



Summercamp 2022

Projekt Automatisiertes Fahren

Karthikeyan Chandra Sekaran 01.03.2022

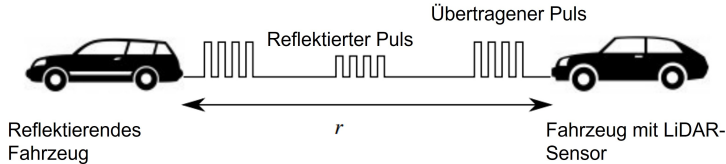


1. Szenariobeschreibung
2. Erläuterung: Sensor Setup
 - 2.1 LiDAR
 - 2.2 ADMA
3. Erläuterung: Datenverarbeitung
 - 3.1 ROS
 - 3.2 Region of Interest
 - 3.3 Ground Subtraction
 - 3.4 Clustering
 - 3.5 Aufgabenstellung und Implementierungshinweise
5. Erläuterung: KI-Algorithmus
 - 5.1 Ermittlung der kritischen Zeit
 - 5.2 Lineare Regression
 - 5.3 Nicht-lineare Basisfunktionen
 - 5.4 Aufgabenstellung und Implementierungshinweise
6. Verwendung der Algorithmen in einer Echtzeit-Anwendung

Das mit LiDAR- und ADMA-Sensorik ausgestattete Ego-Fahrzeug folgt einem Fahrzeug der gleichen Spur, welches überraschend eine Notbremsung einleitet. Um den Auffahrunfall zu verhindern, ist der kritische Zeitpunkt t_{crit} zu bestimmen, zu dem das Ego-Fahrzeug spätestens seine Bremsung einleiten muss.

Ziele:

1. Verarbeitung der Sensorrohdaten des Szenarios
2. Entwicklung eines KI-Algorithmus zur Ermittlung der kritischen Zeit t_{crit} .
3. Anwendung der Algorithmen in einer Echtzeit-Simulation des Szenarios.



Fragen:

1. Welche Assistenzsysteme sind in Fahrzeugen mit LiDAR-Sensor vorstellbar?
2. Welches Assistenzsystem könnte mit diesem Aufbau erprobt werden?
3. Wie kann eine Implementierung aussehen? Welche Eingangsgrößen benötigt die Entscheidungslogik?

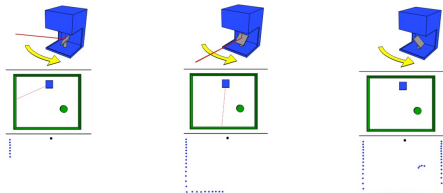
Sensor Setup

Lidar Sensor

LiDAR...

- ... steht als Akronym für „Light Detection And Ranging“
- ... ist ein optisches Verfahren zur Messung von Distanzen
- ... sendet optisches Laserlicht in Pulsen aus.
- ... arbeitet mit Wellenlängen nahe des sichtbaren Bereichs (μm -Wellen)

Visualisierung des Lidar-Prinzips [Wik13]



Sensor Setup

LiDAR Sensor



- **Distanzmessung** per „time-of-flight“-Prinzip:
 - Sende und Empfangszeit des Pulses wird gemessen
 - Aus der Zeitdifferenz τ und der Geschwindigkeit des Lichts $c = 3 \cdot 10^8 \frac{m}{s}$

$$r = \frac{c}{2} \cdot \tau \quad (1)$$

- **Geschwindigkeitsbestimmung:** Nur indirekt auf Basis von Signalverarbeitungstechniken aus der Distanzmessung ableitbar
- **Vorteile:** Großer Öffnungswinkel, Detektion aller möglichen Objekte (ohne Training, da aktiver Sensor), Klassifizierung von Objekten möglich, etc.
- **Nachteile:** Wetteranfälligkeit, keine direkte Geschwindigkeitsmessung, etc.

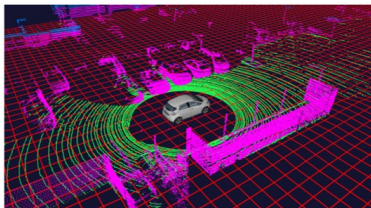
Sensor Setup

LiDAR Sensor: Rohdaten Ausgabe

A I N I N



- **Ausgabeformat:** $[x, y, z, \text{Intensität}] \rightarrow N \times 4$ Matrix für eine Punktwolke mit N Punkten
- **Speicheranforderungen** am Beispiel des Velodyne HDL-64E (rotierender 3D Laserscanner, der z.B. im Open-Source Kitti-Datensatz verwendet wird):
 - Sichtfeld: 360° horizontal, 36.8° vertikal | Reichweite: 120 m | Frequenz: 10 Hz
 - Ausgegebene Punkte: $1.3 \cdot 10^6 \frac{\text{Punkte}}{\text{Sekunde}}$
 - Benötigter Speicher: $1.8 \frac{\text{MB}}{\text{Frame}} \cdot 10 \frac{\text{Frames}}{\text{s}} = 18 \frac{\text{MB}}{\text{s}}$
- **Beispiel:** LiDAR-Aufnahme auf einem Parkplatz [LR12]

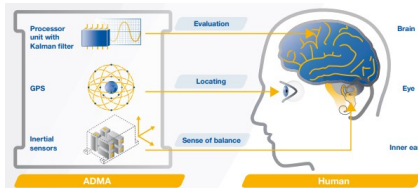


Sensor Setup

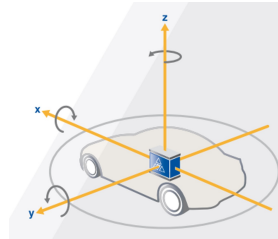
ADMA Sensor

ADMA...

- ... steht als Akronym für „Automotive Dynamic Motion Analyzer“
- ... setzt sich aus IMU, GNSS und einem Prozessor zusammen und kann somit Geschwindigkeit, räumliche Lage des Fahrzeugs und Standort in Echtzeit ausgeben



(a) ADMA als „Sinnesorgan“ des Fahrzeugs [AD14]



(b) ADMA im Fahrzeug [AD14]

Motivation:

- Filtern von störenden oder für den Verwendungszweck nicht relevanten Anteile
- Reduktion des Speicher- und Rechenbedarfs in folgenden Algorithmen
- Ausgabe der Daten entsprechend der Schnittstellenanforderungen folgender Algorithmen
- Nutzen von A-Priori Wissen zur Verkleinerung des ausgewerteten Lösungsraums für folgende Algorithmen → Algorithmus muss diese Eingrenzung nicht mehr erlernen
- Erfüllung der Echtzeitanforderungen

Aufgabenstellung: Ausgabe der Punktwolke für die wichtigen Objekte der Aufnahme

Vorgehensweise:

1. „Region of Interest“-Filterung: Filterung der für den Verwendungszweck unrelvanten Daten
2. „Ground Subtraction“: Filterung der Lidarpunkte der Fahrbahnfläche
3. „Clustering“: Erkennen zusammenhängender Punkte zur Ausgabe von Objekten

Implementierung der Datenverarbeitung

ROS



Motivation:

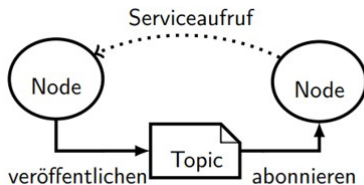
- bietet Struktur, Tools und Algorithmen für die Implementierung eigener Applikationen
- Sensorausgabe kann in dieses System eingebunden und somit verarbeitet werden

Allgemein:

- Open-Source Framework
- Aufgaben: Hardwareabstraktions, Hilfsfunktionen, Interprozesskommunikation, Gerätetreiber, Paketmanagement



- **ROS Nodes:** Eigenständige Programme, die ihre vorgegebene Logik ausführen
- **ROS Master:** Zentraler Haupt-Knoten im System → kennt die Nodes des Systems und ermöglicht dadurch die Kommunikation
- **ROS Topic:** eigener „Kommunikationskanal“, wodurch Erzeugung und Nutzung von Information entkoppelt ist
- Serviceorientierte Architektur, die mit Publisher-Subscriber Beziehungen arbeitet
- Programmiersprachen: C++, Python



Hier die Verbindung zwischen Node des Sensors und Empfang im implementierter Logik darstellen

Motivation

- Je nach Algorithmus ist nur ein Teil des Datensatzes für dessen Auswertung relevant
- Beispiel für A-Piori Wissen im automatisierten Fahren: HD-Maps
 - Im Fahrzeug verfügbare Information über die Umgebung
 - Daraus ableitbar: Regionen mit möglichem Verkehrsaufkommen und potentielle Gefahrenbereiche
 - Folgerung: gezielte Reduktion des ausgewerteten Datenraums auf den für den Algorithmus relevanten Bereich
 - Beispiel: Adaptive Cruise Control wertet Region aus, in der das vorausfahrendes Fahrzeug vermutet wird, da nur jene für die Aufgabe relevant ist.

Region of Interest HD-map

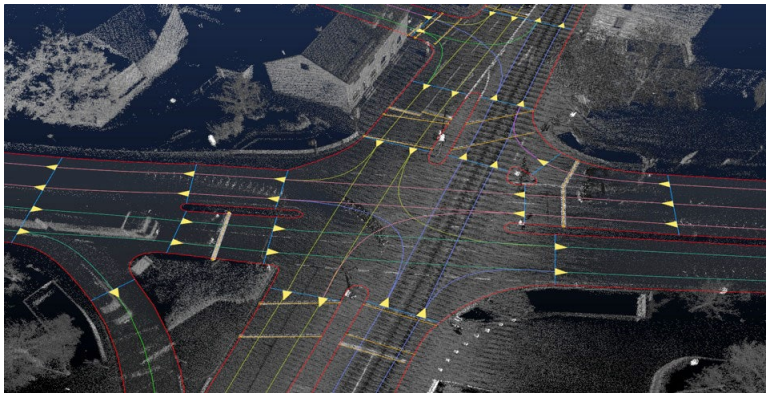


Abbildung: Beispiel einer HD-map [Quelle 4]

Implementierung: Datenverarbeitung

Region of Interest

Anwendung: Wissen über die Teststrecke als im Fahrzeug verfügbare HD-map

Aufgabenstellung: Begrenzung des Datensatzes auf Basis von 4 gegebenen Punkten ${}^G P_i$ des globalen Koordinatensystems G der Teststrecke



Implementierung: Datenverarbeitung

Region of Interest



Gegeben:

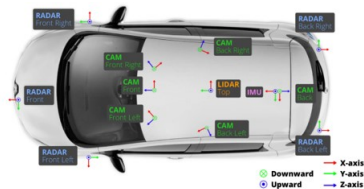
- Punkte: ${}^G P_1 = (x, y, z)$, ${}^G P_2 = (x, y, z)$, ${}^G P_3 = (x, y, z)$, ${}^G P_4 = (x, y, z)$
- Transformationsmatrizen:

Vorgehensweise:

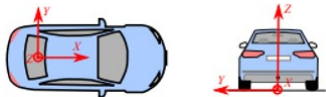
1. Transformation der LiDAR-Punkte ${}^L P_i$ in das globale Koordinatensystem
2. Filterung aller Punkte mit Koordinaten (x, y) außerhalb des durch die 4 Punkte ${}^G P_i$ aufgespannten Bereichs

Region of Interest: Implementierungshinweise

Lokale Koordinatensysteme



(a) Sensorkoordinatensysteme in nuScenes
[Quelle 143]



(b) Fahrzeugkoordinatensystem exterozeptiver Sensoren



(c) Fahrzeugkoordinatensystem der Fahrdynamikgleichungen

Region of Interest: Implementierungshinweise

Lokale Koordinatensysteme



- Zahlreiche Koordinatensysteme im Fahrzeug, die eine Betrachtung der Information aus verschiedenen Perspektive ermöglichen
- Zwischen den lokalen Koordinatensystemen des Fahrzeugs kann ein statischer Zusammenhang, analog einer Starrkörperbewegung, angenommen werden → Transformationsmatrizen sind zeitunabhängig
- Fahrzeugkoordinatensysteme sind bei der Berechnung zwingend zu beachten, da Sie den Variablen Bedeutung verleihen
- Beispiel einer Fahrdynamikgleichung:

$${}^F v_x = \frac{\delta}{\delta t} x \text{ gilt für } x = {}^F x \quad (2)$$

Region of Interest: Implementierungshinweise

Globale Koordinatensysteme

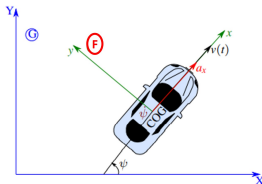


Abbildung: Globales Koordinatensystem

- Stationäres Referenzkoordinatensystem
- Ermöglicht die Beschreibung von Objekten relativ zu einem konstanten Koordinatensystem
- Transformationsmatrix von lokalem Fahrzeugkoordinatensystem zu globalem Koordinatensystem ist für instationäre Objekte zeitabhängig

Region of Interest: Implementierungshinweise

Anwendung auf das Szenario



Nomenklatur:

- $^G P$: Punkt des globalen Koordinatensystem
- $^L P$: Punkt des LiDAR Koordinatensystems
- $^F P$: Punkt des Fahrzeugkoordinatensystem (CoG)
- $^{EF} P$: Punkt des Fahrzeugkoordinatensystems der exterozeptiven Sensoren

Region of Interest: Implementierungshinweise

Anwendung auf das Szenario

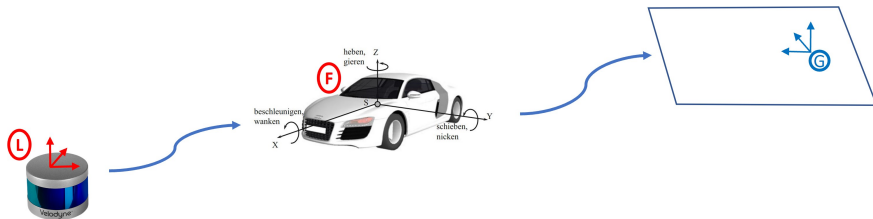


Abbildung: Anwendung der Koordinatentransformation auf das Szenario

- statische Koordinatentransformation → zeitabhängige Koordinatentransformation
- ADMA gibt Position des Fahrzeugs im globalen Koordinatensystem
- Umsetzung mit tf-Modul aus ROS

Region of Interest

Implementierungshinweise

auf ROS Transformatoren eingehen



Motivation

- LiDAR Sensor gibt zahlreiche Reflektionen der Fahrbahnoberfläche aus
- Diese Punkte enthalten keine Objekte und haben deswegen für die Objekterkennung keine Relevanz

Ziele

- Objektdetektionsalgorithmus muss Vernachlässigung von Reflektionen des Bodens nicht mehr erlernen
- Verkleinerung des Lösungsraums

Implementierung: Datenverarbeitung

Ground Subtraction

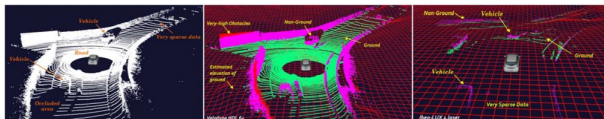


Abbildung: Beispiel Ground Subtraction [LR12]

Implementierung: Datenverarbeitung

Ground Subtraction

Anwendung:

- Es wird eine ebene Fahrbahnoberfläche auf der Teststrecke angenommen
- Zur Vereinfachung wird die Eliminierung der Punkte der Fahrbahnoberfläche auf Basis Höhe z durchgeführt
- Alternativen: RANSAC Algorithmus und weitere → weniger intuitiv, aber robuster



Aufgabenstellung: Verkleinerung des Datensatzes um Punkte der Fahrbahnoberfläche:

Bedingung ${}^{EF}z \leq 0.01m$

Implementierung: Datenverarbeitung

Clustering

Motivation:

- Verwendung der "Ähnlichkeit" von Datenpunkten (bspw. Koordinaten), um zusammengehörende Punkte zu erkennen und zu segmentieren
- Damit können Punkte bestimmten Clustern bzw. Objekten zugeordnet werden
- Ermöglicht unter anderem eine Lokalisierung von Objekten der Umgebung

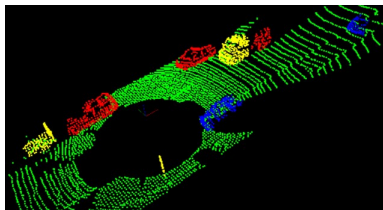


Abbildung: Beispiel Clustering [Quelle CU]

Implementierung: Datenverarbeitung

Clustering



Anwendung: Im Szenario bedeutet dies die Ausgabe eines Clusters für das detektierte Fahrzeug und weiterer Umgebungscluster????

Aufgabenstellung: Auf Basis des k -Means Clusteralgorithmus sollen die Datenpunkte des vorausfahrenden Autos gruppiert und segmentiert werden

Implementierung: Datenverarbeitung

Clustering



Umsetzung:

- Möglichkeiten:
- Gewählter Algorithmus: k-Means Clustering
- Beschreibung des Algorithmus

k-Means-Algorithmus

Erläuterung



Funktion: Aus einer Menge von ähnlichen Objekten wird eine vorher bekannte Anzahl von k Gruppen (Clustern) gebildet

Umsetzung:

- Ziel ist den Datensatz so in k Partionen einzuteilen, dass die Summe der quadratischen Abweichungen von den Cluster-Schwerpunkten minimal wird → Clustering durch Varianzminimierung

$$J = \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \quad (3)$$

mit den Datenpunkten x_j und den Schwerpunkten μ_i der Cluster S_i

k-Means-Algorithmus

Erläuterung



Umsetzung: Iterative Lösungsapproximation ersetzt kompliziertere Optimumssuche

- Lloyd-Algorithmus:
 1. Initialisierung: Wähle k zufällige Mittelwerte (Means): m_1^1, \dots, m_k^1 aus dem Datensatz
 2. Zuordnung: Jedes Datenobjekt wird dem Cluster zugeordnet, bei dem die Cluster-Varianz am wenigsten erhöht wird

$$S_i^t = x_j : \|x_j - m_i^t\|^2 \text{ für alle } i^* = 1, \dots, k \quad (4)$$

3. Aktualisierung: Neuberechnung der Mittelpunkte der Cluster:

$$m_i^{t+1} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i} x_j \quad (5)$$

- Schritte 2-3 werden wiederholt, bis die Zuordnung konstant bleibt

k -Means-Algorithmus

Erläuterung



Vorteil: Leistungsfähigkeit

Nachteile:

- Gefundene Lösung hängt stark von den gewählten Startpunkten ab
- Anzahl der Clusterzentren k bereits im Voraus gewählt

Definition der Bounding Box



Ableitung der Objekt Informationen als Bounding Box auf Basis der Ränder der Cluster

Equations

Intrinsic calibration



$$i_{\text{undist}} = f_{\text{undist}}(i_{\text{raw}}/\theta_d)$$

where, i_{raw} = raw distorted image

i_{undist} : undistorted image

θ_d : Distortion parameters

- 1) [Wk13] Wikipedia Lidar
- 2) [LR12] Ground Estimation and Point Cloud Segmentation using SpatioTemporal Conditional Random Field (Lukas Rummelhard et. al.)
- 3) [AD14] ADMA Präsentation der GeneSys Elektronik GmbH
- 4) [Quelle 4] <https://www.wired.com/2014/12/nokia-here-autonomous-car-maps/>
- 5) [Quelle FD] ISBN 978-3-658-09474-4 ISBN 978-3-658-09475-1 (eBook) DOI 10.1007/978-3-658-09475-1
- 6) [Quelle CU] <https://www.codetd.com/en/article/10631576>