The build-in Matlab function *readImage* decodes them to ($height \times width \times 3$) RGB images using demosaicing.

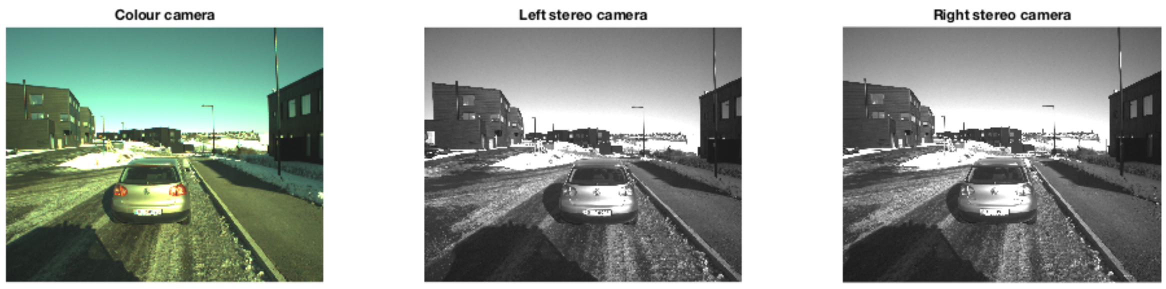


Figure 4.8: *Images captured from the three cameras. From the left: decoded RGB image, left stereo image and right stereo image.*

Odometer data

Drive information from the ego-vehicle is stored with the topic *vehicle_measurements*. The data is given as Ackerman data, including speed, acceleration, steering angle, steering angle velocity and jerk.

Positioning data

GPS data was during the test run collected and stored in a separate bag. This data was the converted to .bin files which can be converted to Matlab struct objects using the methods *read_navlog_GPS.m read_navlog_navp.m.*

4.2.3 Comments on data recording frequency

The maximum speed of the ego-vehicle in use is 40 km/h, i.e. 11.11 m/s. As specified in the section 4.2.1 the LiDAR rotates at a frequency of 10 Hz and the cameras stores 6 frames pr. second (fps) each. Assumed that the max. speed limit are equal for both vehicles, i.e. the leading vehicle never exceed 40 km/h, there is a possible maximum relative displacement between the frames at 1.85 meters and for the LiDAR 1.11 meters.

From table 4.1 and 4.2 maximum recording rate for the cameras and the LiDAR are 43 fps and 20 Hz respectively, i.e. the maximum displacement can be reduced to 0.25 m (camera) and 0.55 m (LiDAR). A high frame- and rotation rate results in a bigger amount of data to be processed and therefore require a system capable to process it accordingly fast. If the computations take too long it will cause a lag that increases for every loop iteration and make the entire system unstable. On the other hand, if processing capacity is not an issue, it's preferable to keep the rates as high as possible to increase the accuracy and give updates for the control loop.

4.3 Calibration

4.3.1 Camera calibration

This test is performed by employees at FFI. The goal was to calculate the transformation matrix, $\mathcal{T}$, for the two stereo cameras. The result is given below:

$$\mathcal{T} = \begin{bmatrix} 0.99987424 & 0.00792202 & -0.01373866 & -0.41291371 \\ -0.00779913 & 0.99992931 & 0.00897542 & 0.00093172 \\ 0.01380879 & -0.00886715 & 0.99986534 & 0.00988889 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.8}$$

Please note: the $\mathcal{T}$ matrix is originally computed with 16 significant figures precision. However only 8 significant figures are displayed in this paper.

The transformation matrix can be used to convert any image point from the left camera, $\boldsymbol{p_L} = [x, y, z, 1]'$, to the corresponding right camera image coordinate, $\boldsymbol{p_R} = [X, Y, Z, 1]'$, and vice versa. This is useful when testing the pose estimation algorithm to ensure the exact corresponding coordinates are obtained from the left and right stereo camera. More on this test in chapter 5.2. From figure 4.9 two corresponding image points are marked,

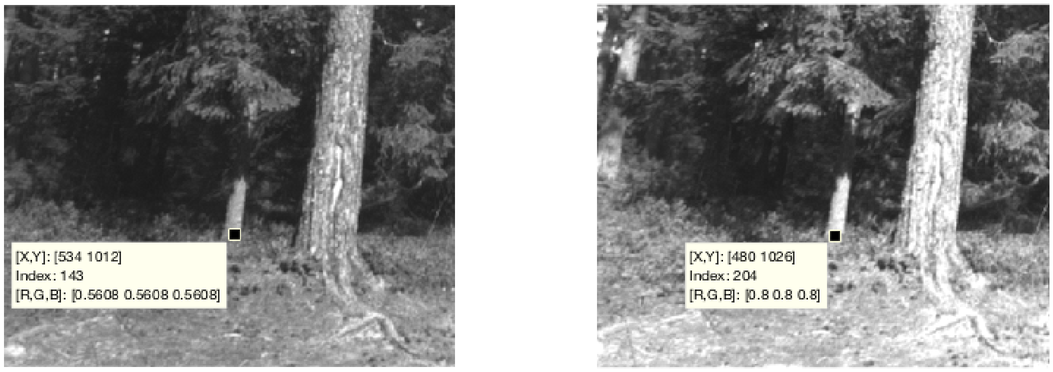


Figure 4.9: *Left and right stereo image of the same scene. Corresponding points are marked manually in Matlab.*

$\boldsymbol{p_L} = [534, 1012]$ and $\boldsymbol{p_R} = [480, 1026]$. The image point in the opposite stereo camera can then be computed:

$$\begin{aligned} \boldsymbol{p_L}' &= \mathcal{T}\ \boldsymbol{p_R} \\ &= \mathcal{T}\ [\boldsymbol{p_R}, 0, 1] \\ &= \mathcal{T}\ [480, 1026, 0, 1] \\ &= [479, 1026, 1.7, 1] \end{aligned} \tag{4.9}$$

$$\begin{aligned} \boldsymbol{p_R}' &= \mathcal{T}\,\boldsymbol{p_L} \\ &= \mathcal{T}\,[\boldsymbol{p_L}, 0, 1] \\ &= \mathcal{T}\,[534, 1012, 0, 1] \\ &= [526, 1016, 0.1, 1] \end{aligned} \tag{4.10}$$

4.3.2 LiDAR vs. camera calibration

The purpose of this test is to find a method that converts a LiDAR point cloud coordinate to the corresponding image coordinate. This opens the possibility to verify founds in the point cloud by looking at the corresponding visual features. Weak perspective projection, as described in chapter 2.3, is used for this purpose. The method consists of two steps.

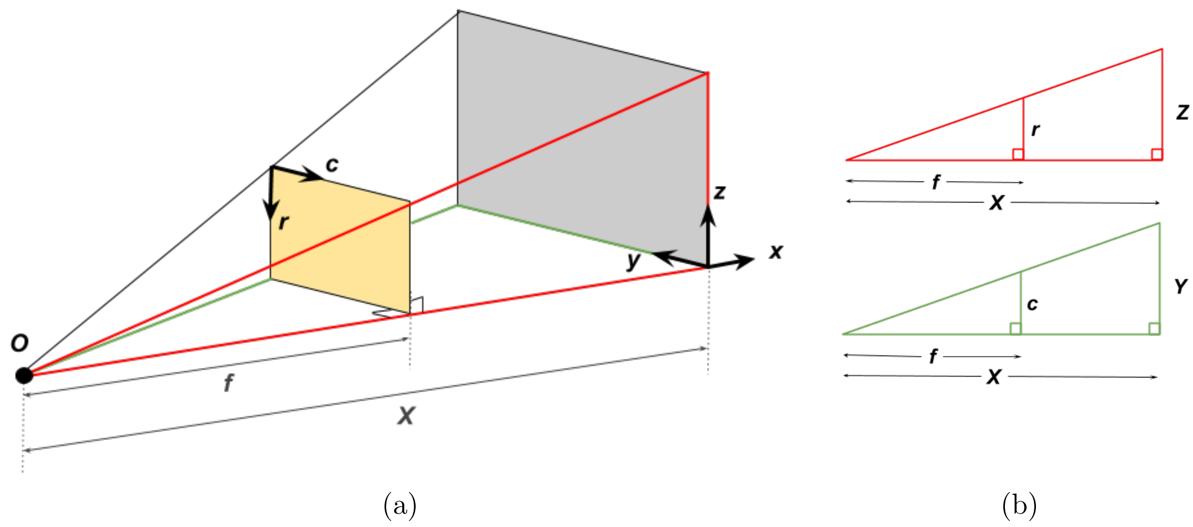


Figure 4.10: ***(a)*** *Image plane (yellow) and world space (gray) relative the observer, O. f is the focal length*

- **Step 1 - weak perspective estimate**
  First the 2D coordinates are estimated using weak perspective projection. f is the focal length and the x-axis the optical axis (see fig. 4.10).

$$x = f \cdot \frac{-Y}{X} \tag{4.11}$$

$$y = f \cdot \frac{-Z}{X} \tag{4.12}$$

  The minus signs ensures correct signs of x and y when converting from the world coordinate system to the image plane coordinate system.

- **Step 2 - remap**
  Secondly x and y from step 1 need to be remapped to match the dimensions of the images, 1352×1688.

$$r = 1 + \frac{(y - y_{min})\,(1352 - 1)}{(y_{max} - y_{min})} \tag{4.13}$$

$$c = 1 + \frac{(x - x_{min})\,(1688 - 1)}{(x_{max} - x_{min})} \tag{4.14}$$

n_{max} and n_{min} are the minimum and maximum values of all x and y computed in step 1.

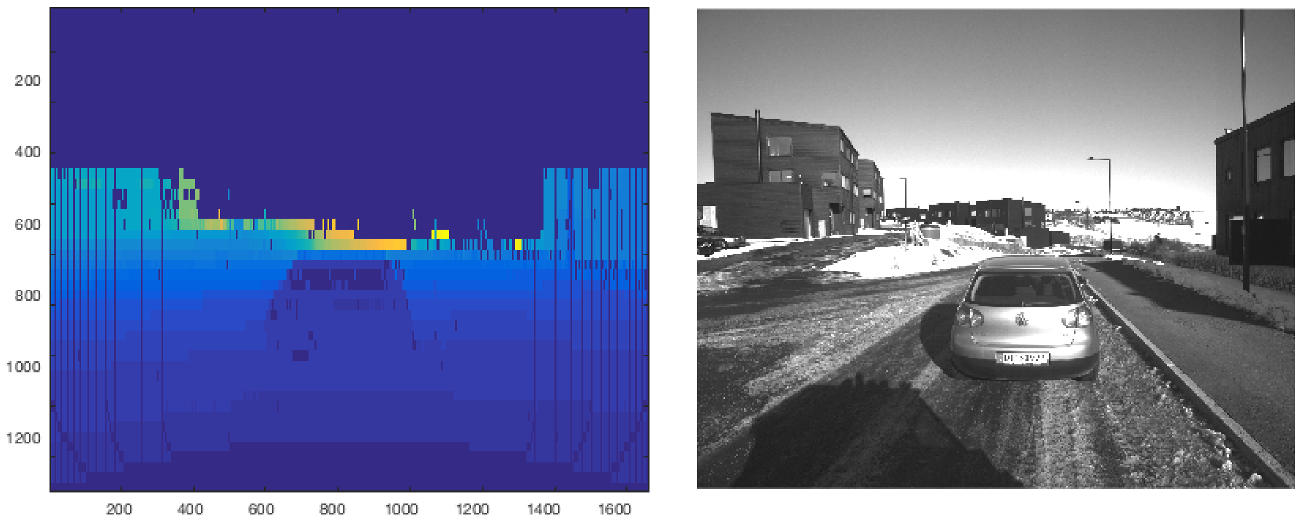


Figure 4.11: *Reconstructed image from point cloud data to the left. Original image from left stereo camera to the right.*

4.4 LiDAR data unit tests

The purpose of this test is to decide and verify the units of the radial distances measured by the LiDAR. The test is performed on two objects with known physical dimensions. By comparing the measured and the actual lengths a correction ratio can be computed. To verify the results this ratio is applied to the data set and a control object is measured. Desired unit is millimetres.

Test 1

The first comparison is made on a person of known height, see fig. 4.12. A plot of the point cloud in Matlab allows access to the coordinates of any point simply by click at it. As shown in fig.4.12a head and toe have this cartesian (x,y,z) coordinates:

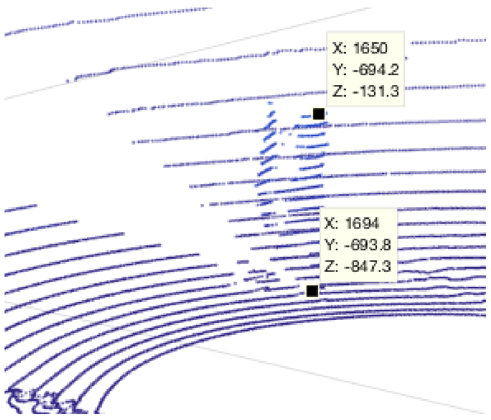
$$\begin{aligned} \text{Head: } & (1650, -694.2, -131.3) \\ \text{Toe: } & (1694, -693.8, -847.3) \end{aligned}$$

This gives the euclidian distance:

$$\begin{aligned} d_{HT} &= \sqrt{(1694-1650)^2 + \big((-693.8)-(-694.2)\big)^2 + \big((-847.3)-(-131.3)\big)^2} \\ &= 717.4 \end{aligned} \tag{4.15}$$

The physical height of this person is 158 cm. This gives the ratio, r_1:

$$r_1 = \frac{158\text{ cm}}{717.4} = 0.2204\text{ cm} = 2.204\text{ mm} \tag{4.16}$$



(a)

(b)

Figure 4.12: *Visual image and corresponding LiDAR data of a person of known height.*

Test 2

The second test is performed on recordings of a square plate made for camera calibration purposes. The plate has outer dimensions of 79.6 x 79.6 cm. Information about the chessboard pattern is here not of interest. From Matlab, see fig. 4.13a, these coordinates of two edges are obtained:

$$\text{Top edge: } (2706, -52.33, -167.8)$$
$$\text{Right edge: } (2576, -310.9, -73.07)$$

which gives the euclidean distance

$$\begin{aligned} d_{TR} &= \sqrt{(2576-2706)^2 + \left((-310.9)-(-52.33)\right)^2 + \left((-73.07)-(-167.8)\right)^2} \\ &= 304.5 \end{aligned} \tag{4.17}$$

Compared with the physical length of 79.6 cm the ratio, r_2 becomes:

$$r_2 = \frac{79.6 \text{ cm}}{304.5} = 0.2614 \text{ cm} = 2.614 \text{ mm} \tag{4.18}$$

Verification

Multiplying any coordinate or distance in the LiDAR point cloud with this ratio converts it into the units of cm or mm respectively. To illustrate this, and to verify whether the two ratios obtained are correct or not, a verification test of the leading vehicle, the Golf, is performed.

From the figure on the left, 4.14, coordinates of the left and right rear wheels are given:

$$\text{Left rear wheel: } (2308, 464.3, -782.6)$$
$$\text{Right rear wheel: } (2294, -364.5, -772.2)$$

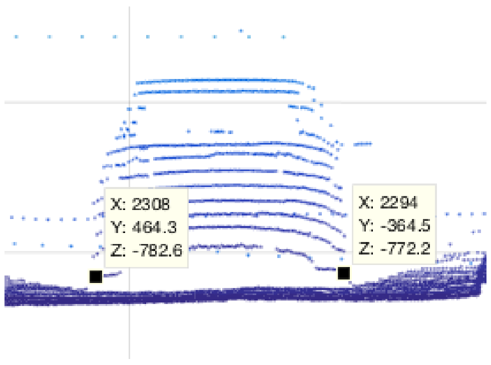
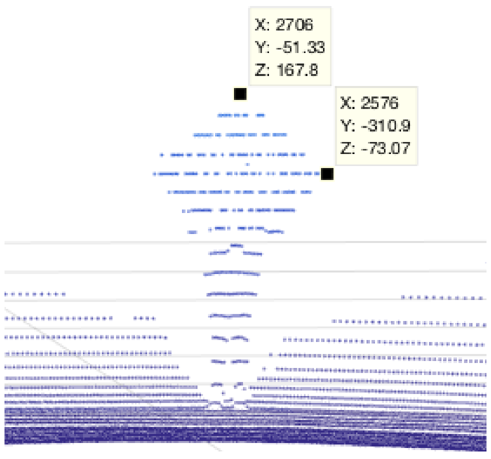
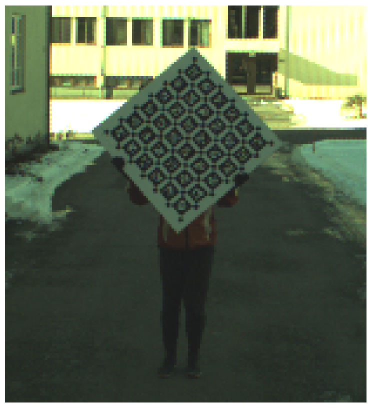


Figure 4.14: *Point cloud data of a Golf V from behind. Outermost points on rear wheels are marked with corresponding coordinates.*



(a)



(b)

Figure 4.13: *Visual image and corresponding LiDAR data of a calibration board with known dimensions.*

with the distance

$$d_{LR} = \sqrt{(2294 - 2308)^2 + \big((-364.5) - 464.3\big)^2 + \big((-772.2) - (-782.6)\big)^2} \\ = 829.0 \tag{4.19}$$

Multiplying with the ratios r_1 and r_2, separately, the actual car width should be returned if the factor are correct:

$$\begin{aligned} w_1 &= d_{LR}\, r_1 \\ &= 829.0 \cdot 2.204 \text{ mm} \\ &= 1827.1 \text{ mm} \end{aligned} \tag{4.20}$$

$$\begin{aligned} w_2 &= d_{LR}\, r_2 \\ &= 829.0 \cdot 2.614 \text{ mm} \\ &= 2167.0 \text{ mm} \end{aligned} \tag{4.21}$$

Comparing these two estimated widths with the actual width 1786 mm, as found in the technical documentation on page 51, the least deviation is found for the ration r_1. The ration, r, is therefor set to 2.2 and 0.22 for conversion to millimetres and centimetres respectively.

After adding the factor, $r = 2.2$, the wheel coordinates now are:

$$\text{Left rear wheel: } (5078, 1021, -1722) \text{ mm}$$
$$\text{Right rear wheel: } (5047, -801.9, -1699) \text{ mm}$$

and the distance

$$d_{LR}{}' = \sqrt{(5047 - 5078)^2 + \big((-801.9) - 1021\big)^2 + \big((-1699) - (-1722)\big)^2} \text{ mm} \\ = 1822 \text{ mm} \tag{4.22}$$

which equals

$$\begin{aligned} r \cdot d_{LR} &= 2.2 \cdot 829.0 \text{ mm} \\ &= 1823.8 \text{ mm} \\ &\approx d_{LR}' \end{aligned} \tag{4.23}$$

Please note that there are uncertainty related to both of these test. Did the LiDAR sweep get reflected by the exact reference points chosen (board edges, foot sole, top of the head), or are the located somewhere in the gap between the LiDAR beams and therefore not visible? The resolution, and the precision, in the LiDAR data decreases with growing distance between the source and the reflecting object. Also, error in the "known" measurements has to be considered. For example how the shoes, hair style etc. affect the height of the person measured by the LiDAR, and uncertainty in the measurement of the calibration board.

4.5 Conversion of positioning data

Positioning data returned by the detection algorithm are given relative the LiDAR sensor placed on the rooftop of the ego vehicle. Implementation of the control routine for autonomous driving requires positioning data relative the ego-vehicles rear shaft center point. A convertion of data is therefore required.

Detailed sketches with accurate measurements the ego-vehicle are given in appendix A, see page 49.

Ego-vehicle: LiDAR - IMU

GPS-positioning data are collected by an IMU attached to the ego-vehicle rooftop[1]. For analysis purposes comparision of IMU data, from both the ego- and leading vehicle, and the estimated relative position, a convertion between the LiDAR and IMU are of interest.

Ref. figures A.2 and A.3, with origin at the LiDAR's position and coordiates pointing x forward, y left and z up ref. the driving direction, the offset vector, $\vec{d}$, is set to

$$\begin{aligned} \vec{d}_{LI} &= [\Delta x, \Delta y, \Delta z] \\ &= [-0.474, +0.03, -0.207] \text{ m} \end{aligned} \tag{4.24}$$

$$\vec{d}_{IL} = -\vec{d}_{LI} \tag{4.25}$$

The positioning vector of any point, $\vec{p}$, given relative to the LiDAR is converted relative the IMU, $\vec{p'}$, as follows:

$$\vec{p'} = \vec{d}_{IL} + \vec{p} \tag{4.26}$$

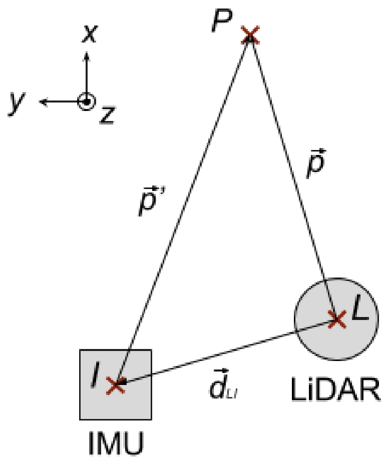


Figure 4.15: *Top-down view of rooftop and a scene point, P. [5]*

[1]internal measurement unit

Ego-vehicle: IMU - ref. point

The center point of the rear shaft is set as the reference point on the ego-vehicle. The IMU is manually assembled underneath the vehicle's roof top, centered along the y-axis. No accurate measurements on the y-axis offset between the IMU and the reference point is given, so it's assumed to be zero. Some minor errors might therefore be considered.

Ref. figure 4.16, distances a, b and c and the offset angle θ are found in the technical documents of the ego-vehicle as found in Appendix A (page 49). The relative position Δx and Δz ($\Delta y \approx 0$ as explained above) is then calculated.

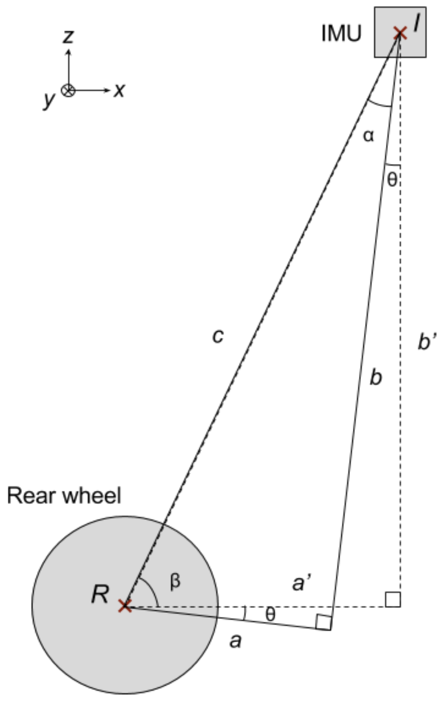


Figure 4.16: *Right view of ego-vehicle illustrating the offset between the IMU and the reference point, R, on the rear shaft. [5]*

$$\begin{aligned}\alpha &= \tan^{-1}\left(\frac{a}{b}\right)\\ &= \tan^{-1}\left(\frac{0.741}{1.538}\right)\\ &= 25.72^\circ \end{aligned} \tag{4.27}$$

$$\begin{aligned}c &= \sqrt{a^2+b^2}\\ &= \sqrt{0.741^2+1.538^2}\ \mathrm{m}\\ &= 1.7054\ \mathrm{m}\end{aligned} \tag{4.28}$$

$$\begin{aligned}\Delta x &= a' \\ &= c\sin(\alpha+\theta) \\ &= 1.7054\sin(25.72^\circ + 2.1^\circ) \\ &= 0.796 \end{aligned} \tag{4.29}$$

$$\begin{aligned}\Delta z &= b' \\ &= c\cos(\alpha+\theta) \\ &= 1.7054\cos(25.72^\circ + 2.1^\circ) \\ &= 1.508 \end{aligned} \tag{4.30}$$

The translation vector, $\vec{d}_{IR}$, then becomes

$$\vec{d}_{IR} = [-0.796, 0, -1.508] \text{ m} \tag{4.31}$$

$$\vec{d}_{RI} = -\vec{d}_{IR} \tag{4.32}$$

Ego-vehicle: LiDAR - ref.point

Given the transition vectors $\vec{d}_{LI}$ and $\vec{d}_{IR}$ the direct convertion from LiDAR positions to position relative the reference point equals the sum of these vectors:

$$\begin{aligned}\vec{d}_{LR} &= \vec{d}_{LI} + \vec{d}_{IR} \\ &= [-0.474, 0.03, -0.207] \text{ m} + [-0.796, 0, -1.508] \text{ m} \\ &= [-1.270, 0.03, -1.715] \text{ m} \end{aligned} \tag{4.33}$$

$$\vec{d}_{RL} = -\vec{d}_{LR} \tag{4.34}$$

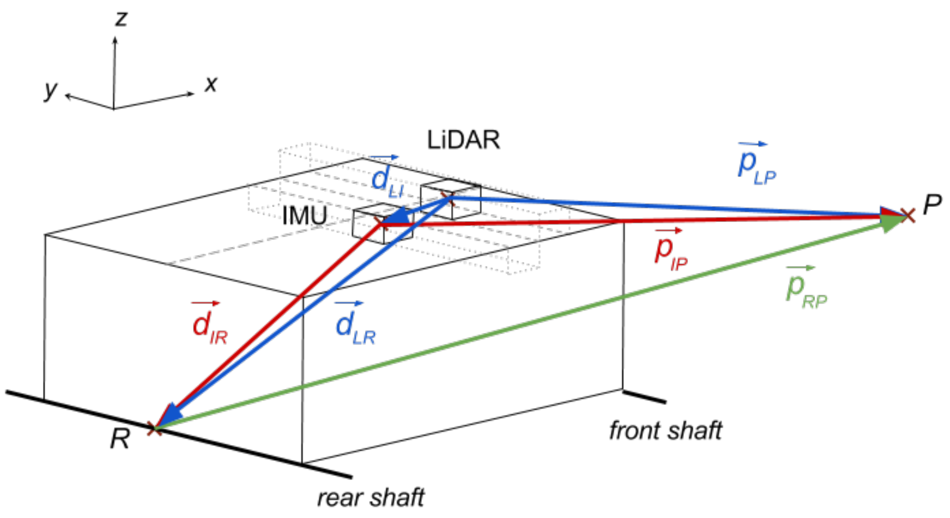


Figure 4.17: *Overview of ego-vehicles translation(?) vectors, $\vec{d}$, and corresponding position vectors, $\vec{p}$, for a random scene point, P. [5]*

Leading vehicle (VW Golf V)

No main reference point is selected for the leading vehicle. For following and anti-collision purposes the shortest distance between the two vehicle is the one of interest, and for the case of pose and velocity estimation the centre point of the license plate is used.

Please see the technical drawings on page 51 and the documentation on the 3D-model in chapter 4.6 for more details.

4.6 Creation of 3D models

To perform a model based pose estimation a 3D model of the object to be detected is required. Two models are created, one of the leading vehicle, a Golf V, and one of the license plate. For both models the license plate's centre point is set as reference point (0,0,0) so that the results are comparable. Results using both models are presented in chapter 5.2.

4.6.1 Leading vehicle

As explained in chapter 2.2 at least four points are required to obtain the rotation and translation matrix of an object. To minimise the risk that occlusion of one or more feature points prevents the pose estimation method from working correctly, several feature points are included in the 3D model. The points are distributed on both sides and the back of the vehicle so that it's rotation relative the ego-vehicle is not an issue, and selected to have visual features that are easy recognisable.

The calculations made to obtain relative position coordinates of the selected feature points are mainly based on measurements found in the technical documentation of the vehicle, see page 51, as it provides the most correct data. Where lack of information prevents obtaining relative coordinate position of a feature point, typically in the z-dimension, data from the LiDAR is applied.

Using the point cloud data for distance calculations between feature points implies a significant uncertainty related to the fact that it don't provide any visual information other than shape. Combined with the resolution vs. distance problem, ref. chapter 4.1.3, the model will be an approximation. To minimize the error as few measurements as possible is based on the point cloud data.

Wheel centre coordinates, W_n

$$W_C = (x_{W_c}, y_{W_c}, z_{W_c}) \tag{4.35}$$

$$x_{W_c} = 4986\text{ mm} + \frac{1}{2}(5878\text{ mm} - 4986\text{ mm}) = 5432\text{ mm} \tag{4.36}$$

$$y_{W_c} = -760\text{ mm} + \frac{1}{2}\big(1021\text{ mm} - (-760)\text{ mm}\big) = 130.5\text{ mm} \tag{4.37}$$

$$z_{W_c} = -1677\text{ mm} + \frac{1}{2}\big((-1722)\text{ mm} - (-1677)\text{ mm}\big) = -1699.5\text{ mm} \tag{4.38}$$

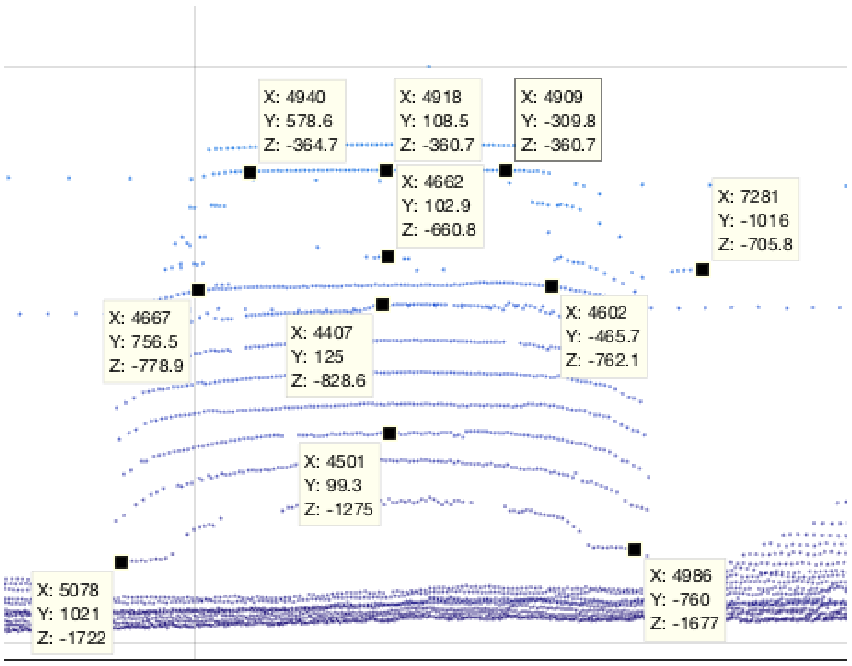


Figure 4.18: *Leading vehicle, point cloud view from behind. Relevant feature coordinates used for calculations in this section are marked with black dots. The vehicle is slightly rotated ref. the ego-vehicle.*

$$\begin{aligned}|P_{ref}W_C| &= \sqrt{(5432-4501)^2+(130.5-99.3)^2+\left((-1699.5)-(-1275\right)^2} \text{ mm}\\ &= 1445.6 \text{ mm}\end{aligned} \tag{4.39}$$

The relative x-axis displacement, Δx_w, between the reference point and the rear shaft can now be computed using pythagoras with catheti Δx_w and Δz_w and the length $|P_{ref}W_C|$ as hypotenuse:

$$\begin{aligned}\Delta x_w &= \sqrt{|P_{ref}W_C|^2-\Delta z^2}\\ &= \sqrt{1445.6^2-\left((-2175)-(-1699.5)\right)^2} \text{ mm}\\ &= 1024 \text{ mm}\end{aligned} \tag{4.40}$$

The displacement in y-direction is found from the technical documentation, and is set to half the length of the shaft width:

$$\Delta y_w = \frac{1}{2}\cdot 1786 \text{ mm} = 893 \text{ mm} \tag{4.41}$$

For the z-direction Δz_w should be the difference between the reference point and the wheel centre, not the wheel point touching the ground as the one found in the LiDAR data. The wheel radius, $2\cdot r_w = 16" = 403.4$ mm, is therefore to be included in the calculations:

$$\begin{aligned}\Delta z_w &= z_{P_{ref}} - z_{W_R} + r_w\\ &= (-1275)\text{ mm} - (-1677)\text{ mm} + \frac{1}{2}\cdot 406.4 \text{ mm}\\ &= 605 \text{ mm}\end{aligned} \tag{4.42}$$

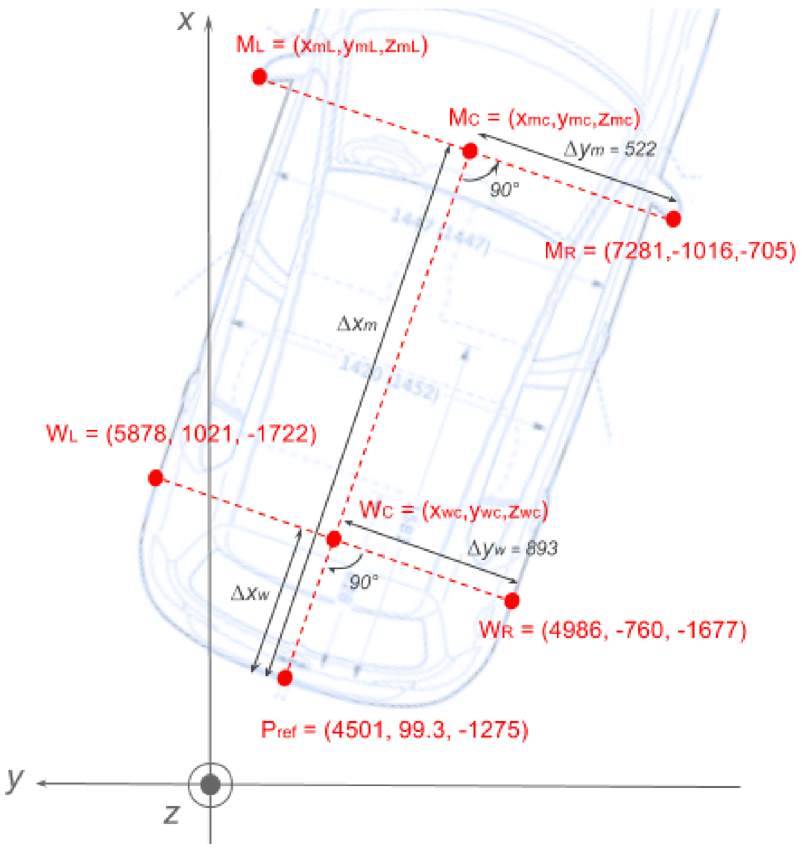


Figure 4.19: *LiDAR measurements of rear wheels, reference point and right mirror. Distances in black are included from the technical documentation of the Golf. All units in mm.*

This gives the wheel centre coordinates relative the reference point as follows:

$$\begin{aligned} W_L{}' &= (\Delta x_w, \Delta y_w, -\Delta z_w) \\ &= (1024, 893, -605) \text{ mm} \qquad (4.43) \\ W_R{}' &= (\Delta x_w, -\Delta y_w, -\Delta z_w) \\ &= (1024, -893, -605) \text{ mm} \qquad (4.44) \end{aligned}$$

Please note these are the rear wheel coordinates! For the front wheels the shaft distance 2575 mm is added in the x-direction, see fig. 4.20 and 4.21.

Mirror coordinates, M_n
For the mirrors only the displacement in x-direction is unknown and need more complex calculations. Δz_m is found directly from the LiDAR data and Δy_m from the technical documentation.

$$\begin{aligned} |W_C\, M_R| &= \sqrt{(x_{M_R} - x_{W_C})^2 + (y_{M_R} - y_{W_C})^2 + (z_{M_R} - z_{W_C})^2} \\ &= \sqrt{(7281 - 5432)^2 + \big((-1016) - 130.5\big)^2 + \big((-705) - (-1699.5)\big)^2} \text{ mm} \\ &= 3215 \text{ mm} \qquad (4.45) \end{aligned}$$

$$
\begin{aligned}
\Delta x_m &= \Delta x_w + \sqrt{|W_C M_R|^2 - \Delta y_m^2} \\
&= 1024 \text{ mm} + \sqrt{(3215 \text{ mm})^2 - (893 \text{ mm})^2} \\
&= 4112 \text{ mm} \qquad (4.46)
\end{aligned}
$$

$$
\begin{aligned}
\Delta y_m &= \frac{1}{2} \cdot 2044 \text{ mm} \\
&= 1022 \text{ mm} \qquad (4.47)
\end{aligned}
$$

$$
\begin{aligned}
\Delta z_m &= z_{M_R} - z_{P_{ref}} \\
&= -705 \text{ mm} - (-1275 \text{ mm}) \\
&= 570 \text{ mm} \qquad (4.48)
\end{aligned}
$$

This result in the following mirror coordinates, relative the reference point:

$$
\begin{aligned}
M_L{}' &= (\Delta x_m, \Delta y_m, \Delta z_m) \\
&= (3215, 1022, 570) \text{ mm} \qquad (4.49) \\
M_R{}' &= (\Delta x_m, -\Delta y_m, \Delta z_m) \\
&= (3215, -1022, 570) \text{ mm} \qquad (4.50)
\end{aligned}
$$

Complete model
A 3D model of the Golf is created by means of the coordinates obtained above and some additional simple calculation based on figures 4.18 and B.1. The results are presented in figures 4.20 and 4.21.

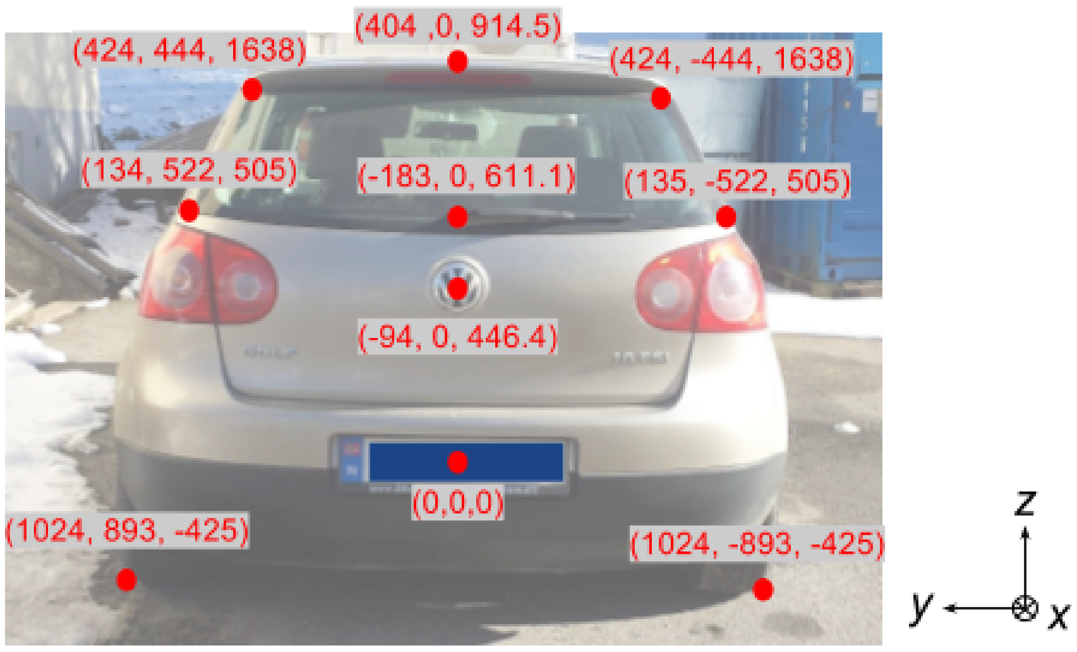


Figure 4.20: *3D model of the Golf. All feature points are relative the centre of the license plate. Units in millimetres.*

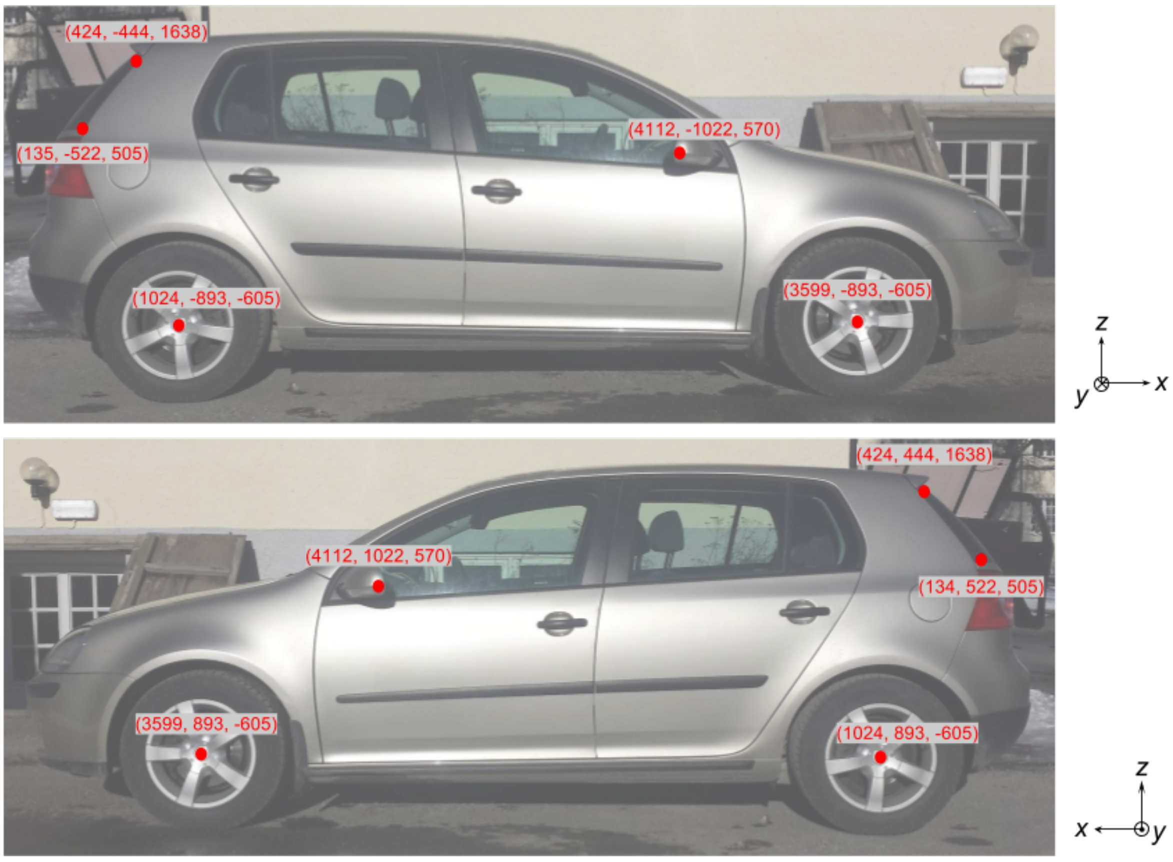


Figure 4.21: *3D model of the Golf. All feature points are relative the centre of the license plate. Units in millimetres.*

4.6.2 License plate

The Norwegian license plate is a rectangular metal or plastic plate with EU standardised dimensions, 520 × 110 mm [30][2]. Assuming no depth variations and origin located at the plate's centre, the four corners have the following coordinates:

Upper left corner	(0, 260, 55)	mm
Upper right corner	(0, -260, 55)	mm
Lower left corner	(0, 260, -55)	mm
Lower right corner	(0, -260, -55)	mm

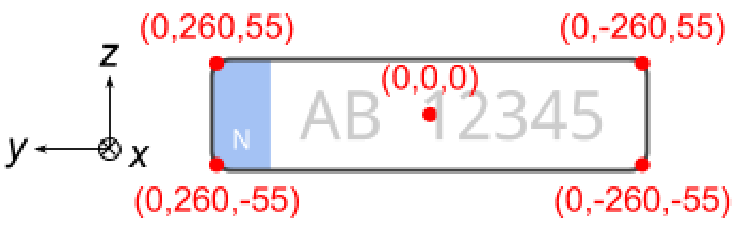


Figure 4.22: *Illustration of a lincense plate and it's corners and center posistions. Coordinate system corresponds to the back plate to a vehicle in front of the detecting camera.*

[2]car plate, narrow version

5. Results

This chapter presents results obtained from applying the methods described in chapter 3 on the described data sets.

5.1 Vehicle detection

This section present some results using the point cloud detection methods on a LiDAR point cloud. The results are presented in the same order as the methods were described in chapter 3.1.2.

Normal estimation and object extraction

Figure 5.1 displays the first three stages of the point cloud detection algorithm. The top shows the original point cloud used as input. The data used is LiDAR message 10 from bag file 2017-02-24-12-36-28_0.bag.

In the middle the estimated normals based on a 3×3 neighbourhood are included. Only every 50th normal is plotted to ensure good visual appearance. Outliers are defined as points with a point-plane distance greater than 50 mm.

Last is the result after removing the ground points. The threshold, τ, is here set to 85°. That implies that the remaining points have estimated surface angles $\alpha \in [85, 90]°$.

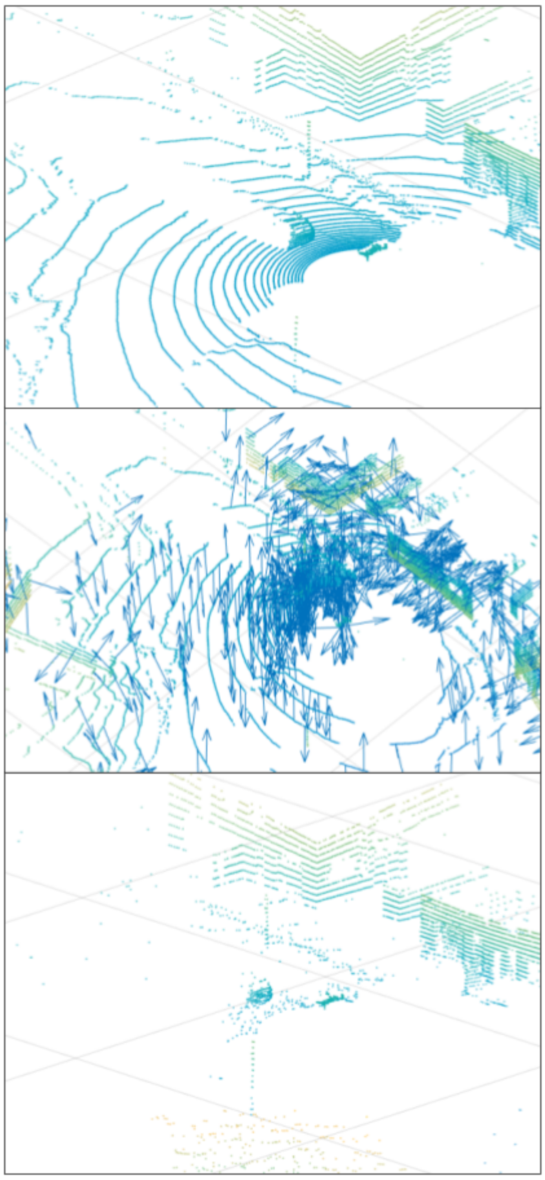


Figure 5.1

Vehicle clusters

In this section the vehicle clusters detected are visualised by a pair of figures. First are intensity images of the point cloud data with a FOV of 120°, and below is the vehicle cluster extracted and presented in a binary image. In total three results are presented.

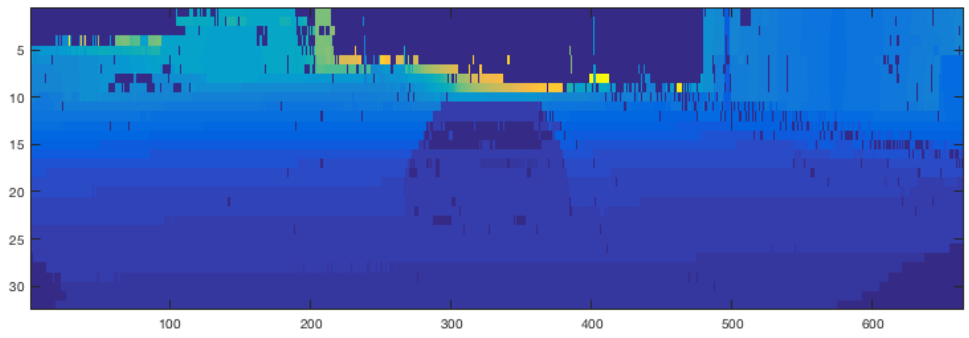


Figure 5.2

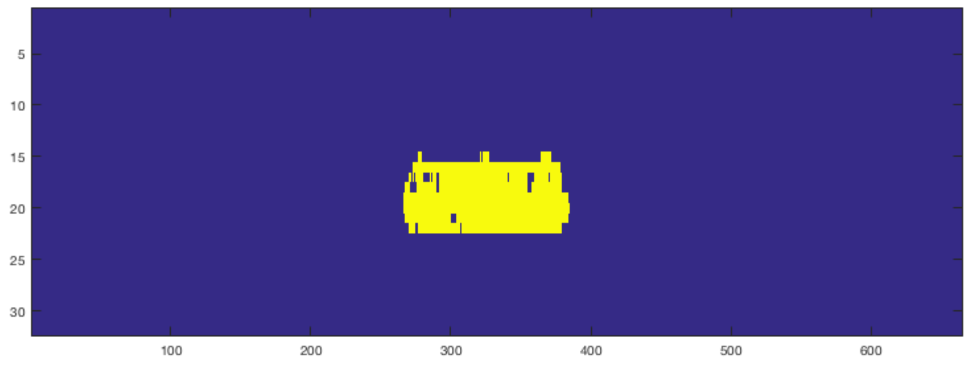


Figure 5.3

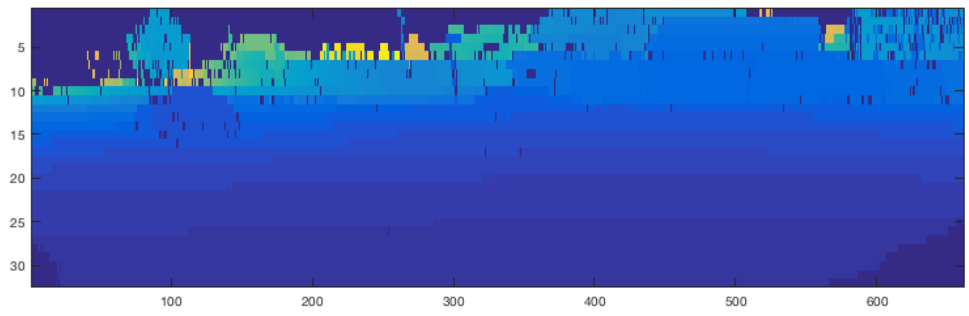


Figure 5.4

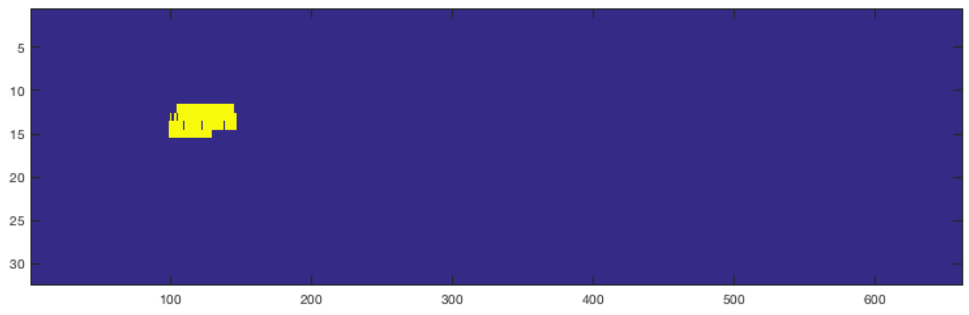


Figure 5.5

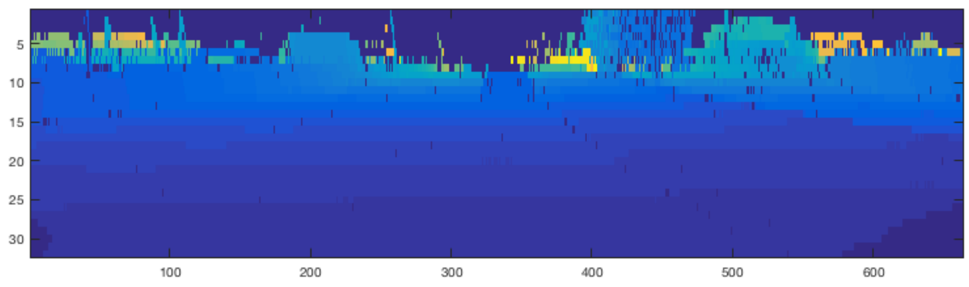


Figure 5.6

5.2 Pose estimation

In the following section the results from the pose estimation methods described in chapter 3 are presented.

5.2.1 About the tests

The POSIT method is the foundation of the estimations. It require minimum 4 nonplanar points and their corresponding 3D properties as input to calculate a proper estimate. For each input image the pose is estimated for the following inputs:

- License plate, 5 feature points (lp)
- Vehicle body, 5 feature points (b5)
- Vehicle body, 9 feature points (b9)

Methods using images as input are performed on both the left and right stereo images. The average of the two corresponding outputs is then computed and set as the final estimate. This is done for two reasons. First, the average of two estimates are less affected by error in one of them. Secondly and most important, to compensate for the ± 15 m left/right offset camera location relative the rooftop centre axis (see figure A.3).

The tests are all performed on the same data sets so that the results are comparable. The bag-file message IDs are listed in table 5.1. LiDAR and image data with equal time stamps are sorted column wise.

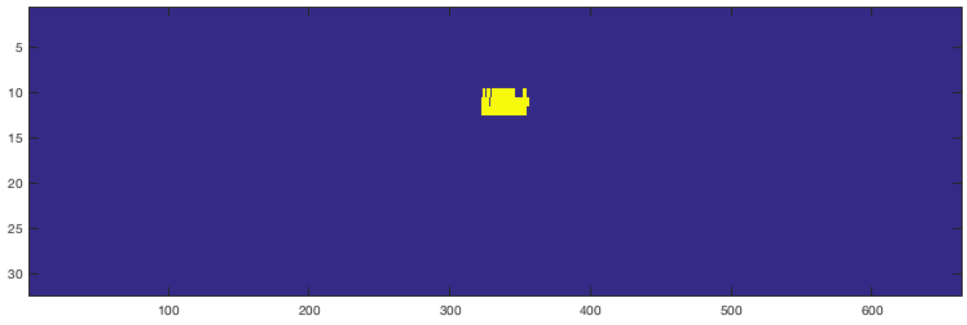


Figure 5.7

i	1	2	3	4	5	6	7	8
Stereo msg. ID	#1	#440	#598	#453	#460	#200	#490	#370
LiDAR msg. ID	#10	#660	#900	#680	#700	#330	#740	#584

Table 5.1: *Data used to test pose estimation methods. Corresponding LiDAR and image message ID in the same column.*

5.2.2 Distance estimates, $\hat{\rho}$

As explained in chapter 3.2.3, two methods for distance estimation are to be tested. First the results using method 1 are presented, followed by the results obtained by method 2. Both methods are applied on the exact same data set so the results are comparable. The distances listed are the radial distance from the lidar/stereo camera center (roof top) to the licence plate centre point of the leading vehicle.

Method 1 - Results

Table 5.2 shows the results using the translation matrix, $\boldsymbol{T}$, to estimate the relative distance, ρ as described in chapter 3.2.3. The normalised standard deviation estimates

i	1	2	3	4	5	6	7	8
ρ [mm]	4 679	11 268	12 459	14 036	15 424	17 057	17 791	20 473
$\hat{\rho}_{lp}$ [mm]	2 703	9 663	14 249	13 725	13 485	13 804	15 855	19 197
$\hat{\rho}_{b5}$ [mm]	4 452	7 999	14 180	12 391	12 958	14 180	15 176	19 065
$\hat{\rho}_{b9}$ [mm]	5 008	7 465	15 848	15 457	15 520	16 482	21 751	22 871

Table 5.2: *Distance estimates of eight stereo image pairs. See table 5.1 for data info.*

of the three different outputs are:

$$\hat{\sigma}_{\rho_{lp}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\rho_i - \hat{\rho}_{lp,i}}{\rho_i}\right)^2} = 18.9\% \tag{5.1}$$

$$\hat{\sigma}_{\rho_{b5}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\rho_i - \hat{\rho}_{b5,i}}{\rho_i}\right)^2} = 15.8\% \tag{5.2}$$

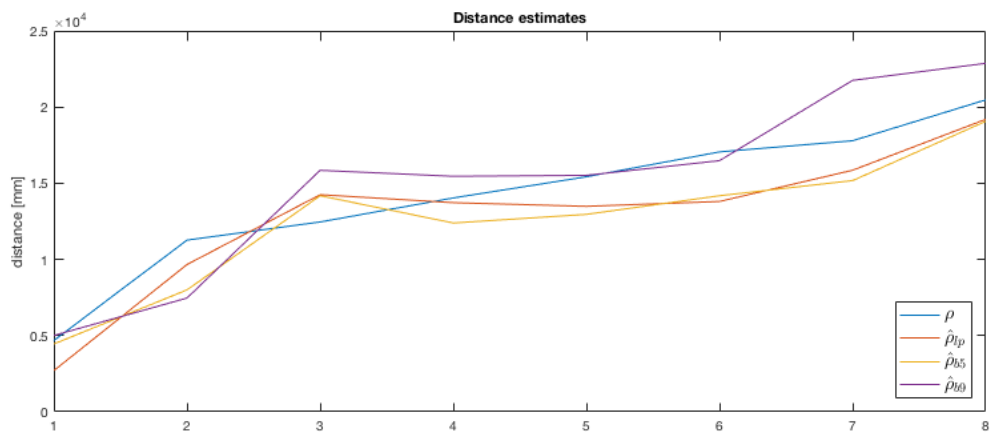


Figure 5.8: *Plot of results given in table 5.2.*

$$\hat{\sigma}_{\rho_{b9}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\rho_i - \hat{\rho}_{b9,i}}{\rho_i}\right)^2} = 18.2\% \tag{5.3}$$

Another relation of interest is to study wheter the deviation increases with the distance to the object, i.e. the distance between the two cars. In figure 5.9 the true relative distance, ρ, is plotted against the mean normalised deviation for each input image.

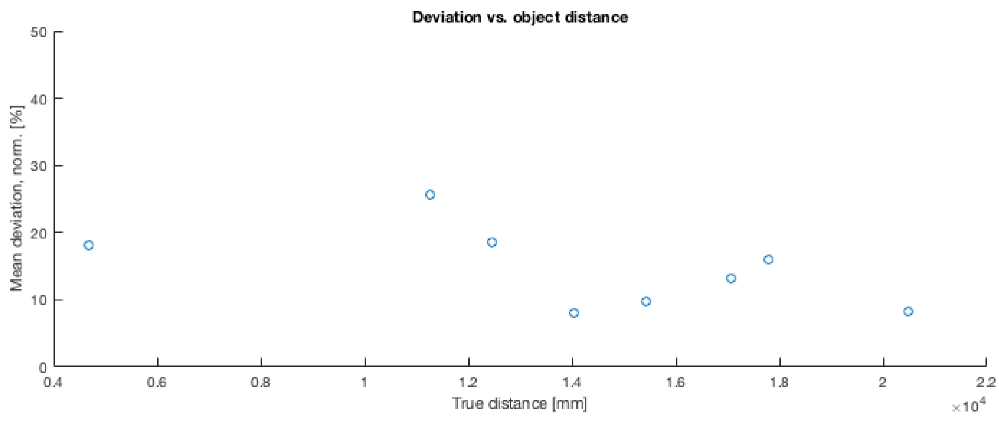


Figure 5.9

Method 2

Table 5.3 displays the distance estimates, $\hat{\rho}_2$. The normalised standard deviation with respect to true distance is calculated. Then two plots of true distance vs. $\hat{\rho}_2$ (fig. 5.10) and normalised deviation vs true relative distance (fig. 5.11).

i	1	2	3	4	5	6	7	8
ρ [mm]	4 679	11 268	12 459	14 036	15 424	17 057	17 791	20 473
$\hat{\rho}_2$ [mm]	2 030	7 711	9 235	5 303	6 376	6 920	8 082	5 537

Table 5.3: *Distance estimates of eight point cloud data sets. See table 5.1 for data info.*

$$\hat{\sigma}_{\rho_2} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\rho_i - \hat{\rho}_{2,i}}{\rho_i}\right)^2} = 54.7\% \tag{5.4}$$

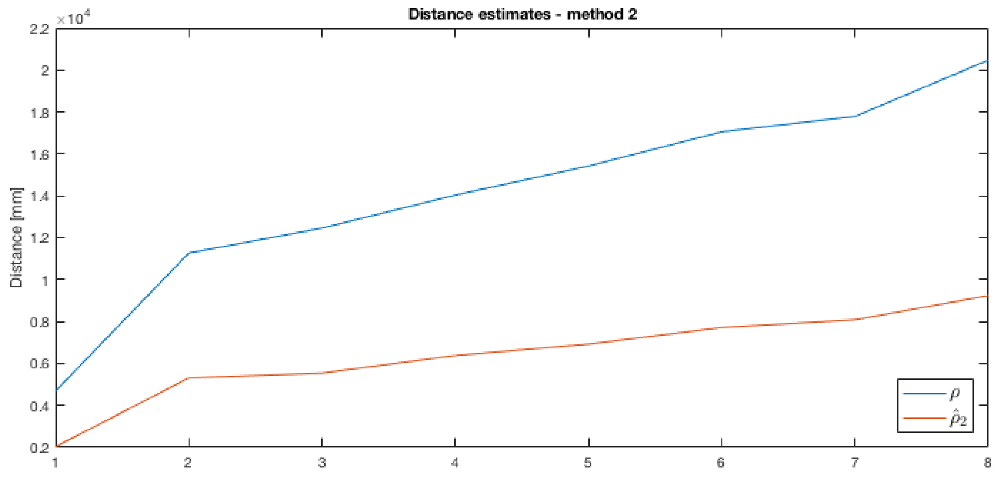


Figure 5.10

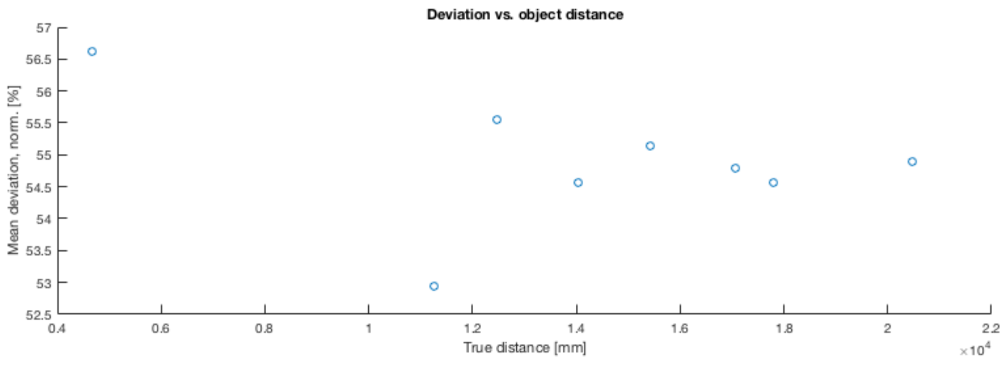


Figure 5.11

5.2.3 Rotation estimates, $\hat{\omega}$

i	3	6	1	5	7	4	8	2
ω	-44.0	-34.0	-5.0	-4.0	13.0	18.0	38.0	66.0
$\hat{\omega}_{lp}$	-0.4	-36.6	7.0	1.7	7.2	2.1	1.6	78.8
$\hat{\omega}_{b5}$	-61.4	-27.6	-8.5	-17.1	33.0	35.1	44.0	40.3
$\hat{\omega}_{b9}$	-35.8	-62.2	-6.2	-11.4	35.5	1.5	57.7	53.1

Table 5.4

$$\hat{\sigma}_{\omega lp} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\omega_i - \hat{\omega}_i)^2} = 21.6^\circ \tag{5.5}$$

$$\hat{\sigma}_{\omega b5} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\omega_i - \hat{\omega}_i)^2} = 15.4° \tag{5.6}$$

$$\hat{\sigma}_{\omega b9} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\omega_i - \hat{\omega}_i)^2} = 16.7° \tag{5.7}$$

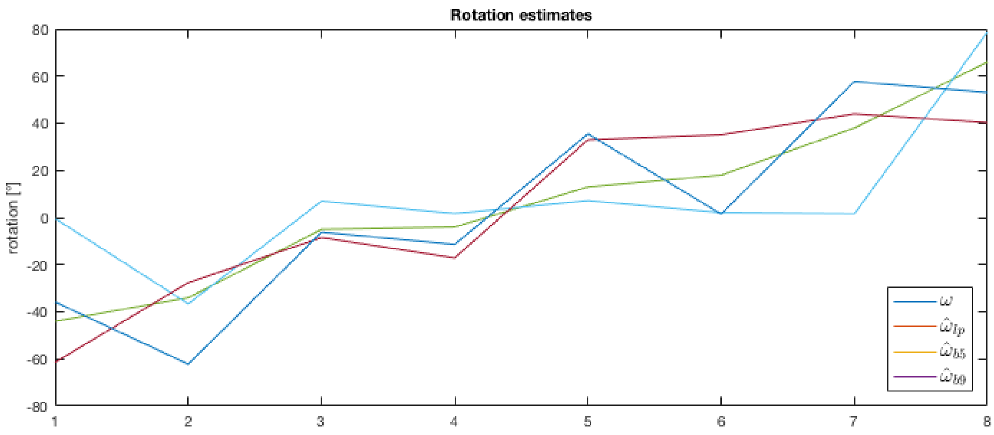


Figure 5.12

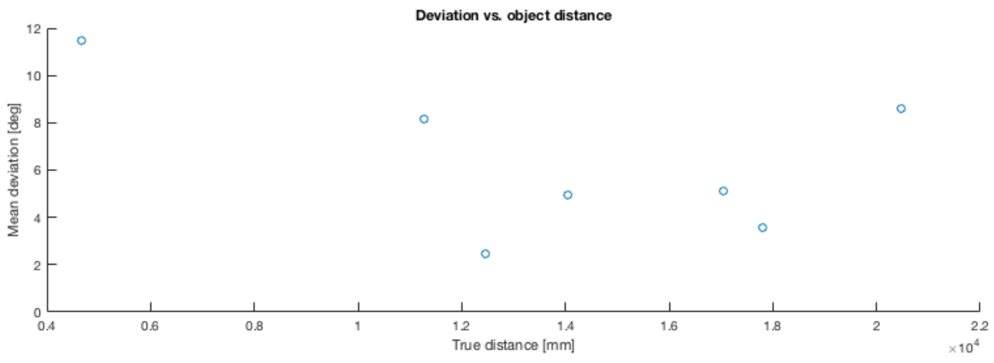


Figure 5.13

5.2.4 Orientation estimates, $\hat{\gamma}$

$$\hat{\gamma}_1 = \tan^{-1}\left(\frac{t_y}{t_x}\right) \tag{5.8}$$

Standard deviation:

$$\hat{\sigma}_{\gamma} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\gamma_i - \hat{\gamma}_i)^2} = 3.85° \tag{5.9}$$

Normalised:

$$\hat{\gamma}_{2n} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\gamma_i - \hat{\gamma}_i}{\gamma_i}\right)^2} = 12.53\% \tag{5.10}$$

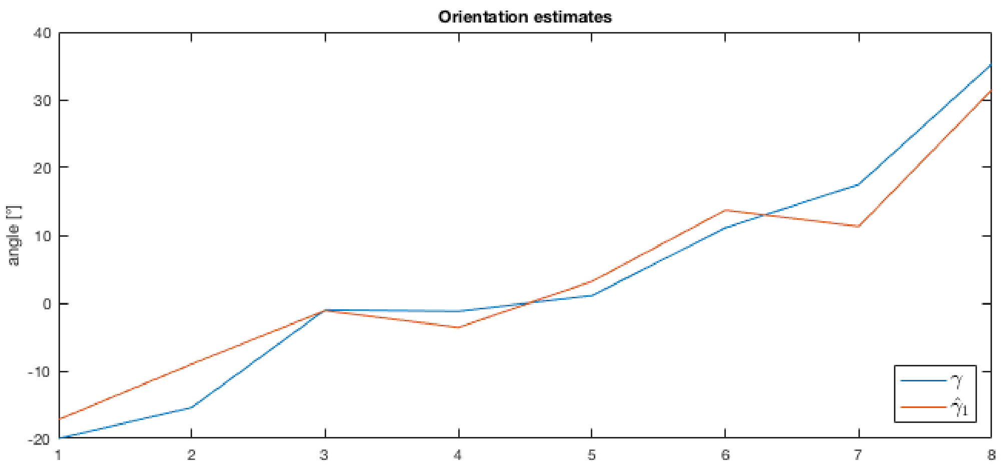


Figure 5.14: *Plot of results given in table* **??**.

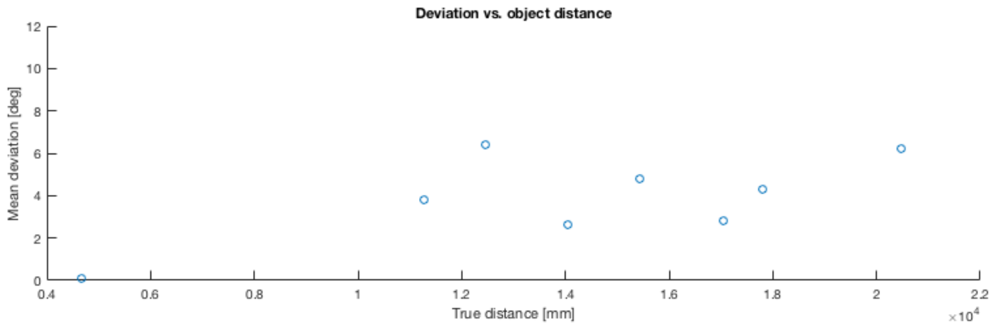


Figure 5.15

6. Discussion

This chapter is used to discussed different sides of this thesis. First some comments on the results are presented, followe

6.1 Comments on the results

A hypothesis was that the POSIT estimates become more accurate with more feature points. Based on the computed deviations for estimates using five and nine feature points this is not considered true. Actually it could look like the incresing number of feature points *reduced* the accuracy.

Another hypothesis that I wanted to study was if there were any correlation between the relative distance and the deviation. From the scatter plots in chapter 5 no such dependencies could be found.

The orientation angle estimates have a high variance, up to 20°, which is considered not satisfying. The distance estimates on the other hand only have an mean normalised error 15%.

There are many sources of error related to the computations. First is the 3D model used which is created using point cloud data from the test runs and the technical documentation of the current vehicle. The accuracy of the model is therefore not known. An improved model could be obtained if LiDAR data was gathered explicitly for this purpose. The more precise estimates, the more accurate model is required.

Another source of error that may affect the results is the accuracy of the data stated as true values. With lack of actual measurements the distance, orientation and rotation values was measured manually. See figure 6.1

6.2 LiDAR range

The LiDAR maps the surroundings by sweeping 32 laser beams with a fixed angular spacing. With increasing distance between the sensor and the object to be detected the resolution decreases with growing relative distance. With a vertical resolution of 1.33°, ref. table 4.2, cosine theorem defines the minimum height of a object to be detected.

$$h_{min} \geq \sqrt{{r_1}^2 + {r_2}^2 - 2r_1r_2\cos(n1.33°)} \tag{6.1}$$

where n is the number of rays hitting the object. To be able to recognize a vehicle n should be greater than three.

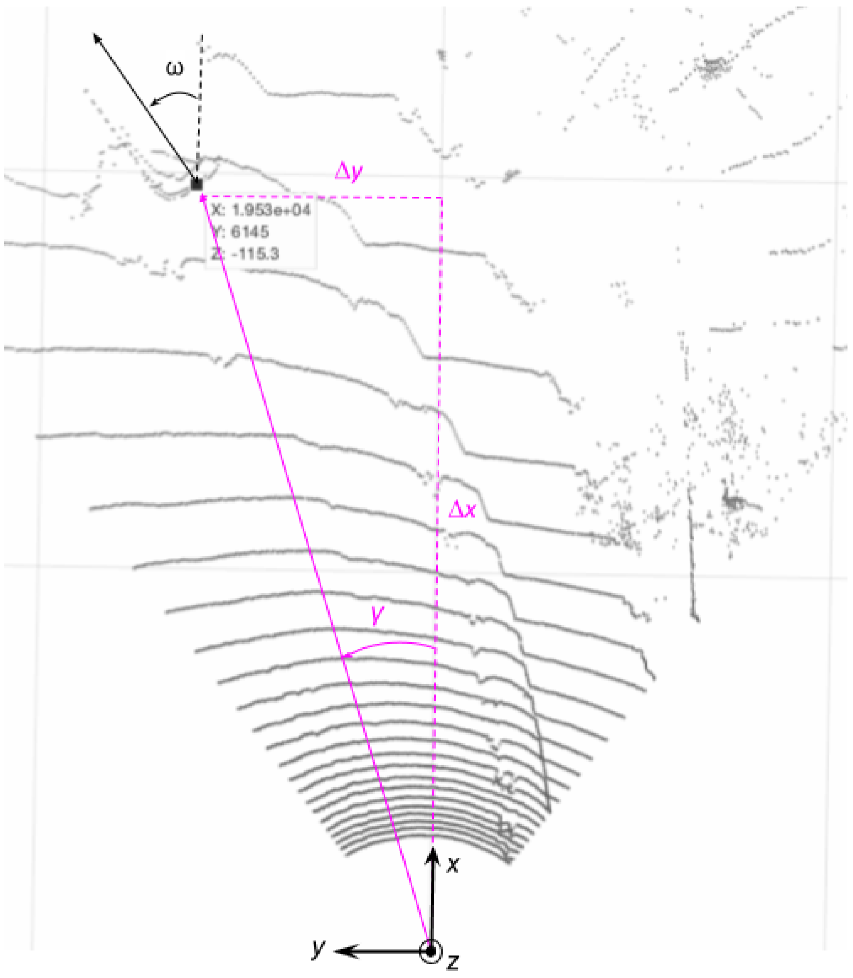


Figure 6.1: *Description of how actual measurements were calculated.*

7. Conclusion and Future Work

In this thesis I have been studying several method for vehicle detection and pose estimation, mainly based on 3D point cloud data. There are obviously advantages using the LiDAR, but it also causes some challenges related to conclude if the object detected is the correct one or only one of similar shape.

Even with lots of potential error sources in the data, like an potential inaccurate 3D model and assumptions an accuracy of 15% was accomplished.

For future work a better 3D model of the leading vehicle is preferrable. Also merging the LiDAR data with camera images to take advantage of image classifiers for improved results should be considered.

A. Vehicle sketches - Ego-vehicle

(Polar Ranger, customized)

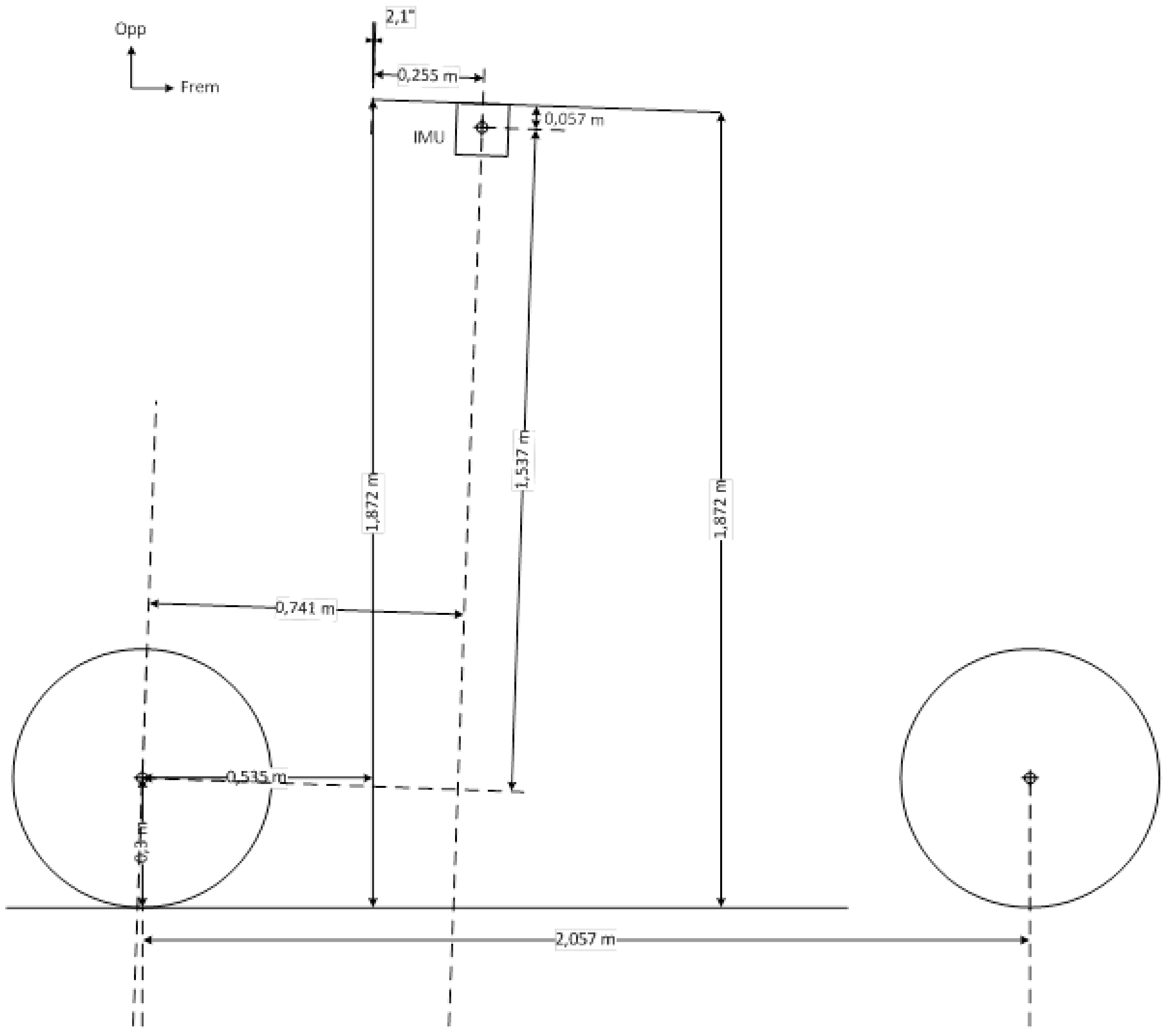


Figure A.1: *Ego-vehicle, right view [7]*

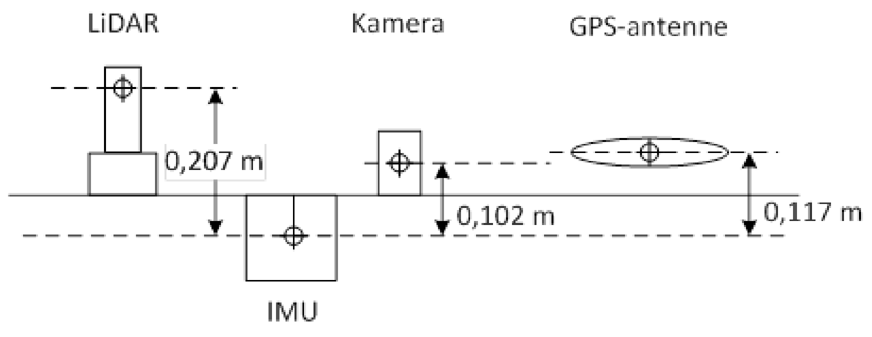


Figure A.2: *Sensor roof top, vertical positions [7]*

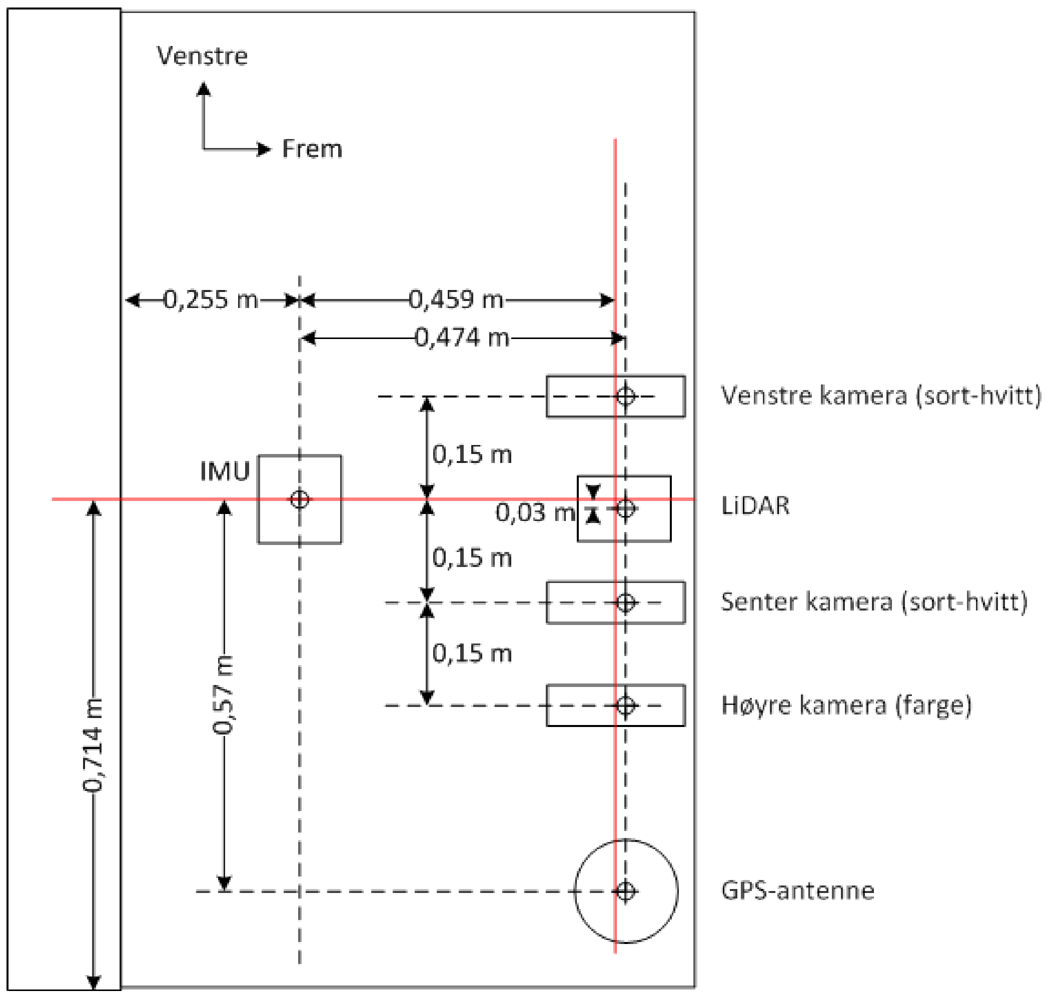


Figure A.3: *Sensor rooftop, top view. Red lines indicate roof top center lines. [7]*

B. Vehicle sketches - Leading vehicle

VW Golf V, mod. 2008. Measurements in millimetres.

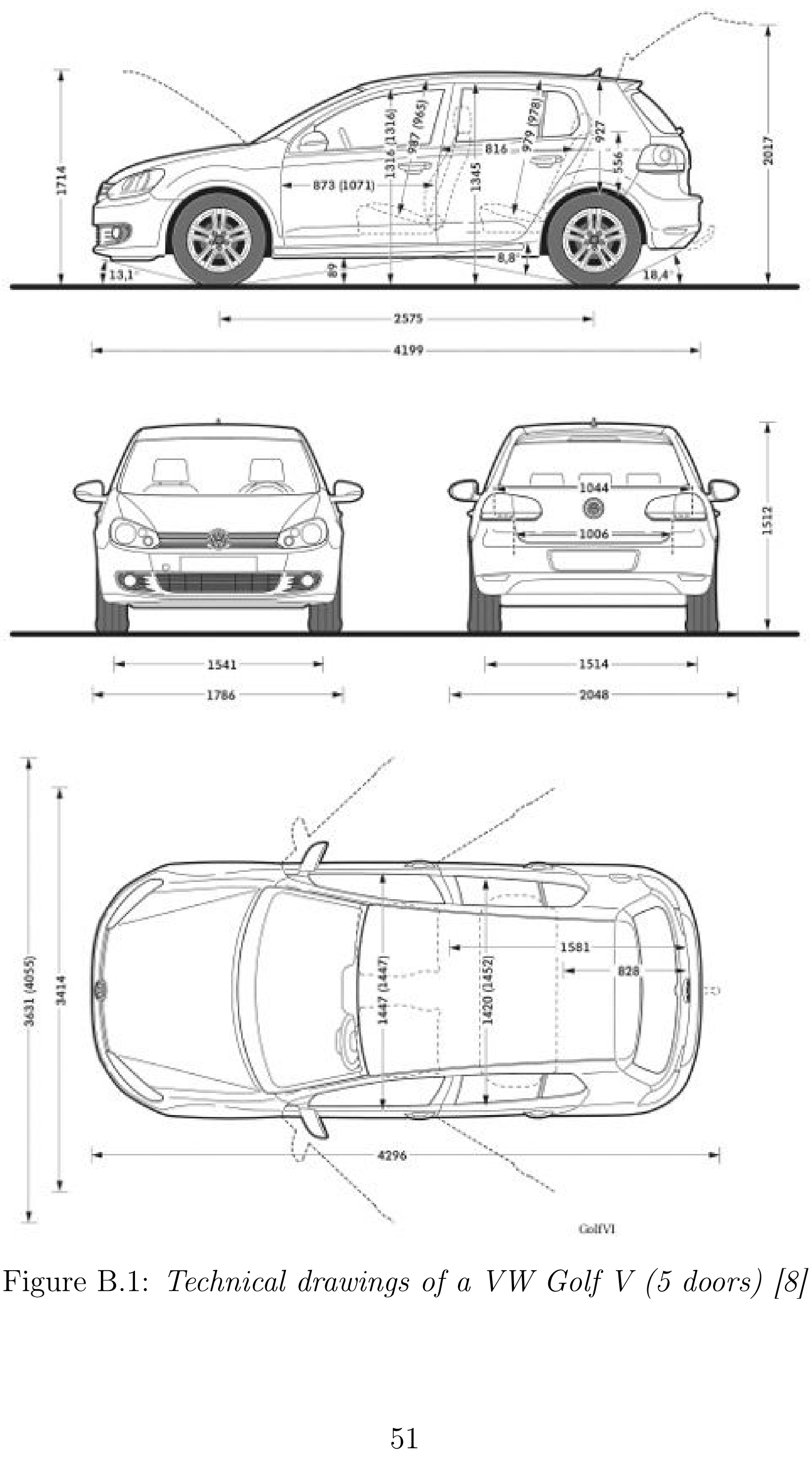


Figure B.1: *Technical drawings of a VW Golf V (5 doors) [8]*

C. Source code

!

. Bibliography

[1] “Dft commision ’platooning’ study.” http://www.roadsafetygb.org.uk/news/3301.html. Accessed: 2017-02-03.

[2] “Lecture notes on imaging geometry.” http://www.cse.psu.edu/~rtc12/CSE486/lecture12.pdf. Accessed: 2017-06-05.

[3] “Wikipedia - random samle consensus.” https://en.wikipedia.org/wiki/Random_sample_consensus. Accessed: 2017-06-07.

[4] http://polarisindustries.no/atv-ranger/. Accessed: 2017-03-06.

[5] E. V. B. Baardseth.

[6] “Polaris.com: Screenshot of 360° overview.” http://www.polaris.com/en-us/ranger-utv/ranger-570-efi-sage-green. Accessed: 2017-05-18.

[7] M. Baksaas.

[8] “Vw golf v - technical documentation.”

[9] “Manual - velodyne hdl-32e lidar.” http://velodynelidar.com/hdl-32e.html. Accessed: 2017-01-27.

[10] P. Kavathekar and Y. Chen, “Vehicle platooning: A brief survey and categorization,” *Proceedings of the ASME 2011 IDETC/CIE*, pp. 1–17, 2011.

[11] “Ffi homepage.” http://www.ffi.no/en/About-FFI/. Accessed: 2017-02-02.

[12] M. F. Levedahl, A. and G. Mouzakitis, “Platooning dynamics and control on an intelligent vehicle transport system,”

[13] “Wikipedia: Platoon (automobile).” https://en.wikipedia.org/wiki/Platoon_(automobile)#How_it_works. Accessed: 2017-06-10.

[14] “Eu truck platooning - homepage.” https://eutruckplatooning.com/. Accessed: 2017-03-12.

[15] “The verge - article on successful eu plaoon.” https://www.theverge.com/2016/4/7/11383392/self-driving-truck-platooning-europe. Accessed: 2017-03-12.

[16] S. M. R. P. T. R. P. e. a. Kelly, A., “Toward reliable off road autonomous vehicles operating in challenging environments,” *The International Journal of Robotics Research*, vol. 25, pp. 449–483, 2006.

[17] M. M. Lüttel, T. and H. Wünsche, "Unbemanntes konvoi-fahren auf und abseits von wegen," *Wehrtechnischer Report*, vol. 7, pp. 79–81, 2011.

[18] H. H. L. T. Manz, M. and H. Wünsche, "Fusing lidar and vision for autonomous dirt road following,"

[19] L. T. Fries, C. and H. Wünsche, "Combining model- and template-based vehicle tracking for autonomous convoy driving," *IEEE Intelligent Vehicles Symposium (IV)*, 2013.

[20] C. D. Zhang, F. and A. Knoll, "Vehicle detection based on lidar and camera fusion," *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1620–1625, 2014.

[21] D. F. DeMenthon and L. S. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision*, vol. 15, pp. 123–141, June 1995.

[22] "Aforge.net: 3d pose estimation." `http://www.aforgenet.com/articles/posit/`. Accessed: 2017-05-20.

[23] "Uio lecture notes on ransac." `http://www.uio.no/studier/emner/matnat/its/UNIK4690/v16/forelesninger/lecture_3_3-robust-estimation-with-ransac.pdf`. Accessed: 2017-06-07.

[24] C. Fries and H. Wünsche, "Monocular template-based vehicle tracking for autonomous convoy driving," *IEEE/RJS Intelligent Robots and Systems (IROS)*, 2014.

[25] K. J. Steinemann, P. and J. e. a. Dickmann, "Determining the outline contour of vehicles in 3d-lidar-measurements," *IEEE Intelligent Vehicle Symposium (IV)*, pp. 479–484, 2011.

[26] P. O. Moosman, F. and C. Stiller, "Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion," *IEEE Intelligent Vehicle Symposium*, pp. 215–220, 2009.

[27] "Manual - flir grasshopper3 usb camera color." `https://eu.ptgrey.com/grasshopper3-89-mp-color-usb3-vision-sony-pregius-imx255-2`. Accessed: 2017-01-27.

[28] "Manual - flir grasshopper3 usb camera monochrome." `https://eu.ptgrey.com/grasshopper3-89-mp-mono-usb3-vision-sony-pregius-imx255-2`. Accessed: 2017-01-27.

[29] "Ros wiki - documentation." `http://wiki.ros.org`. Accessed: 2017-02-27.

[30] "Skiltfarger og -størrelser, statens vegvesen." `http://www.vegvesen.no/kjoretoy/Eie+og+vedlikeholde/skilt/bestille-skilt/skiltfarger-og-storrelser`. Accessed: 2017-05-23.