



Master's Thesis

Autonomous driving in urban environments, roundabouts

Julian-B. Scholle
March 8, 2017

Contents

1	Introduction	4
1.1	Motivation	4
1.2	Research Question	4
2	Related Work	5
2.1	Autonomous Driving	5
2.2	Roundabouts in Law	5
2.2.1	Elements of a Roundabout	5
2.2.2	Types of Roundabouts	6
2.3	Middleware OpenDAVINCI	8
2.4	Hardware Platform	8
3	Methodology	9
3.1	Selection of Sensors	9
3.2	Selection of Algorithms	9
3.3	Simulation Enviroment	9
4	Research	10
4.1	Objekt Detection	10
4.1.1	Ground Removal	10
4.1.2	Clustering	13
4.1.3	Tracking	15
4.1.4	Classification	17
4.1.5	State Estimation	18
4.2	Mapping	18
4.2.1	OpenDAVINIC Map	18
4.2.2	Simplification	18

5	Evaluation	19
5.1	Simulation	19
5.2	Real Measurements	19
6	Conclusions	20
7	Future Work	21

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig 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 Prüfungsamt vorgelegt und auch nicht veröffentlicht.

Göteborg, den March 8, 2017

Julian-B. Scholle

Das autonome Fahren und die Vernetzung von Fahrzeugen mit Ihrer Umwelt sind zusammen mit der Elektromobilität die meistdiskutierten Themen der Automobilbranche. Zu Recht: Autonomes Fahren besitzt das Potenzial, im Mobilitätsmarkt völlig neue Strukturen entstehen zu lassen.

1.1 Motivation

Da die Technische Hochschule Chalmers ergänzend zu Volvos “DriveMe” Projekt das Projekt “CampusShuttle” initiiert, “CampusShuttle” ist ein interdisziplinäres Forschungsprojekt der Technischen Hochschule Chalmers und der Universität Göteborg.

Das Projekt ist dabei im ReVeRe (Chalmers Research Vehicle Resource) angesiedelt. Die Vision ist dabei ein selbstfahrendes Auto zwischen den beiden Campus der Technische Hochschule Chalmers.

Dabei soll, im Rahmen des Projekts, das Fahrzeug in verschiedenen Verkehrsszenarien untersucht werden. Der Fokus liegt dabei besonders auf den Stadtverkehr, das Fahrzeug muss dabei nicht nur in der Lage sein mit anderen Autos zu interagieren, sondern ebenfalls mit Straßenbahnen, Bussen, Fahrrädern und Fußgängern sicher agieren.

1.2 Research Question

2.1 Autonomous Driving

2.2 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 [RAL] and the Directives for the Design of Urban Roads [RASt]. 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 [RASt] are relevant for this dispute. Since the access the RASt ist limited, most of the information is coming from [7] whereupon RASt is based on.

2.2.1 Elements of a Roundabout

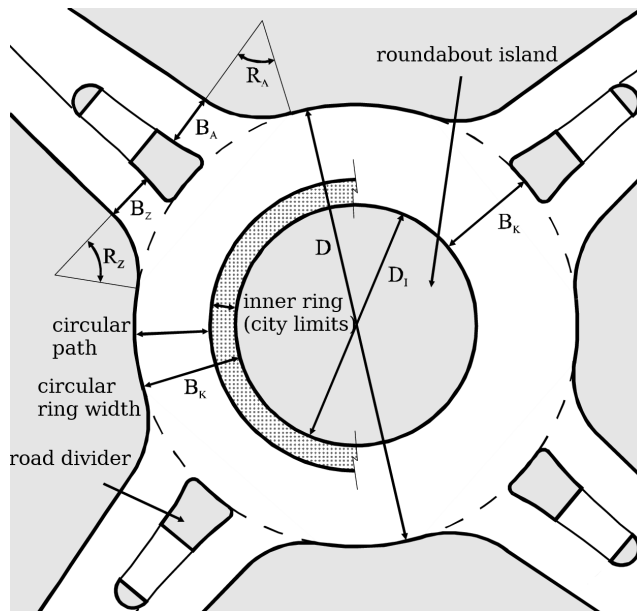
Definition 2.1 (roundabout island) *The roundabout island is the constructional area in the middle of the roundabout, which is surrounded by vehicles. For miniature roundabouts, the roundabout island is crossable. [7]*

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). [7]*

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 routing (one or two lanes). The edge strip width is oriented on the relevant continuous roadway. [7]*

Definition 2.4 (outer diameter (D)) *The outer diameter is measured at the*

Figure 2.1: Definition of individual design elements and dimensions of a roundabout [7]



outer edge of the circular ring. It is the essential measure for describing the size of the roundabout. [7]

Definition 2.5 (inner diameter (D_1)) The inner diameter is the diameter of the roundabout island. [7]

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. [7]

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. [7]

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. [7]

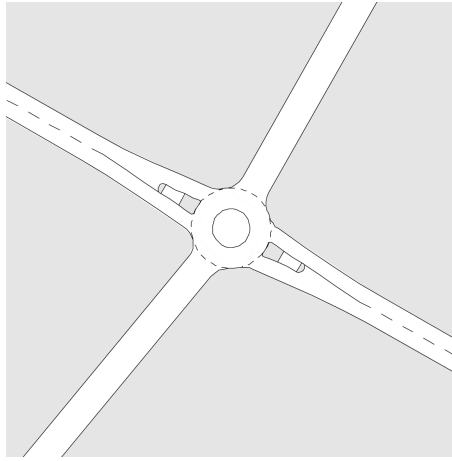
2.2.2 Types of Roundabouts

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. [7]

Mini Roundabout

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

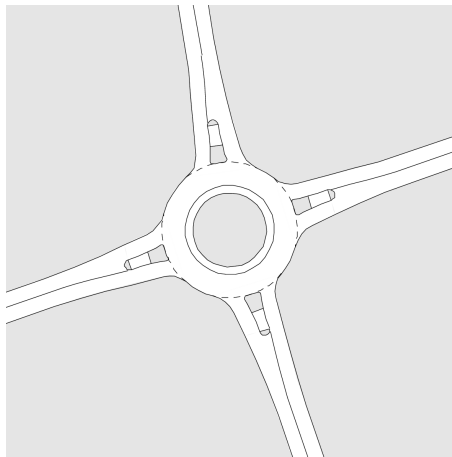
Figure 2.2: Mini Roundabout [7]



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 2.3: Small Roundabout [7]



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

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.

Figure 2.4: Two-lane Passable Roundabout [7]

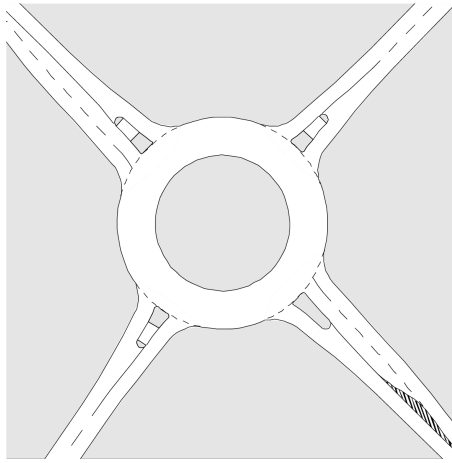
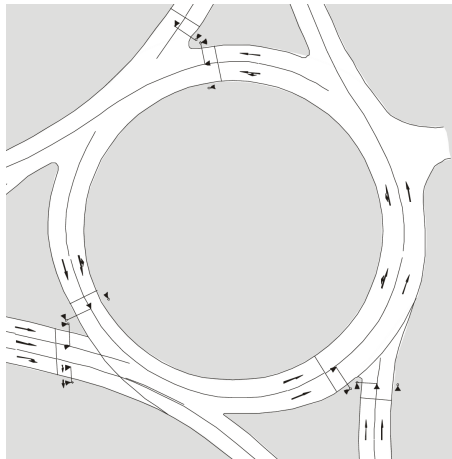


Figure 2.5: Large Roundabout [7]



Large Roundabout

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.

2.3 Middleware OpenDAVINCI

2.4 Hardware Platform

Gefragt wird hier nach den Kriterien dafür, welche Methode für eine bestimmte Art der Anwendung geeignet ist, warum eine bestimmte Methode angewandt werden muss oder angewendet wird und keine andere. Verständnisfragen zum methodischen Weg werden hier geklärt. Die Methodologie ist demnach eine Metawissenschaft und somit eine Teildisziplin der Wissenschaftstheorie. Demgegenüber bezeichnet Methodik das Methodenwissen des Praktikers oder des Wissenschaftlers.

3.1 Selection of Sensors

3.2 Selection of Algorithms

3.3 Simulation Environment

Klare, logische Gliederung

Möglichst ausgeglichene Kapitel (bezüglich Umfang und Zahl der Unterkapitel)

Gesamte Arbeit enthält so wenig Redundanz wie möglich

Ist auch innerhalb der einzelner Kapitel oder Abschnitte sinnvoll strukturiert

Kapitel und Unterkapitel beginnen stets mit einer ganz kurzen Einleitung (in der Regel 1-3 Sätze, die erklären, was im Folgenden zu erwarten ist)

Kurze, aussagekräftige Überschriften in einheitlichem Stil

Beschreibung von Konzepten. Technische Details, wie z.B. Quellcode, umfangreiche Auflistungen, ergänzende Abbildungen usw. kommen in den Anhang.

4.1 Objekt Detection

4.1.1 Ground Removal

Um in einer PointCloud Objekte zu erkennen ist es nötig, zu wissen, welche Messungen zu Boden und welche zu Objekten gehören. Es gibt viele Möglichkeiten dieses zu erreichen. Die Naivste Methode ist das Entfernen der Bodenplatte anhand ihrer Z-Koordinate. Diese Methode hat allerdings viele Nachteile, zum einen muss der LIDAR Sensor exakt gerade auf ein Fahrzeug angebracht werden, zum anderen muss das Fahrzeug ein sehr steifes Fahrwerk haben, um eventuelle Neigungen des Sensors zu verhindern. Weiterhin erlaubt dies ausschließlich die Entfernung von planaren Grundflächen, also flache nicht hügelige Untergründe. Eine weitere verbreitete Methode ist das Entfernen der Bodenplatte auf Basis eines statistischen Mittelwertes [9]. Diese Methode benötigt allerdings auch eine Kalibrierung der Sensorabstände zum Boden. Und die

Bestimmung weiterer Schwellwerte, welche umgebungsabhängig sind. Die Vorteile beider Methoden sind ihre geringe Rechenleistung und Laufzeit $O(n)$. Bessere Methoden wie Gradientenbasierende Expansionsalgorithmen benötigen einen Startpunkt, der als Bodenplatte identifiziert werden kann. Eine weitere Möglichkeit ist die Beschreibung von Objekten als konvexe Objekte [6], die ebenfalls auf Basis der Gradienten beschrieben werden kann. Vorteil dieser Methode ist, dass keine initiale Position für die Bodenplatte benötigt wird.

Für unseren Anwendungsfall mit dem Velodyne VLP-16 besteht das Problem darin, dass die Auflösung des Sensors in der Höhe sehr gering ist. Abhängig von der Entfernung des Fahrzeuges innerhalb der benötigten Reichweite fallen nur zwei Lagen auf die Testfahrzeuge, weshalb Gradientenbasierende Methoden hier zuverlässig versagen. Da die Gradienten zu klein sind und die Verkeilerung der nötigen Thresholds zu häufigen falschen Positiven führt. Die Methode des statistischen Mittelwertes und die Methode auf Basis der Z-Koordinate, leider am Fahrwerk des Volvo XC90 SUV. Die Höhe des Fahrzeuges ändert sich außerdem durch Veränderung des Fahrprofils (Sport/Eco, etc.) um mehrere Zentimeter. Auch leicht erhöhte Geschwindigkeiten im Kreisverkehr (ca 30 km/h) führen zu einer deutlichen Seitenneigung des Fahrzeuges. Darum wird nun eine weitere Methode vorgeschlagen. Die Erkennung einer Grundfläche in den Messdaten.

Für die Erkennung des Bodens gehen wir von folgenden Annahmen aus, die Straße lässt sich approximativ als Ebene im R3 darstellen. Weiterhin ist die Grundfläche die niedrigste Fläche im Koordinatensystem. Daher wird im ersten Schritt der in Polarkoordinaten vorliegende Datensatz in mehrere Tortenstückförmige Segmente geteilt. Innerhalb dieser Tortenstücke wird dann eine Suche nach den 10 Messungen mit dem niedrigsten Z-Wert gesucht. Die Einteilung in Segmente ist deshalb nötig, um zu verhindern, dass alle Messwerte in ein einziges lokales Minimum laufen. Aus diesen vorgefilterten Messwerten werden nun für einen RANSAC drei zufällige, jedoch nicht in benachbarten Segmenten liegende Messwerte herausgesucht. Aus diesen drei Punkten wird nun eine Ebene in der Hesseschen Normalform gebildet, was eine effiziente Distanzberechnung zu anderen Punkten erlaubt. Danach sammeln wir alle weiteren Punkte aus unseren Minima, anhand eines Distanzkriteriums. Danach wird aus der Ebene und den neu gesammelten Punkten durch einen Plane-fitting Algorithmus [section 4.1.1] eine neue Ebene und deren Fehler berechnet. Der Fehler wird über die Summe der quadratischen Abstände aller Punkte zur Ebene berechnet.

Bevor wir die Ebene jedoch als eventueller Lösungskandidat hinzufügen wird geprüft, ob sich die Ebene innerhalb von einem plausiblen Parameterbereich befindet. Dazu zählt, dass die Entfernung der Ebene zwischen 1.9m und 2.2m bewegen sollte, dies entspricht in etwa der Montagehöhe des Velodyne-Sensors.

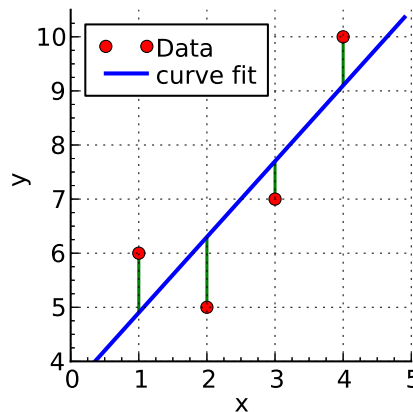
Die Anzahl an Iterationen des RANSAC ist auf 50 begrenzt. Nach dem Durchlauf des RANSAC werden alle Punkte in der Pointcloud anhand ihrer Distanz zur Ebene als Grundfläche markiert. Als Threshold wurde hier ein Wert von 0.5m genommen.

Plane-fitting

Zum Plane-fitting einer Ebene wird üblicherweise eine SVD (Singular Value Decomposition) genutzt. SVD hat eine Komplexität von $\mathcal{O}(\min\{mn^2, m^2n\})$ [3], da das Plane-fitting innerhalb des RANSAC's sehr häufig mit einer großen

Anzahl an Punkten ausgeführt wird, führt das Ausführen des SVD innerhalb des RANSAC zu einer sehr hohen Laufzeit. Deshalb wird an dieser Stelle ein "Linear least Squares (LLSQ)" Algorithmus mit einigen Optimierungen eingesetzt. Bei der Verwendung des LLSQ gilt es zu beachten, dass nicht der Abstand der Punkte zur Ebene optimiert wird, sondern der Abstand der Punkte zur Ebene entlang einer Achse (in unserem Fall der z-Achse) siehe fig. 4.1. Das kann zu Problemen führen, wenn die Punkte weit gestreut, also weit von der optimalen Ebene entfernt sind. Da wir unsere Punkte innerhalb des RANSAC allerdings anhand eines Distanzkriteriums vorselektieren, stellt dies kein Problem dar.

Figure 4.1: Linear least Squares (LLSQ) [8]



Die Darstellung einer Ebene in Koordinatenform sieht wie folgt aus: $a\vec{x} + b\vec{y} + c\vec{z} + d = 0$. Da wir eine Ebene im \mathbb{R}^3 betrachten, ist dieses Gleichungssystem überbestimmt. Da wir unsere Ebene in Richtung der z-Achse optimieren wollen, setzen wir Parameter c auf 1 und können unser Gleichungssystem nun einfach nach z auflösen: $a\vec{x} + b\vec{y} + d = -\vec{z}$. Die Vektoren $\vec{x}, \vec{y}, \vec{z}$ stellen dabei die zu fittenden Punkte dar. In Matrixschreibweise:

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

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

Dieses System hat üblicherweise keine Lösung, unser eigentliches Ziel ist jedoch auch nicht exakte Lösungen für $\vec{\beta}$ zu finden, sondern eine gute Näherung $\hat{\beta}$ dafür:

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

Das können wir tun, indem wir unsere Gleichung mit der Transponierten unserer Punktmatrix X multiplizieren:

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

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

Dieses Gleichungssystem könne man nun mit der Berechnung der Inverse von $(X^T X)$ auflösen. Da die Berechnung von Inversematritzen mit $\mathcal{O}(n^3)$ ebenfalls aufwändig ist, nun ein weiterer Trick um rechenleistung zu sparen. Nach dem Multiplizieren der Transponierten erhalten wir:

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

Gut zu sehen sind hier die Summen in den Randbereichen der Matrix X und dem Vektor \vec{z} . Diese können wir auf Null setzten, wenn wir alle Punkte relativ zum Mittelwert-Punkt aller Punkte definieren, also $P_i = P_i - \bar{P}$. Nun erhalten wir:

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

Nun können wir d ebenfalls auf Null setzten, denn wenn alle unsere Punkte relativ zum Mittelwert-Punkt sind, dann läuft auch unsere Ebene immer durch diesen Punkt. Daher können wir nun eine komplette Dimension streichen:

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

Das Gleichungssystem können wir nun einfach mit der Cramer's rule lösen

$$\begin{aligned} D &= \sum x_i x_i \cdot \sum y_i y_i - \sum x_i y_i \cdot \sum x_i y_i \\ a &= \frac{\sum y_i z_i \cdot \sum x_i y_i - \sum x_i z_i \cdot \sum y_i y_i}{D} \\ b &= \frac{\sum x_i y_i \cdot \sum x_i z_i - \sum x_i x_i \cdot \sum y_i z_i}{D} \\ \vec{n} &= [a, b, 1]^T \end{aligned}$$

Dabei gibt es zu beachten, dass die Determinante nicht Null oder nahe Null sein darf. Da der Winkel zwischen dem Fahrzeug und der Ebene jedoch immer nahe 90 Grad liegt, ist die Determinante typischerweise sehr groß. Sollte die Determinante doch nahe 0 (nicht gleich 0) sein, wird die Berechnung trotzdem durchgeführt, da dies auch zu einem großen Fehler im Fitting führt. Dies ist an dieser Stelle erwünscht, da der RANSAC ungeeignete Ebenen anhand des Fehlers aussortiert. Ist die Determinante exakt Null, wird die Berechnung stattdessen mit einem kleinen Wert für D fortgesetzt.

Aus dem Normalenvektor \vec{n} und dem Mittelwert-Punkt \bar{P} können wir nun wieder die Hessesche Normalenvektor bestimmen.

Letztendlich haben wir so den Algorithmus von $\mathcal{O}(m^2 n)$ auf $\mathcal{O}(n)$ runterbrechen können.

4.1.2 Clustering

In aktuellen Abreiten mit 3D-LIDAR Daten werden die Daten häufig als erstes in eine Heightmap projiziert [9, 2, 5]. Danach werden direkt benachbarte

Messungen mit ähnlichen Messwerten zusammengefasst. Alternativ werden die Messungen auch anhand eines Distanzkriteriums zusammengefasst. Erstere Methode hat den Nachteil, dass einzelne Ausreißer dazu führen, dass das Objekt in mehrere Cluster zerfällt. Letztere wird meist mit einem KD-Tree oder einer ähnlichen Datenstruktur kombiniert, welche typischerweise hohe Kosten für die Erstellung verursachen. Da der Baum nach jeder 360 Gradmessung neu aufgebaut werden muss, ist das problematisch.

Hier wird eine Methode vorgeschlagen, welche die Vorteile beider Methoden kombiniert. Dazu ist es nötig zu wissen, wie die Daten von der OpenDAVINCI Middleware geliefert werden. Das OpenDAVINCI setzt auf der Übertragung der Daten mit UDP Multicast, werden die Daten in einer kompakten Form übertragen, welche in einen einzigen UDP Frame passt.

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

Dabei wird von einer konstanten Drehrate des Sensors ausgegangen, was in einer äquidistanten der Messwerte resultiert. Die Anzahl der Messungen pro Azimuth wird in `entriesPerAzimuth` festgehalten und entspricht für den Velodyne VLP16 16. Um nun an die eigentlichen Messwerte zu kommen, müssen jeweils zwei distance Werte zu einem Unsigned 16Bit Integer umgewandelt werden, welcher dann die Messung in cm enthält. Jeweils 16 dieser Werte ergeben dann einen Messframe, in dem der Polarwinkel auf einen Bereich zwischen -15 und +15 abgebildet werden muss. Nachdem die sphärischen Daten wiederhergestellt wurden, werden diese in Kartesische umgewandelt und in eine Punkt Datenstruktur gespeichert.

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

Diese wird wiederum in ein statisches 2 Dimensionales Array gespeichert: `Points[2000][16]`. Die Reihenfolge der Daten wird dabei beibehalten. Diese Datenstruktur stellt nun im weiteren Verlauf unsere Pointcloud dar.

Auf dieser Basis wird nun ein DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [1] ausgeführt. Der DBSCAN Algorithmus hat dabei folgende Vorteile. Im Gegensatz beispielsweise zum K-Means-Algorithmus, muss nicht im Vorhinein bekannt sein, wie viele Cluster existieren. Der Algorithmus kann Cluster beliebiger Form (z.B. nicht nur kugelförmige) erkennen.

Das macht den DBSCAN damit für uns zu optimalen Kandidaten. DBSCAN selbst ist von linearer Komplexität. Die meiste Rechenzeit wird jedoch üblicherweise durch die Nachbarschaftsberechnung verursacht. Genau hier setzen wir an, anstatt der Bereichsanfrage über eine Baum-Struktur nutzen wir aus, dass Messungen in einer kleinen Nachbarschaft einen ähnlichen Azimutwinkel haben. Dazu untersuchen wir für jeden Messwert jeweils zwei weitere Einträge nach links und rechts in unserem Array. Effektiv müssen wir daher $5 \cdot 16 = 80$ Werte überprüfen. Die Laufzeit der Bereichsanfrage kann deshalb in linearer Komplexität durchgeführt werden. Alle Messwerte, die zuvor als Grund klassifiziert wurden, werden bei der Berechnung übersprungen, zusätzlich entfällt der Aufbau eines KD-Trees.

4.1.3 Tracking

Das Tracking ist in zwei Abschnitte unterteilt. Dem Tracking der Cluster vom DBSCAN und dem Erstellen und Tracken von Hindernissen.

Object Tracking

Für das Tracking der Cluster nehmen wir an, dass sich Objekte von Zeitschritt zu Zeitschritt nur geringfügig bewegen, weiterhin ändert sich die Form der Cluster ebenfalls nur leicht. Das ist wichtig, da die Position eines Clusters durch einen Mittelwertpunkt definiert ist. Das Tracking wird im R2 durchgeführt. Im Initialen Schritt wird jedem Cluster eine aufsteigende ID zugeordnet. In jedem weiteren Schritt wird jedem neuen Cluster die ID des alten Clusters zugeordnet, welcher über die Zeit hinweg die geringste Entfernung aufweist. Für diese Entfernung gibt es eine großzügige obere Schranke von 3m, Cluster, die nicht innerhalb dieser Grenze sind, erhalten eine neue ID. Das führt dazu, dass mehreren Clustern die selbe ID zugeordnet werden kann, das ist wichtig, da Objekte manchmal in mehrere Cluster zerfallen.

Object Tracking

Basis für das Objekt Tracking sind die zuvor getrackten Cluster. Im Initialen Schritt werden aus allen Clustern mit der selben ID Objekte gebildet. In jedem weiteren Schritt werden alle Cluster mit der zuvor gleichen ID zum Update der Objekte genutzt. Aus Clustern mit neuen IDs werden neue Objekte gebildet.

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 Position des Fahrzeuges fehlerbehaftet sind, wird die Bewegungsrichtung nur bei einer minimalen Bewegung von 2m geupdated.

$$\begin{aligned}\Delta x &= P_x(t) - P_x(t-2m) + \Delta C_x \\ \Delta y &= P_y(t) - P_y(t-2m) + \Delta C_y \\ \theta &= \text{atan2}(\Delta y, \Delta x)\end{aligned}$$

mit P - Position des Objekts, C - Position des eigenen Fahrzeuges

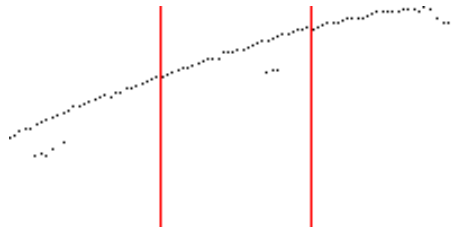
Das Ergebnis kann in fig. 4.2 betrachtet werden. Gut zu erkennen ist, dass die Bewegungsrichtung (Pfeil) nicht mit der Ausrichtung des Objekts (schwarz) übereinstimmt, die Boundingbox jedoch korrekt ausgerichtet ist. Wie diese Berechnung zustande kommt wird im folgenden geklärt.

Figure 4.2: Obstacle Movement



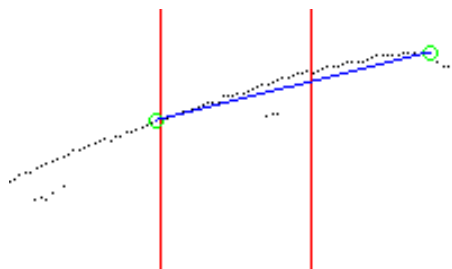
Basierend auf der Bewegungsrichtung wird nun die Ausrichtung des Fahrzeuges berechnet. Dazu werden alle dem Objekt zugewiesenen Cluster zusammengefasst und um $-\theta$ gedreht. Danach wird das Objekt in 3 gleich große Segmente unterteilt (fig. 4.3).

Figure 4.3: Obstacle Cutting



Weiterhin wird bestimmt ob sich das Objekt oberhalb oder unterhalb der x-Achse befindet. Dies ist wichtig, da wir wissen müssen, welche Seite des Objekts wir messen. Befindet sich das Objekt also unterhalb der x-Achse wird im nächsten Schritt der y-Wert maximiert, befindet es sich oberhalb, wird er minimiert. Im folgenden gehen wir davon aus, dass sich das Objekt unter der x-Achse befindet. Deshalb maximieren wir nun im linken und rechten Segment des geteilten Hindernisses die y-Werte. Die Unterteilung in 3 Segmente ist nötig um zu verhindern, dass bei perfekt wagerechte ausgerichteten Objekt beide Maxima in den selben Punkt laufen. Mit diesen Punkten ($\vec{R}; \vec{L}$) wird nun eine Korrektur der Drehung des Objektes berechnet:

Figure 4.4: θ - Correction



$$\Delta x = R_x - L_x$$

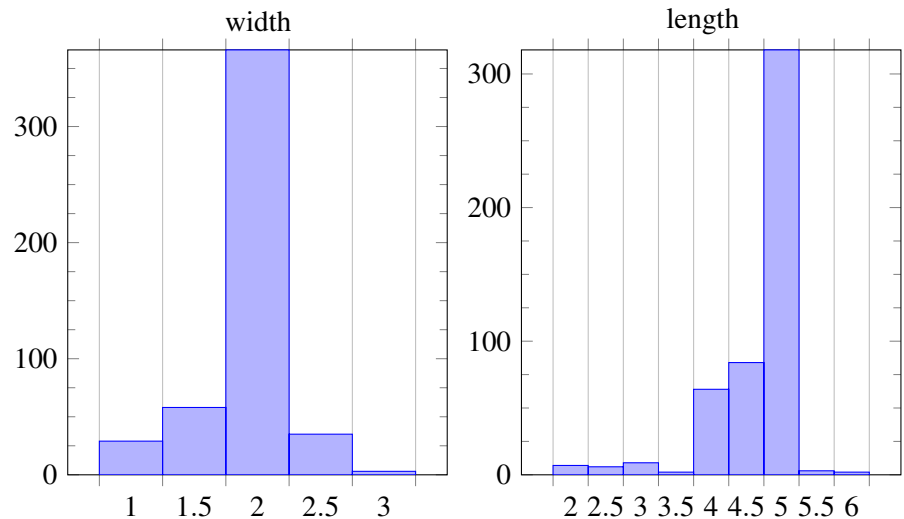
$$\Delta y = R_y - L_y$$

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

Nach Anwendung der Korrektur wird die Größe des Hindernisses berechnet. Dazu werden die maximalen und minimalen x und y Werte herangezogen. Mit diesen Werten wird nun über die Zeit ein auf 0.5m gerundetes Histogramm

für die Länge und Breite des Hindernisses aufgebaut. Anhand diesem wird dann der warscheinlichste Wert ausgewählt. Dadurch ändert sich die Größe des Hindernisses zu begin häufiger, bevor die Größe auf einen stabilen wert konvergiert. Da die größe des Objekts zu begin sehr klein sein kann, gibt es für beide Werte einen unteren Grenzwert. Messerte für ein Beispielobjekt sind in fig. 4.5 zu sehen.

Figure 4.5: Object Size Histogram



Leicht zu sehen wird für das Object eine breite von 2m und eine Länge von 5m berechnet. Das Objekt ist in diesem Fall ein Volvo S60, welcher Außenmaße von ca. 1.9m und 4.6m hat, womit die Abweichungen sich korrekt innerhalb der Rundung der Werte befinden. Für die nachfolgende Filterung der Messwerte mit Hilfe eines Kalman Filters, wurde nun die Position des Fahrzeuges aus dem Mittelpunkt der Boundingbox bestimmt.

Die für die Berechnung der Bewegungsrichtung genutzte Position ist jedoch eine andere, da die so eben berechnete Position zu diesem Zeitpunkt noch nicht zur Verfügung steht und die Position kurz nach der Initialen Erkennung durch die häufigen Größenänderungen sehr instabil ist. Deswegen wird als Position immer die maximale x-Koordinate der Cluster genutzt. Da θ im initialen Zeitschritt Null ist, entspricht dies der globalen Maximalen x-Koordinate des Clusters. Dies führt dazu, dass wir im initialen Zeitschritt annehmen, dass sich das Objekt in die positive x-Richtung bewegt. Bei Objekten bei denen das nicht der Fall ist, führt das zu einem kurzzeitigen oszillieren der Orientierung, welche sich jedoch über die Zeit schnell stabilisiert.

4.1.4 Classification

Die Klassifizierung der Objekte findet anhand ihrer Größe statt, es findet keine Klassifizierung nach bewegten und unbewegten Objekten statt. Unterschieden werden lediglich Fußgänger, Radfahrer, und Fahrzeuge, und Sonstige. Als Klassifizierungskriterien werden dabei unter anderem, die Größe der Cluster und ihre Geschwindigkeit genutzt. Objekte die eine Größe von 2×1.5 m überschreiten werden dabei als Fahrzeug klassifiziert, Objekte die kleiner als 1×1 m sind werden als Fußgänger klassifiziert. Alles dazwischen wird als Radfahrer klassifiziert. Weiterhin wird anhand der Geschwindigkeit ein Plausibilitätstest durchgeführt. So darf ein Fußgänger eine Geschwindigkeit von 10 km/h nicht überschreiten und die Maximalgeschwindigkeit für einen Radfahrer beträgt 30 km/h. Da für Fahrzeuge keine sinnvolle Geschwindigkeits-

grenze angenommen werden kann, wir an ihrer stelle die Änderung der Orientierung genutzt. Als Maximale Drehrate wird eine Messung von 0.3 rads/sec aus [4] angenommen. Da der Wert eine Obere Grenze darstellen soll nehmen wir einen etwas höheren Wert von 0.3 rads/sec an.

4.1.5 State Estimation

4.2 Mapping

4.2.1 OpenDAVINIC Map

4.2.2 Simplification

5

Evaluation

5.1 Simulation

5.2 Real Measurements

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 einen Ausblick auf mögliche Konsequenzen oder notwendige weitere zu lösende Probleme

7

Future Work

Bibliography

- [1] Martin Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise”. In: AAAI Press, 1996, pp. 226–231.
- [2] M Himmelsbach, T Luettel, and H.-J. Wuensche. “Real-time object classification in 3D point clouds using point feature histograms”. 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>.
- [3] 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.
- [4] Alonzo Kelly. *A 3D state space formulation of a navigation Kalman filter for autonomous vehicles*. Pittsburgh: Carnegie Mellon University, 1994, p. 95.
- [5] Bo Li, Tianlei Zhang, and Tian Xia. “Vehicle Detection from 3D Lidar Using Fully Convolutional Network”. In: *Robotics: Science and Systems XII*. Robotics: Science and Systems Foundation, 2016. ISBN: 9780992374723. DOI: 10.15607/RSS.2016.XII.042. arXiv: 10.15607. URL: <http://www.roboticsproceedings.org/rss12/p42.pdf>.
- [6] 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.
- [7] Forschungsgesellschaft für Straßen- und Verkehrswesen. Arbeitsgruppe Straßenentwurf. *Merkblatt für die Anlage von Kreisverkehren*. 2006.
- [RAL] Forschungsgesellschaft für Straßen- und Verkehrswesen. Arbeitsgruppe Straßenentwurf. *Richtlinien für die Anlage von Landstrassen (RAL)*. FGSV (Series). FGSV-Verlag, 2013.

- [RASt] Forschungsgesellschaft für Straßen- und Verkehrswesen. Arbeitsgruppe Straßenentwurf. *Richtlinien für die Anlage von Stadtstraßen: RASt 06*. FGSV (Series). FGSV-Verlag, 2007. ISBN: 9783939715214.
- [8] 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 03/06/2017).
- [9] 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>.