

BACHELOR THESIS
Marco Förster

Freiraumerkennung mit V2X-Unterstützung für autonome Fahrzeuge

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Engineering and Computer Science
Department Computer Science

Marco Förster

Freiraumerkennung mit V2X-Unterstützung für autonome Fahrzeuge

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Angewandte Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Thomas Schmidt
Zweitgutachter: Prof. Dr.-Ing. Michael Reke

Eingereicht am: 04. Mai 2026

Marco Förster

Thema der Arbeit

Freiraumerkennung mit V2X-Unterstützung für autonome Fahrzeuge

Stichworte

V2X, Autonom, CPM, Freiraum

Kurzzusammenfassung

Eine der wichtigsten Fähigkeiten autonomer Fahrzeuge ist die Umgebungswahrnehmung durch diverse Sensoren, um zu jedem Zeitpunkt eine sichere Entscheidung über die nächste Handlung zu treffen. Eine hohe Präzision der Messungen bei gleichzeitiger Echtzeitfähigkeit und Ausfallsicherheit sind nötig, um die Sicherheit aller Verkehrsteilnehmer zu maximieren. Zwei wichtige Säulen der Wahrnehmung sind die Erkennung anderer Objekte, z.B. anderer Verkehrsteilnehmer wie Autos und das Erfassen freier Regionen. Dies ermöglicht eine Einteilung der direkten Umgebung grob in zwei Klassen; frei, hier darf das Fahrzeug entsprechend der Verkehrsregeln normal fahren oder nicht frei, und muss entsprechend der Situation schnell und sicher reagieren. Im Straßenverkehr kommt es jedoch häufiger zu Situationen in denen die Sicht eingeschränkt wird, wodurch relevante Information nicht Wahrgenommen werden können. Mithilfe von Vehicle to Everything (V2X) können Fahrzeuge ihre Sensordaten mit anderen Teilnehmern über eine Funkverbindung austauschen. Die Collective Perception Message (CPM) ist eine standardisierte Nachricht speziell für Objekt- und Freirauminformationen und gewährleisten eine Kommunikation zwischen verschiedenen Herstellern. Bei Funkverbindungen treten Latenzen und Paketverluste auf. Diese Arbeit stellt einen Ansatz vor, wie diese CPM Freiraumdaten verarbeitet werden können um die lokale Wahrnehmung zu erweitern und evaluiert diesen in einem Simulierten Kreuzungsszenario unter verschiedenen Bedingungen.

Marco Förster

Title of Thesis

Free space detection with V2X support for autonomous vehicles

Keywords

V2X, autonomous, CPM, free space

Abstract

One of the most important capabilities of autonomous vehicles is environmental perception through various sensors, enabling them to make safe decisions about their next action at all times. High measurement precision, combined with real-time processing and reliability, is essential to maximize the safety of all road users. Two key pillars of perception are the detection of other objects—such as other road users like cars—and the identification of clear areas. This allows the immediate surroundings to be roughly divided into two categories: clear, where the vehicle may drive normally in accordance with traffic rules, or not clear, requiring the vehicle to react quickly and safely depending on the situation. In road traffic, however, situations frequently arise in which visibility is restricted, preventing relevant information from being perceived. With the help of V2X, vehicles can exchange their sensor data with other road users via a wireless connection. The CPM is a standardized message specifically designed for object and free-space information and ensures communication between different manufacturers. Radio connections are subject to latency and packet loss. This paper presents an approach for processing this CPM free-space data to enhance local perception and evaluates it in a simulated intersection scenario under various conditions.

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
Tabellenverzeichnis	viii
Abkürzungen	ix
1 Einführung	1
1.1 Motivation	1
1.2 Problemstellung	3
1.3 Zielsetzung und Forschungsfrage	4
1.4 Aufbau der Arbeit	4
2 Stand der Technik	5
2.1 Kollektive Wahrnehmung und V2X-Kommunikation	5
2.2 Infrastrukturgestützte Wahrnehmung	5
2.3 Okklusion und Sichtbereichsmodellierung	6
2.4 Latenz und Unsicherheiten in kooperativen Systemen	7
3 Systemumgebung und Grundlagen	8
3.1 Software- und Systemarchitektur	9
3.2 Kommunikationsgrundlagen	11
3.3 Projektkontext (Autonomous Driving Project)	11
4 Konzeption	14
4.1 Problemformalisierung	14
4.2 Freiraum-Repräsentation	16
4.3 Freiraum extraktion aus CPM	17
4.4 Freiraum Interpolation	17
4.5 Freiraum Kombination	19
4.6 Entscheidungslogik (Freigabealgorithmus)	19

5	Evaluation	21
5.1	Versuchsaufbau	21
5.2	Szenarien	23
5.3	Metriken	26
5.4	Ergebnisse	26
5.5	Diskussion	27
6	Fazit und Ausblick	29
6.1	Ausblick	29
	Literaturverzeichnis	31
7	Anhang	35
7.1	Verwendete Hilfsmittel	35
	Selbstständigkeitserklärung	36

Abbildungsverzeichnis

1.1	Hindernis verdeckt Sicht auf Straße mit verstecktem Verkehrsteilnehmer für Sensor und Fahrzeugführer	2
4.1	Bildschirmaufnahme: JOSM Lanelet2-Karten Editor	14
4.2	Kreuzung mit Debuggvisualisierung während Entwicklung	20
5.1	Bildschirmfoto: Google Maps T-Kreuzung gegenüber vom KMAC Aachen	21
5.2	Rviz Bildschirmfoto: Startposition des Fahrzeugs für Szenarien	22
5.3	Rviz Bildschirmfotos: Sensor lokal teilblockiert, Fahrfreigabe nicht erteilt .	25
5.4	Rviz Bildschirmfoto: Fahrfreigabe nicht erteilt	26

Tabellenverzeichnis

7.1	Verwendete Hilfsmittel und Werkzeuge	35
-----	--	----

Abkürzungen

ADP Autonomous Driving Project.

CPM Collective Perception Message.

DDS Data Distribution Service.

GNSS Global Navigation Satellite System.

HMETC Hyuandi Motor Europe Technical Center.

IKA Institut für Kraftfahrzeuge.

KMAC Kompetenzzentrum Mobilität.

QoS Quality-Of-Service.

ROS2 Robot Operating System 2.

UDP User Datagram Protocol.

V2X Vehicle to Everything.

1 Einführung

1.1 Motivation

Die menschliche Wahrnehmung, so wie die klassischer Sensoren im Bereich autonomer Fahrzeuge, basiert auf linearem Sichtkontakt, dies sind z.B. Kamera-, LiDAR- und Radarsensoren oder das menschlichen Auge. Hindernisse schränken diesen jedoch regelmäßig ein und verdecken dadurch hinter ihnen liegende Informationen, die eine Weiterfahrt ohne Geschwindigkeitsreduktion zu einem Glücksspiel für sich und andere Verkehrsteilnehmer machen. Aber auch bei einem nicht beschränkten Sichtfeld ist die Wahrnehmung in ihrer Reichweite begrenzt.[1] Die Folge von Verdeckung sind schlecht einsehbare Einfahrten, enge Kreuzungen oder unübersichtliche Kurven, eine entsprechende Beispielgrafik ist in 1.1 dargestellt. In der Grafik zeigt das blaue Rechteck (A) die Position eines Fahrzeuges mit 360 Grad Sensor, welcher in der Fahrzeugmitte verbaut ist. Das Fahrzeug steht in einer Ausfahrt und möchte in die Straße abbiegen. Das orangene Rechteck (B) zeigt die Position eines aktiven Fahrzeuges auf eben dieser Straße. Die Pfeile in A und B verweisen auf die jeweilige Fahrtrichtung. Die roten Rechtecke (C) sind parkende Fahrzeuge. Der Sensor von A wird durch C blockiert, sodass A innerhalb des rot-gestrichelt umrandeten Bereichs keine freie Sicht hat. Folglich ist B für A nicht zu erkennen und berücksichtigt es nicht in seiner Pfadplanung. Wenn A nun zum Zeitpunkt der Darstellung beschleunigt, würde es mit Fahrzeug B zu einer Kollision kommen. Dies führt zu konservativem Verhalten, also einer Reduktion der Geschwindigkeit entsprechend der Situation, um im Falle eines plötzlich erscheinenden Verkehrsteilnehmers, genügend Zeit zu haben, auf diesen zu reagieren oder im Falle einer Kollision den Schaden zu minimieren. In besonders extremen Fällen ist ein garantiert sicheres Fahren ohne externe Hilfe gar nicht möglich. Als klassische Abhilfe haben sich für Menschen großflächige Konvexspiegel etabliert, die an kritischen Stellen montiert werden und einen Blick um ein Hindernis herum ermöglichen. Diese Arbeit untersucht unter anderem, wie eine digitale Analogie solch eines Spiegels, mithilfe von CPM realisiert werden kann – mit dem Ziel, sowohl autonome als

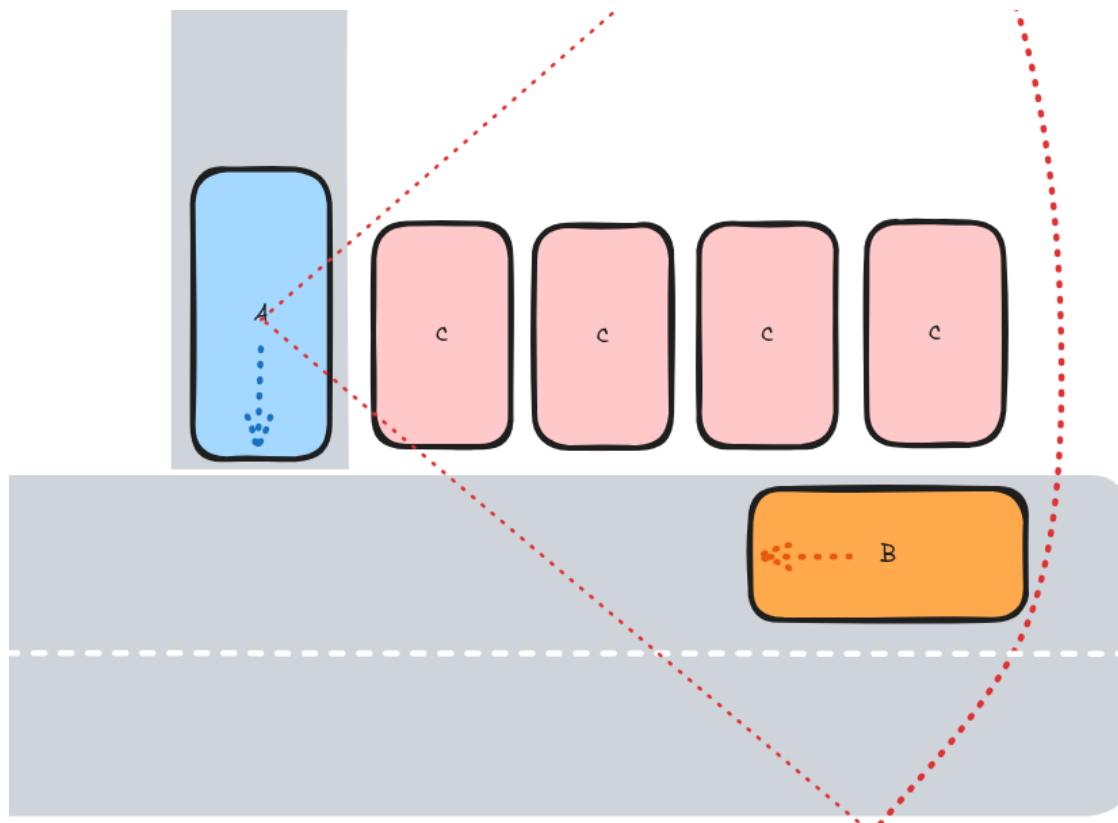


Abbildung 1.1: Hindernis verdeckt Sicht auf Straße mit verstecktem Verkehrsteilnehmer für Sensor und Fahrzeugführer

auch manuell gesteuerte Fahrzeuge mit Informationen zu versorgen, die durch rein lokale Sensorik nicht erfassbar sind. Die zusätzlichen Informationen zielen auf die Erhöhung der Fahrsicherheit und die Erweiterung des Einsatzbereichs autonomer Fahrzeuge. Um eine möglichst hohe Sicherheit zu gewährleisten, wird oftmals (konfigurations- und situationsabhängig) der theoretische Worst Case betrachtet. Basierend auf physikalischen Berechnungen der aktuellen Situation sollte immer genügend Zeit vorhanden sein, um auf alle möglichen Notfallsituationen zu reagieren. In solchen Fällen verzögert das autonome Fahrzeug seine Geschwindigkeit, bleibt stehen und bricht die autonome Fahrt gegebenenfalls komplett ab.

Im Rahmen des in [2] beschriebenen Projektes wurde der Objektaustausch der CPM-Kommunikation innerhalb des Autonomous Driving Project (ADP) implementiert und verifiziert. Die dort verwendete Implementierung überträgt Objektinformationen wie Position, Orientierung, Abmessungen und Geschwindigkeit. Dabei wurde jedoch ein wesentliches Defizit identifiziert: Die Sichtbereiche der beteiligten Sensoren werden bislang nicht

übermittelt. Dies führt zu einem kritischen Interpretationsproblem – empfängt ein Fahrzeug eine CPM-Nachricht ohne Sichtbereiche der Sensoren und einer leeren Objektliste, kann daraus nicht zwingend geschlossen werden, dass der relevante Bereich tatsächlich frei ist. Es ist ebenso möglich, dass die Sensorik den fraglichen Bereich nicht erfasst, etwa aufgrund seiner Ausrichtung oder physischer Hindernisse im Sichtfeld. Ziel dieser Arbeit ist es daher, ein Pendant zu diesem Projekt zu entwickeln, dass sich mit der Implementierung des Freiraums der CPM-Kommunikation, ohne Objekte, befasst.

1.2 Problemstellung

Der Empfang von CPM Nachrichten geschieht über eine Funkverbindung. Dabei kann es vorkommen, dass vereinzelt Nachrichten ausfallen oder die Verbindung ganz abbricht, falls die Hardware komplett ausfällt oder aus der Funkreichweite fällt. Durch eine Funkverbindung entsteht zudem immer eine zusätzliche Latenz, die durch die Signallaufzeit verursacht wird. Beide Effekte führen dazu, dass extern empfangene Daten in ihrer zeitlichen Qualität immer weiter in der Vergangenheit liegen, als lokal gemessene Daten bei Nutzung gleicher oder ähnlicher Hardware. Lokale Daten sind auch immer verzögert, bedingt durch die Messdauer selber und Verarbeitungslatenzen. Zudem sind Messungen generell immer nur zu einem bestimmten Grad akkurat, im Bezug auf die Freiraumdaten bezieht sich dies auf Distanzen, was sich auf die Genauigkeit der Positionierung auswirkt. Außerdem haben verschiedene CPM-Sender unterschiedliche Hardwaresetups, die jeweils unterschiedliche Messungenauigkeiten haben können. Sämtliche Latenzen unterliegen des Weiteren unterschiedlich starken Schwankungen, die von verschiedenen Faktoren abhängig sind wie z.B. verschiedene Wetterbedingungen. Die Freiräume, sowohl von externen als auch lokalen Sensoren, besitzen also immer eine zeitliche und örtliche Ungenauigkeit. Autonome Fahrzeuge müssen auf Basis dieser Daten Entscheidungen treffen, die Handlungen in der Zukunft betreffen. Wie kann trotz dieser Herausforderungen eine sichere Entscheidung für die Zukunft getroffen werden?

1.3 Zielsetzung und Forschungsfrage

Die Basis soll ein Programm zur Kreuzungsfreigabe sein, das eine deterministische und nachvollziehbare Entscheidung trifft, ob die nächste Kreuzung durchfahren werden darf oder nicht. Als Grundlage für diese Entscheidung dienen unter anderem Freirauminformationen, welche im Umfang der Arbeit als synthetisch erzeugte CPM ROS2 Nachrichten, sowohl für die externe als auch die lokale Sensorik, empfangen werden. Die Verarbeitung geschieht durch ein zweites Programm, das die im CPM-Format vorliegenden Freiraumdaten der externen und lokalen Sensoren, in eine interne Repräsentation umwandelt, anschließend auf den aktuellen Zeitpunkt interpoliert und zu einem gemeinsamen Freiraum vereinigt.

Forschungsfrage:

Inwiefern verbessern externe Freirauminformation die Entscheidungsqualität autonomer Fahrzeuge bei eingeschränktem Sichtfeld ihrer Sensoren?

1.4 Aufbau der Arbeit

Die Arbeit gliedert sich in sechs Kapitel. In der Einleitung wird die Motivation, Zielsetzung und Forschungsfrage dargelegt. Kapitel 2 gibt einen Überblick über den Stand der Technik. Dabei werden relevante wissenschaftliche Arbeiten vorgestellt und analysiert, um bestehende Ansätze und Konzepte im Bereich der kollektiven Wahrnehmung und V2X-Kommunikation einzuordnen. Kapitel 3 beschreibt die im Rahmen dieser Arbeit verwendete bestehende autonome Softwarearchitektur und die verwendeten Frameworks. Kapitel 4 widmet sich der Konzeption des entwickelten Systems und beschreibt das Gesamtmodell auf konzeptioneller Ebene. Kapitel 5 erläutert den Versuchsaufbau, betrachtet verschiedene Szenarien und diskutiert die Ergebnisse dieser Arbeit. Kapitel 6 enthält abschließend ein Fazit, beantwortet die Forschungsfrage und bietet einen Ausblick für zukünftige Forschung.

2 Stand der Technik

2.1 Kollektive Wahrnehmung und V2X-Kommunikation

Die Idee, Informationen zwischen Fahrzeugen auszutauschen, ist seit vielen Jahren Bestandteil der Forschung im Bereich des autonomen Fahrens. Ziel ist es, die durch lokale Sensorik begrenzte Wahrnehmung zu erweitern und so ein vollständigeres Bild der Umgebung zu erhalten. Bereits Mourllion et al. [3] zeigen, dass eine kollaborative Wahrnehmung mehrerer Fahrzeuge die Kollisionsvermeidung deutlich verbessern kann. Aufbauend darauf untersuchen Rauch et al. [4], welche Anforderungen an die Kommunikation gestellt werden müssen, damit solche Systeme zuverlässig funktionieren. Dabei wird deutlich, dass insbesondere Latenz, Reichweite und Datenrate eine zentrale Rolle spielen. Günther et al. betrachten zunächst das grundsätzliche Potenzial kollektiver Wahrnehmung in Fahrzeugnetzwerken [5] und demonstrieren später auch eine praktische Umsetzung in einem realen Fahrzeug [6]. Diese Arbeiten zeigen, dass der Ansatz nicht nur theoretisch relevant ist, sondern auch technisch realisierbar. Auch auf strategischer Ebene wird die Bedeutung von V2X-Kommunikation hervorgehoben. Initiativen wie das CAR 2 CAR Communication Consortium [7] oder die 5G Automotive Association [8] sehen kollektive Wahrnehmung als zentrale Technologie für zukünftige Verkehrssysteme. Ergänzend dazu zeigen Feifel et al. [9], dass V2X-basierte Systeme bereits heute das Unfallrisiko in realen Verkehrssituationen reduzieren können.

2.2 Infrastrukturgestützte Wahrnehmung

Neben der direkten Kommunikation zwischen Fahrzeugen spielt auch die Einbindung stationärer Infrastruktur eine zunehmend wichtige Rolle. Der grundlegende Gedanke ist dabei ähnlich: Sensoren erfassen Informationen, die einem einzelnen Fahrzeug nicht zugänglich sind, und stellen diese anderen Verkehrsteilnehmern zur Verfügung. Gabb et al.

[10] zeigen, wie solche Systeme mithilfe von Edge Computing umgesetzt werden können. Sensordaten werden dabei zentral verarbeitet und zu einem gemeinsamen Umgebungsmodell zusammengeführt. Klöppel-Gersdorf et al. [11] demonstrieren mit der Smart Intersection ein konkretes Beispiel für eine solche intelligente Infrastruktur. Ein besonders relevanter Ansatz für diese Arbeit ist das MEC-View-Projekt von Buchholz et al. [12]. Hier werden Sensordaten von Kameras und LiDAR-Sensoren an einer Kreuzung erfasst, über einen Edge-Server fusioniert und anschließend an Fahrzeuge übertragen. Die Ergebnisse zeigen, dass Fahrzeuge dadurch effizienter und flüssiger durch Kreuzungen navigieren können, da sie weniger häufig anhalten müssen. Gleichzeitig wird auch eine Reduktion des Energieverbrauchs beobachtet. Weitere Arbeiten [13, 14, 15] belegen, dass solche Systeme nicht mehr nur im Labor existieren, sondern bereits in realen Testumgebungen eingesetzt werden. Zimmermann et al. [16, 17] zeigen zusätzlich, dass insbesondere in schlecht einsehbaren Kreuzungsszenarien ein deutlicher Sicherheitsgewinn erzielt werden kann. Ein wichtiger Aspekt dieser Ansätze ist, dass die finale Entscheidung weiterhin im Fahrzeug getroffen wird. Die Infrastruktur liefert nur zusätzliche Informationen, ersetzt jedoch nicht die lokale Wahrnehmung. Dieses Prinzip wird in der vorliegenden implementation der Arbeit bisher nicht umgesetzt, da es für die Evaluation nicht notwendig war, aber dies in der Konzeption berücksichtigt wurde.

2.3 Okklusion und Sichtbereichsmodellierung

Ein zentrales Problem autonomer Systeme ist die eingeschränkte Sicht durch Hindernisse. Solche Okklusionen führen dazu, dass relevante Verkehrsteilnehmer nicht erkannt werden können, was insbesondere an Kreuzungen oder Einfahrten kritisch ist. Llatser et al. [18] adressieren diese Problematik im Kontext von 5G-V2X und zeigen, wie kooperative Systeme helfen können, solche Sichtlücken zu schließen. Ein anderer Ansatz wird von Moller et al. mit dem Phantom-Agent-Modell verfolgt [19]. Hierbei werden in verdeckten Bereichen hypothetische Verkehrsteilnehmer angenommen, um mögliche Risiken frühzeitig zu berücksichtigen. Der Ansatz mit der Modellierung von Phantom Agents ähnelt dem Ansatz dieser Arbeit, jedoch werden diese direkt in den Statues des Freiraums abstrahiert und nicht separat behandelt. Zhang et al. [20] verfolgen einen ähnlichen Gedanken, kombinieren diesen jedoch mit einem POMDP-basierten Planer. Dabei werden lokale Sensordaten und V2X-Informationen getrennt verarbeitet. Die Okklusionsbereiche werden zunächst identifiziert, anschließend werden darin potenzielle Verkehrsteilnehmer modelliert und durch externe Informationen verfeinert. Im Gegensatz dazu wird in dieser Arbeit ein

Fusionsansatz verfolgt, bei dem Freirauminformationen aus verschiedenen Quellen zu einer gemeinsamen Repräsentation zusammengeführt werden und dabei in ihrer Wertigkeit nicht unterschieden werden. Als Grundlage dient dabei das Lanelet2-Kartenformat [21], das eine präzise geometrische Zuordnung der Sichtbereiche ermöglicht.

2.4 Latenz und Unsicherheiten in kooperativen Systemen

Ein wesentliches Problem kooperativer Systeme ist die zeitliche Verzögerung der übertragenen Daten. Während lokale Sensoren bereits eine gewisse Latenz aufweisen, kommt bei V2X-Kommunikation zusätzlich die Übertragungszeit hinzu. Im MEC-View-Projekt wurde eine Gesamtlatenz von durchschnittlich 242 ms gemessen [12]. Diese setzt sich aus Sensorverarbeitung, Serverlaufzeit und Kommunikation zusammen und zeigt, dass externe Informationen grundsätzlich zeitlich verzögert sind. In der Forschung existieren verschiedene Ansätze, um mit dieser Problematik umzugehen. Wong et al. schlagen mit dem LRCP-Framework [22] eine lernbasierte Methode vor, bei der zeitlich versetzte Daten direkt fusioniert werden. Ein ergänzender Ansatz ist V2X-INCOP von Chen et al. [23], bei dem fehlende Informationen aus vergangenen Daten rekonstruiert werden. Beide Ansätze basieren auf Deep Learning und erfordern umfangreiche Trainingsdaten sowie hohe Rechenleistung. Der in dieser Arbeit verfolgte Ansatz unterscheidet sich bewusst davon. Statt komplexer Modelle wird eine physikalisch motivierte Worst-Case-Betrachtung genutzt. Dabei wird berechnet, wie weit sich ein potenzielles Objekt während der zu kompensierenden Zeitdifferenz zum gewünschten Zeitpunkt, maximal bewegen könnte und direkt in die Freiraumdaten abstrahiert. Dieser Ansatz ist deterministisch, nachvollziehbar und in Echtzeit einsetzbar. Dadurch ist kein Training notwendig und ist damit einfacher in bestehende Systeme integrierbar. Diese Eigenschaften sind insbesondere im Kontext sicherheitskritischer Anwendungen von Vorteil.

3 Systemumgebung und Grundlagen

Die getroffenen Entscheidungen im Rahmen dieser Arbeit basieren auf technischen Standards und wissenschaftlichem Konsens. Die Entscheidungen werden sofern notwendig wissenschaftlich begründet. Die Arbeit erfolgt in Kooperation mit der FH Aachen im Rahmen des ADP Projektes, über welches Soft- und Hardware bereitgestellt wurde. Die bereitgestellte Soft- und Hardware basiert auf der gleichen Grundlage, wodurch diese Arbeit im Rahmen der kontinuierlichen Weiterentwicklung des ADP nahtlos eingebettet werden kann.

In der bereitgestellten Soft- und Hardware fehlt eine Freiraumerkennung über echte Sensoren. Während es technisch möglich ist, Freiraum über V2X zu empfangen, ist die ADP-Software zum aktuellen Zeitpunkt nicht in der Lage, diese zu verarbeiten. Die in dieser Arbeit entwickelte Erweiterung der ADP-Software bietet hier die entsprechende Grundlage für diese Funktionalität. Im Verlauf der Arbeit werden Entscheidungen und Annahmen getroffen, die durch diese Zusammenarbeit entstehen. Gleichzeitig schränkt dies die allgemeine Aussagekraft dieser Arbeit nicht ein, und ihre Ergebnisse sind somit wissenschaftlich unabhängig von dieser gültig. Im Rahmen der Weiterentwicklung der ADP-Software dienen diese auch als Vergleichsgrundlage für später folgende Tests mit echter Hardware. Zum aktuellen Stand ist dies ein 360-Grad-LiDAR, der für die Freiraumerkennung verwendet werden soll, sowie eine OBU, die eine ETSI-konforme V2X-Kommunikation bereitstellt, zu der auch die CPM gehören. Die Hochschule verfügt zudem über eine RSU, die als externer CPM-Sender dient und an der Gebäudefassade verbaut werden könnte. Daraus resultieren realistische Beschränkungen, die als Grundlage für diese Arbeit dienen. z.B. Nur ein externer CPM-Sender reicht erstmal. Die gesamte Kommunikation erfolgt im CPM-Format. Zunächst soll nur die Freiraumerkennung validiert werden, bevor in einem nachfolgenden Projekt die Arbeiten mit CPM-Objekten und den CPM-Freiräumen vereint werden. Die Implementierung der Objektfelder der CPM ist daher zu diesem Zeitpunkt nicht erforderlich.

3.1 Software- und Systemarchitektur

- Robot Operating System 2 (ROS2)

Das ROS2 ist ein Open-Source-System, das verschiedene Softwarebibliotheken und Tools bereitstellt und nutzt, welche vor allem im Bereich der Robotik Anwendung finden. Es ist, wie der Name fälschlicherweise suggeriert, kein eigenes Betriebssystem. [24] Die Vorgängerversion von ROS2, ROS(1) wurde bereits im Jahr 2007 veröffentlicht. Eine der grundlegendsten Funktionen von ROS2 ist die eventbasierte, lokal verteilte System- und Interprozesskommunikation. Diese ermöglicht dem Anwender eine einfache und schnelle Kommunikation zwischen verschiedenen Prozessen, welche im gesamten Netzwerk auf unterschiedlichsten Geräten verteilt sein können. [25] ROS2 nutzt den Data Distribution Service (DDS)-Standard (Data Distribution Service), um eine Kommunikation zu gewährleisten. Dadurch können verschiedene Einstellungsmöglichkeiten zur Zuverlässigkeit der Kommunikation beitragen, indem die Quality-Of-Service (QoS) (Quality-of-Service) von DDS angepasst wird. Mit dem Update auf ROS2 wurde auch das Echtzeitverhalten der Kommunikation optimiert, was in der Robotik von großer Bedeutung ist. In der grundlegenden Struktur von ROS2 werden verschiedene Nodes erstellt, die jeweils Nachrichten senden und/oder empfangen können. Ein Sender, auch Publisher genannt, muss seine Nachricht immer mit einem Topic versehen. Dies ist vergleichbar mit einer E-Mail- oder Postadresse, mit dem Unterschied, dass nicht ein direkter Empfänger, sondern eine öffentliche Adresse adressiert wird, auf die jeder Zugriff hat, der die Information benötigt. Dies geschieht durch eine simple Zeichenabfolge, z.B. könnte ein LiDAR-Sensor seine Punktwolken als `lidar_pointcloud` publizieren. Die Nachricht selbst muss auch ROS intern vorliegen. Hierfür kann entweder auf die bereits integrierten Standardformate zugegriffen werden oder es muss eigenständig eine custom Messages aus Grundtypen und/oder anderen Messages definiert werden. Der Empfänger, auch subscriber genannt, kann nun durch Angabe des Topics und des Nachrichtentyps auf diese öffentliche Adresse zugreifen und die Nachrichten lesen. Es können beliebig viele Nodes auf ein Topic senden und empfangen, jedoch muss der Nachrichtentyp hierfür übereinstimmen. Somit bildet der Nachrichtentyp implizit die Schnittstelle der verschiedenen ROS2 Nodes. [26]

- Rviz2

Rviz2 ist eine grafische Oberfläche, die zur Visualisierung verschiedener ROS2-Daten verwendet wird. Dies ermöglicht die Anzeige von zum Beispiel Roboter,

Sensordaten, Karten. Im Rahmen der Entwicklung des ADP wurde dieses Programm als visuelle Hilfestellung verwendet. Aufbauend darauf wurde es ebenfalls in dieser Arbeit als visuelle Hilfestellung verwendet, sodass viele Darstellungen dieser Arbeit Aufnahmen aus dieser Software sind.

- CarMaker

CarMaker ist eine Simulationsumgebung, welche bevorzugt für die Simulation von PKWs und leichten Nutzfahrzeugen verwendet wird. Es handelt sich dabei um eine echtzeitfähige Physiksimulation von Fahrzeugen mit sehr realistischem Verhalten, vor allem was interne Prozesse der Fahrzeuge betrifft. Die Software ist auch in der Lage, verschiedene Sensoren zu simulieren. CarMaker wird primär dazu genutzt, geskriptete Szenarien abzuspielen und mit internen Tools ggf. das Fahrerverhalten zu parametrisieren. Es bietet zudem eine ROS2 Schnittstelle an, welche es Nutzern ermöglicht, in Echtzeit Daten auszulesen und einzuspielen. Da die Schnittstelle der ADP zum Fahrzeug durch eine einzelne Node durchgeführt wird, musste nur von dieser Node eine spezielle Version erstellt werden. Dadurch kommuniziert die Software 1:1 wie mit einem echten Fahrzeug. Desweiteren lassen sich mit CarMaker Sensoren simulieren, wodurch eine Objekterkennung, beispielsweise mit Radar, ermöglicht wird. Im Rahmen des ADP wurde dieses Feature jedoch nicht verwendet. Stattdessen wurde über eine Schnittstelle direkt auf die Simulationsdaten zugegriffen und diese dann in die benötigten Datenformate umgewandelt. Diese Lösung bietet keine korrekte Sensorsimulation, jedoch sind die so erhaltenen Daten absolut korrekt und enthalten keine Ungenauigkeiten. Dies ermöglicht eine effiziente Identifikation nicht sensorrelevanter Fehler innerhalb der entwickelten Software. Tests, die für die Sensorverarbeitung relevant sind, werden idealerweise erst nach erfolgreichem Abschluss aller anderen Tests in der Simulation mit echter Hardware durchgeführt. Eine zeitgleiche Unterstützung von gängigen Global Navigation Satellite System (GNSS)-Protokollen ermöglicht die Verwendung tatsächlicher GNSS-Daten. Dabei können beispielsweise Lanelet-Karten, welche GNSS-Daten gebunden sind, in CarMaker eingepflegt werden. Gleichzeitig können Werte wie die Fahrgeschwindigkeit an CarMaker gesendet werden. Somit ist es möglich, verschiedene Szenarien vollständig zu simulieren, ohne reale Tests mit verschiedenen Risiken oder hohen Kosten durchführen zu müssen.

3.2 Kommunikationsgrundlagen

- ETSI CPM Standard

CPM sind von der ETSI [27][28] standardisierte Nachrichten. Im ADP hat die "v2x_interface"-Node die Verantwortung, sämtliche V2X Nachrichten, darunter auch die CPM, zu verarbeiten. Die implementierte Schnittstelle, anhand derer Nachrichten ausgetauscht werden, hat jedoch nicht alle Datenfelder der ETSI CPM bereitgestellt. Der für diese Arbeit relevante Teil mit den Freirauminformationen ist hier nicht enthalten. Das Institut für Kraftfahrzeuge (Institut für Kraftfahrzeuge (IKA)) der RWTH Aachen bietet jedoch auf seiner Github-Page eine Sammlung von ROS2-Nodes und Messages an, um schnell und einfach mit einer OBU über ihre nativen User Datagram Protocol (UDP) Nachrichten zu kommunizieren. Darunter fällt auch die CPM inklusive der Freiraumdaten. Die vom IKA bereitgestellte Software bildet somit die Schnittstelle zwischen der in dieser Arbeit entwickelten Software und einer OBU. Ihre CPM-ROS2-kompatible Nachricht ist die Grundlage für das Empfangen von CPM-Nachrichten.

3.3 Projektkontext (Autonomous Driving Project)

Das Autonomous Driving Project wurde 2015 an der FH Aachen in Zusammenarbeit von Herrn Prof. Dr.-Ing. Reke und einigen Studenten gegründet. Das Ziel des Projekts ist die Entwicklung einer plattformunabhängigen Softwarearchitektur, die eine schnelle und einfache Autonomisierung verschiedener Fahrzeuge und sonstiger mobiler Roboter ermöglicht. Dies wird unter anderem dadurch ermöglicht, dass die ADP Software ein verteiltes System ist, welches auf dem zuvor vorgestellten ROS2-Framework basiert. Dadurch ist es möglich, einzelne Komponenten nach Belieben auszutauschen, zu entfernen oder neue hinzuzufügen. Diese werden im ROS2-Kontext "Nodes" genannt. Jede Node trägt eine abgeschlossene Verantwortung. Es gibt im Projekt grundlegend zwei Arten von Nodes: Fahrzeugabhängige sowie -unabhängige Nodes. Die fahrzeugabhängigen Nodes teilen sich wiederum in zwei weitere Kategorien auf; jene, die nur konfiguriert werden müssen und solche, die spezifisch gegen die Hardware implementiert werden müssen. Mögliche Beispiele für die Konfiguration sind die Ausmaße, die Radabstände, die Lenkwinkelübersetzung usw. Dadurch bleibt die interne Logik unverändert, während trotzdem ein breites Spektrum an verschiedenen Fahrzeugen kompatibler Klasse unterstützt wird. Die Entwicklung der ADP Software erfolgt in enger Zusammenarbeit mit der Hyuandi

Motor Europe Technical Center (HMETC). Diese stellte der FH Aachen einen Kia Niro, inklusive hochwertiger, nicht zur Serienausstattung gehörender Sensorik zur Verfügung, die spezifisch für das Projekt anwendungsdefiniert ausgewählt wurde. Parallel dazu besitzt die FH Aachen auch einen eigenen Kia Niro, mit deutlich günstigerer Sensorik. Dies ist ein ideales Beispiel für den ersten Teil, der hardware-spezifischen Nodes auf; für jeden Sensor gibt es mindestens eine Treiber-Node, die die Schnittstelle zwischen dem Sensor und der Anbindung ins ROS2-Framework bietet. Die Konfiguration der ADP-Software beider Fahrzeuge unterscheidet sich lediglich in den Nodes, die für die unterschiedliche Sensorik zuständig sind. Der Rest kann unverändert bleiben. Aufgrund der spezifischen Eigenschaften der Sensorik, ist der Kia Niro der HMETC in der Lage, alle Fähigkeiten, die durch die ADP-Software zur Verfügung gestellt werden, zu nutzen. Er verfügt über ein 360-Grad-Radar, das Pendant der FH Aachen hat hingegen nur einen temporär montierbaren Frontradar. Die Entwicklung der ADP-Software ist dadurch für diesen speziellen Fahrzeugtyp am weitesten fortgeschritten, speziell für den vollausgestatteten Kia Niro der HMETC. Der zweite Teil beschreibt die Schnittstelle zwischen Software und fahrzeugspezifischer Regelungshardware, der Steuereinheit. Im Falle des Kia Niro gibt es beispielsweise Drivekit oder SignalAuto. Für ein klassisches Auto wären dies z.B. die Regelung von Lenkwinkel, Gas-, Bremspedal und ggf. des Schaltgetriebes. Die Bezeichnung dieser Node entspricht der Klasse und der genauen Typbezeichnung. Für den Kia Niro der FH Aachen ist dieser damit `“ros2car_KiaNiro”`.

Zum Zeitpunkt dieser Arbeit ist die Basis aller Operationen im ADP eine statische GPS Karte, welche zuvor manuell erzeugt und überprüft wird. Die automatische Generierung ist bisher nicht möglich, weshalb sie nicht an unkartierten Orten eingesetzt werden kann. Diese Karte liegt in Form einer durch Lanelet2 erweiterten Open Street Map als `“.osm”` vor. Die Abhängigkeit der GPS Karte ist zudem an ein hochpräzises GPS-System gekoppelt, das auf wenige Zentimeter genau sein sollte, um eine sichere Fahrt zu gewährleisten. Das aktuelle System kann metaphorisch mit einem Zug verglichen werden. Er kann nur dort fahren, wo Schienen vorhanden sind und folgt diesen stur. Das ADP gesteuerte Fahrzeug folgt quasi `“virtuellen GPS Schienen”` und berücksichtigt ansonsten keine umliegenden Straßenmarkierungen, Schilder o.ä. Sind die Kartendaten falsch oder das empfangene GPS Signal zu ungenau, gibt es keine weiteren Korrekturen. Das aktuelle System verlässt sich also absolut auf die Kartendaten und die Werte des GPS. Darauf aufbauend, kann der Kia Niro Objekte per Radar erkennen und anschließend auf den Kartendaten in eine entsprechende Fahrspur zuweisen, wenn es sich innerhalb einer dieser aufhält. Unter der Annahme, dass Verkehrsteilnehmer nicht die Straße verlassen, ermöglicht es frühe-

re und bessere Prädiktionen über das Verhalten dieser Objekte zu machen und somit früher Entscheidungen zu treffen. Dies erhöht die Sicherheit, weil potenziell gefährliche Situationen früher erkannt und entsprechend behandelt werden können. Zudem erhöht es den Fahrtenkomfort durch die Reduzierung von starken Geschwindigkeitsreduktionen. Dies ist besonders wichtig für Autobahnauffahrten. Wie zuvor dargestellt, ist die Sensorreichweite begrenzt und von Hindernissen eingeschränkt, in dem spezifischen Fall oftmals Leitplanken und große Autobahnschilder. Zudem treten häufig hohe relative Geschwindigkeitsunterschiede auf, während ein Fahrzeug in die Autobahn auffährt. Es bleibt also wenig Zeit zu reagieren, als im normalen Verkehrsfluss in der Stadt oder auf der Autobahn. Im Rahmen eines Hi-Drive Projectes in Zusammenarbeit mit HMETC, wurde hier auch bereits die V2V basierten MCM implementiert, die ähnlich wie CPM, den Austausch von Informationen zwischen autonomen Fahrzeugen ermöglicht. In einer MCM sind jedoch nur Fahrzeugeigene Daten enthalten, speziell die aktuell gefahrene Trajektorie. Diese besteht in der Regel aus 30 Elementen, welche jeweils einen Zeitpunkt, eine Position und eine Geschwindigkeit in der Zukunft bereitstellen. Zudem gibt es noch eine "gewünschte" Trajektorie. Diese wird nur bei bestimmten Manövern gesendet, z.B. wenn ein Spurwechsel in eine blockierte Spur stattfinden soll. Blockiert bedeutet, dass zum aktuellen Zeitpunkt ein Objekt in der Zielspur zu nah am sendenden Fahrzeug ist, und die interne Abfrage für Einhaltung des Sicherheitsabstandes für diese Spur nicht eingehalten ist. In diesem Szenario ist das blockierende Fahrzeug auch autonom und sendet und empfängt auch MCM. Die Fahrzeuge wissen so gegenseitig, was der jeweils andere gerade plant und treffen basierend darauf eigene interne Entscheidungen, ob und wie diese reagieren wollen. Dies erweitert den Einsatzbereich der ADP, allerdings gibt es noch eine wichtige Fähigkeit, die der ADP-Software fehlt. Wie bereits in n1.1 erwähnt, ist es notwendig Objektdaten im Kontext des erfassenden Sichtbereiches zu interpretieren. Das bedeutet aktuell, fällt die Sensorik aus oder sendet eine leere Objektliste, trifft die ADP Software möglicherweise unsichere Entscheidungen, die zu einer Kollision führen könnten. Besonders kritisch ist dies beim FH Niro, welcher kein 360 Grad Radar besitzt und die Software dies komplett ignoriert. Diese Arbeit legt die Basis, dieses Defizit zu beheben. Es ermöglicht die Fähigkeit, Freiraumdaten beliebiger Quellen im CPM-radial-shapes Format zu empfangen, zu verarbeiten und dem Restlichen System in einer internen Datenstruktur für Entscheidungen zur Verfügung zu stellen. Zusätzlich wird ein Entscheidungsalgorithmus für Kreuzungsbereiche implementiert, der es der ADP-Software erstmals ermöglicht, Freiraumdaten für Entscheidungen zu berücksichtigen.

4 Konzeption

4.1 Problemformalisierung

Eine Kreuzung ist "sicher" und die Fahrtfreigabe damit erteilt, wenn der sicherheitsrelevante Bereich über den gesamten Zeitraum der Durchfahrt frei ist. Der sicherheitsrelevante Bereich ist innerhalb der Kartendaten definiert. 4.1. Die pinken Linien zeigen die Fahrbahnränder und definieren so ein Fahrspursegment. Die grünen Linien zeigen alle rechten Fahrbahnränder, die blauen alle linken Fahrbahnbegrenzungen. Er erstreckt sich

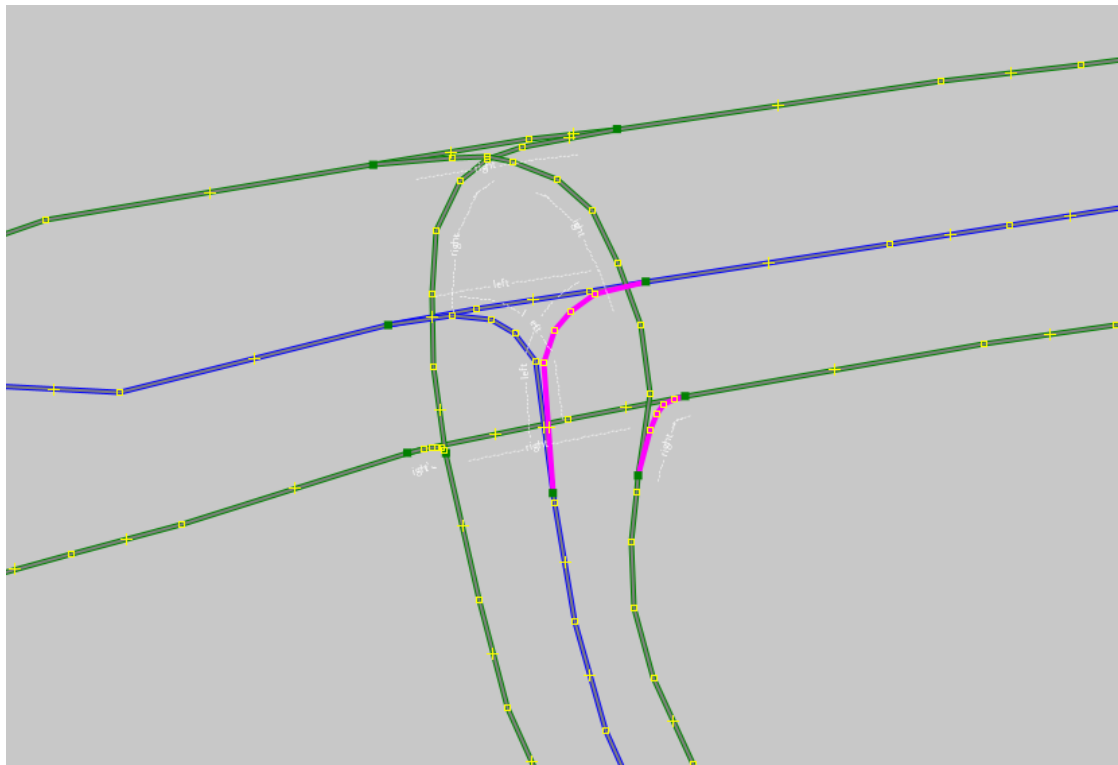


Abbildung 4.1: Bildschirmaufnahme: JOSM Lanelet2-Karten Editor

immer über ein ganzes Segment einer Fahrspur. Der Eintritt startet an der Fahrzeugfront und das Verlassen endet mit dem Fahrzeugheck. Das bedeutet, dass das Fahrzeug mit seiner kompletten Länge hinter dem fahrspurüberlagernden Bereich sein muss, sodass es einen theoretisch vorhandenen Verkehr nicht behindert. Folgende physikalische Berechnung definiert, wann der sicherheitsrelevante Bereich über den gesamten Zeitraum der Durchfahrt frei ist:

Alle Berechnungen basieren auf Worst-Case Annahmen. Im Rahmen dieser Arbeit sind diese wie folgt definiert: Es wird davon ausgegangen, dass alle Verkehrsteilnehmer regelkonform am Straßenverkehr teilnehmen. Die Fahrzeugspuren werden entsprechend ihrer regulären Fahrrichtung und mit maximal der zulässigen Geschwindigkeit befahren. Dies hat zwei Fälle zur Folge: Ein theoretisches Objekt fährt immer mit der Maximalgeschwindigkeit oder steht still.

Berechnung der worst-case zurückgelegten Distanz eines Verkehrsteilnehmers in überlappender Spur zum autonomen Fahrzeug.

t_1s = Zeit, die vergeht, bis das Fahrzeug von der aktuellen Position aus den Eintrittspunkt in den sicherheitsrelevanten Bereich erreicht

t_2s = Zeitspanne, die das Fahrzeug benötigt, um den sicherheitsrelevanten Bereich zu durchfahren

$t_1s + t_2s = t_3s$ = Gesamter Zeitraum von aktueller Fahrzeug Position inklusive Durchfahrt, durch sicherheitsrelevantem Bereich.

$v_1m s^{-1}$ = Geschwindigkeit der überlagernden Spur mit aktueller Fahrzeugspur.

$t_3s * v_1m s^{-1} = s_1m$ = Distanz, die ein theoretischer Verkehrsteilnehmer bei Einhaltung der maximal angenommenen Fahrtgeschwindigkeit zurücklegt.

Bei einer Freigabe auf Basis dieser Entfernung, würde das autonome Fahrzeug mit dem theoretischen Objekt kollidieren, da sie nach Ablauf der Zeit den selben Ort erreicht hätten. Damit dies nicht geschieht, wird zusätzlich der gewünschte Sicherheitsabstand addiert. Dieser wird in Abhängigkeit der Maximalgeschwindigkeit und einem gewünschten Sicherheitsabstand in Sekunden gebildet. Die im Rahmen der Arbeit genutzte Formel dafür ist sehr simpel und wurde innerhalb der ADP öfter verwendet.

$$t_{4m} = t_{\text{Sicherheitsabstand}m}$$

$$s_2m = t_{4m} * v_1m s^{-1} = t_{\text{Sicherheitsabstand}m}$$

Daraus ergibt sich folgende Gesamtformel:

$$(t_1s + t_2s + t_{4m}) * v_1m s^{-1} = s_2m$$

s_2 beschreibt die gesamte Distanz, die frei sein muss, damit zum aktuellen Zeitpunkt, die Fahrfreigabe erteilt wird, also die Kreuzung als “sicher” definiert ist.

4.2 Freiraum-Repräsentation

Für die Freiraum-Repräsentation wird ein Ansatz auf Basis einer “Occupancy Grid” genutzt. [29] Die einzelnen Grids, im weiteren Verlauf als Felder bezeichnet, können “frei” oder “unbekannt” als Zustand haben. Im Rahmen der Arbeit ist der Zustand “besetzt” nicht notwendig, da alle Szenarien so gewählt wurden, dass keine Objekte vorkommen. Normalerweise ist das Grid dabei homogen vom Ursprung aus verteilt. Das Grid wird im Rahmen dieser Arbeit nicht homogen genutzt, sondern auf die internen Fahrspurdaten abstrahiert. Die Feld-Dimensionen entsprechen dabei der Breite der Fahrspur und der Auflösung, der entsprechenden Fahrspur. Diese besteht aus einer Liste an aufeinander folgenden Punkten, die Durchschnittlich einen Abstand von \tilde{m} zueinander haben. Dieser Ansatz wurde gewählt, weil der aktuelle Stand der ADP-Software eine Fahrspur nur als “frei” oder “nicht frei” erkennt, dementsprechend war es nicht notwendig hier in der Breite eine feinere Auflösung zu wählen. Diese Repräsentation ermöglicht es später im Freigabealgorithmus, schnell und einfach abzufragen, ob die notwendigen Bereiche “frei” oder “unbekannt” sind. Zudem garantiert die Positionierung der Felder an der Mittellinie, dass Freiraumdaten beliebiger Quellen, kongruent zueinander sind. Dies vereinfacht die Zusammenführung beliebig vieler Freiraumdaten. Wie das funktioniert wird in 4.5 erläutert. Im weiteren Verlauf dieser Arbeit wird diese Freiraum-Repräsentation als Fahrspurfelder bezeichnet. Sämtliche Freiräume werden in einem “Hauptfreiraum” gesammelt. Wie dies im Detail funktioniert, wird im nächsten Abschnitt erläutert. Der “Hauptfreiraum” ist die Basis für alle nachfolgenden Berechnungen, die Informationen aus dem Freiraum benöti-

gen. Der Hauptfreiraum wird alle 100ms in das ROS2-Netz gesendet. Dazu gehört auch der später vorgestellte Freigabealgorithmus.

4.3 Freiraum extraktion aus CPM

Im Rahmen der Arbeit ist es nicht notwendig alle Möglichkeiten des CPM Standarts für das Darstellen der Freirauminformationen zu nutzen. Die CPM besitzt dafür ein Datenfeld Namens "Region". Hier gibt es unterschiedliche Formen zur Auswahl, davon wurde nur eine implementiert, die so genannten "radial shapes". Bei diesen wurde auf das Verwenden des vertikalen Winkels verzichtet, die beschriebene Region ist damit nur 2D, was jedoch keine Auswirkung auf die Aussagekraft hat, aber die Komplexität beim Implementieren reduziert. Die "radial shapes" haben als Eigenschaften einen Horizontalen Start- und Endwinkel, sowie eine Entfernung. Die so entstehende Form ist also ein Kreisabschnitt. Zur weiteren Vereinfachung wird die Form als gleichschenkliges Dreieck interpretiert. Die "radial shapes" eignen sich gut für einen 360 Grad LiDAR, der im Rahmen der ADP Forschung in Aachen für zukünftige Tests mit echter verwendet wird. Diese Dreiecke werden aus der CPM Nachricht extrahiert und in Rviz visualisiert. Im nächsten Schritt, gilt es die so genannte "DreieckFreiraum" Repräsentation in die internen Fahrspurfelder umzuwandeln. Dazu wird jedes einzelne Feld geprüft, ob dieses komplett von einem oder mehreren Dreiecken überlagert wird. Ein Feld gilt also nur als "frei", wenn es komplett erkannt wird. Dies verschiebt die Entscheidung der Kreuzungsfreigabe weiter in Richtung Worst Case Betrachtung, da so eher ein eigentlich freies Feld als "unbekannt" belegt wird und erhöht dadurch die Sicherheit.

4.4 Freiraum Interpolation

Die lokalen und externen Freiraumdaten haben immer einen Erstellungszeitpunkt, der in der Vergangenheit liegt. Dies wurde bereits in 1.2 aufgezeigt. Die Freiraum Interpolation basiert auf einem physikalischen Ansatz, die zeitliche Verschiebung in die Vergangenheit, rechnerisch auszugleichen. Dies ist zum einen notwendig, um im nachfolgenden Schritt zwei Freiräume korrekt kombinieren zu können und zum anderen, damit der Freigabealgorithmus eine Aussage mit zeitlich aktuellen Daten treffen kann. Es ist auch möglich, den Zeitpunkt in die Zukunft zu interpolieren. müssen diese Zeitlich aneinander und an den aktuellen Zeitpunkt angeglichen werden. Die Interpolation kann auch genutzt werden,

um beliebige Zwischenzeitpunkte, auch in die Zukunft, zu berechnen. Dies ermöglicht es auch für einen gewissen Zeitraum, auch dann noch sichere Entscheidungen zu treffen, wenn keine neuen und / oder aktuellen Freiraumdaten vorhanden sind. Im Zuge der aktuellen Implementierung liegt die Senderate bei 10Hz, da dies die standard Konfiguration in der ADP ist. Für die Interpolation wird zunächst die zeitliche Differenz zum aktuellen Zeitpunkt gebildet. Die zeitliche Auflösung der CPM liegt im Millisekundenbereich. Wie zuvor erwähnt, gilt auch hier weiterhin die gleiche Annahme für den Worst Case aus 4.1. Im nächsten Schritt werden die zu interpolierenden Fahrspurfelder in Reihenfolge der Fahrtrichtung durchlaufen. Es werden sämtliche Felder gesucht, bei denen der Zustand vom aktuellen Feld "frei" ist und im nächsten Feld auf "unbekannt" wechselt. An diesen Stellen, wird im "frei" Feld mit folgender Formel berechnet, wie weit ein theoretisches Objekt aus dem "unbekannt" Feld in der Spur fahren würde:

$$t_{\text{Interpolationszeit}} \cdot v_{\text{Geschwindigkeit}} \text{ m s}^{-1} = s_{\text{Strecke}}$$

Wie viele Felder ab dem ersten freien Feld interpoliert werden müssen, wird über die jeweiligen Längen der einzelnen Felder Berechnet, bis die aufsummierte Distanz größergleich der s_{Strecke} ist. Im aktuellen Ansatz wird das letzte freie Feld auch auf "unbekannt" gesetzt, obwohl dieses nicht komplett verdeckt ist. Dies ist damit begründet, dass es im Zweifel mehr verfallen lassen kann, als zu wenig. Dies ist Equivalent zu einer höheren theoretischen Objektgeschwindigkeit, wodurch die worst-case Annahme nicht verletzt, sondern verstärkt wird. Würde das letzte Feld auf "frei" gelassen werden, so müsste dieser Fall entweder gesondert behandelt werden, z.B. durch merken der Restdistanz des Feldes. Diese könnte dann im gleichen Durchlauf abgefragt werden. Falls dies nicht getan wird, wäre dies Equivalent zu einer geringeren theoretischen Objektgeschwindigkeit. Dies würde die zuvor gesetzten worst-case Annahmen verletzen. Es könnte nun nicht mehr gewährleistet werden, dass ein freier Bereich wirklich frei ist, da ein theoretisches Objekt mit maximaler Geschwindigkeit schneller fahren könnte, als die Interpolation diese verfallen lässt. Bei sehr kleinen Interpolationsschritten und/oder sehr niedrigen Geschwindigkeiten könnte dieser in der aktuellen Implementation sogar im ersten freien Feld abbrechen. Das wäre der Fall, wenn die resultierende Distanz kürzer ist, als die Länge des entsprechenden Feldes. Basierend darauf werden nun alle folgenden Felder, die unter die s_{Strecke} fallen und "frei" sind auf "unbekannt" gesetzt. Für den umgekehrten Fall, wo ein Feld von "frei" auf "unbekannt" wechselt, muss aufgrund der Worst Case Annahme Objektgeschwindigkeit 0km/h, keine weitere Berechnung erfolgen.

4.5 Freiraum Kombination

Bei der Kombination können in der Theorie beliebig viele Freiräume parallel verglichen werden. Ob das betrachtete Feld im Resultat “frei” oder “unbekannt” ist, wird durch folgende Logik entschieden:

- (1) Wenn mindestens ein Feld frei ist, wird dies als frei gewertet.
- (2) Wenn alle Felder unbekannt sind, wird dies als unbekannt gewertet.

4.6 Entscheidungslogik (Freigabealgorithmus)

Um die in der Problemformalisierung 4.1 mathematisch definierte Freigabe zu erteilen, müssen zunächst die Entsprechenden Segmente des jeweiligen Kreuzungsbereichs geklärt werden. In Abbildung 4.2 ist dies veranschaulicht. In der Abbildung startet die Fahrzeugtrajektorie am rechten Rand. Sie biegt dann in die linke Spur ab. Bei diesem Vorgang werden zwei andere Spuren gekreuzt, in der Abbildung als (Segment 0) und (Segment 1) gekennzeichnet. Jedes Segment verfügt über einen Start- und Endpunkt. Zudem hat die gesamte Kreuzung, abhängig von der aktuellen Trajektorie auch jeweils einen Start- und Endpunkt. Diese definieren den gesamtbereich einer Kreuzung, in dem das autonome Fahrzeug nicht stehen bleiben darf, um andere Verkehrsteilnehmer nicht zu blockieren. Der Freigabealgorithmus hat nur zwei mögliche Zustände, “Freigabe erteilt” oder “Freigabe nicht erteilt”. Wenn die “Freigabe erteilt” ist, führt das autonome Fahrzeug seine Fahrt ganz normal weiter, wie es bereits zuvor in der Lage war. Ist die “Freigabe nicht erteilt”, bleibt das Fahrzeug vor dem Eintreten in den sicherheitskritischen Bereich stehen. Wie dieser Bremsvorgang umgesetzt wird, ist dem nachfolgenden Programm überlassen. Für die Durchführung der Evaluation wurde hier folgende Implementierung gewählt: Sobald die Distanz zur betroffenen Kreuzung geringer ist, als der benötigte Bremsweg, mit der gewünschten Entschleunigung, so wird diese umgesetzt. Für die Durchführung der Szenarien, ist die ausreichend, für eine theoretische Nutzung in der realen Anwendung, ist dies jedoch nicht ausreichend. Auch die Entscheidung, wann die autonome Fahrt abgebrochen wird, ist abhängig von dem nachfolgenden Programm und hat keine Relevanz für die Auswertung und ist daher nicht Implementiert. Die Formel, auf dessen Basis die Freigabe erteilt wird, wurde bereits in 4.1 vorgestellt.

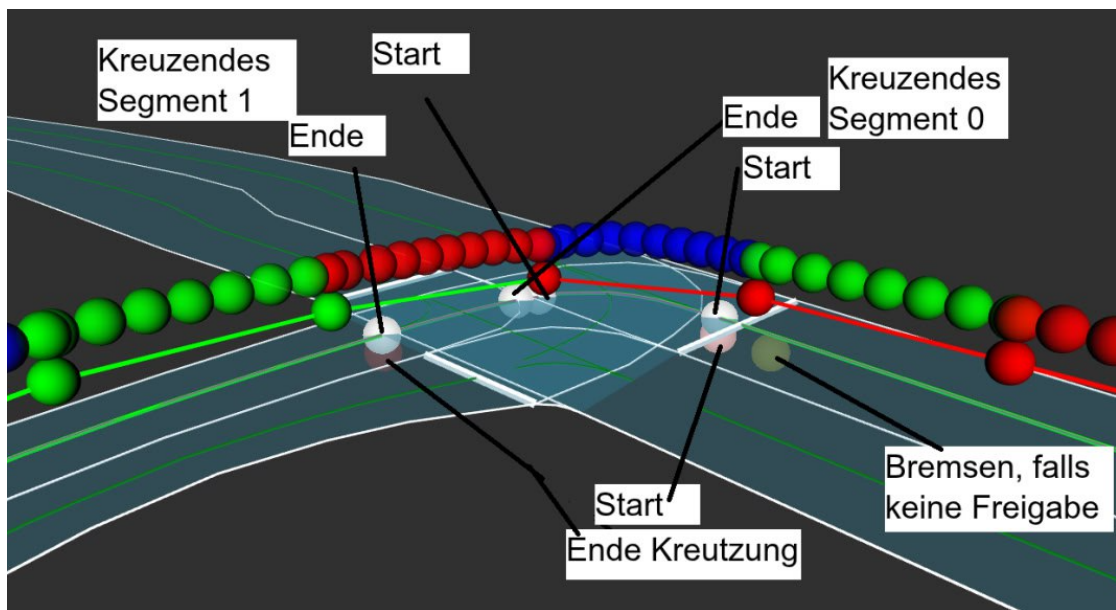


Abbildung 4.2: Kreuzung mit Debugvisualisierung während Entwicklung

5 Evaluation

5.1 Versuchsaufbau

- Versuchsumgebung:

Eine T-Kreuzung vor dem Kompetenzzentrum Mobilität (KMAC) in Aachen wird als Versuchsort genommen. Eine Abbildung dieser Kreuzung ist in 5.1 aus Fahrzeugperspektive zu sehen.

Diese Kreuzung bietet ein breites Spektrum an Herausforderungen für die Wahrnehmung



Abbildung 5.1: Bildschirmfoto: Google Maps T-Kreuzung gegenüber vom KMAC Aachen

autonomer Fahrzeuge, aufgrund der dortigen Verhältnisse. Die Kreuzung eignet sich also ideal für ein reales Szenario, um die zu testende Soft- und Hardware unter schwierigen Bedingungen zu validieren. Im Rahmen der in 3 erklärten Weiterentwicklung der ADP-Software ist deswegen geplant, dort mit dem echten Fahrzeug und entsprechender

Sensorik zu testen. Diese Arbeit ist die Erste, die auf diesem Szenario aufbaut und damit den Grundstein für nachfolgende Projekte legt. Dadurch können die Ergebnisse dieser Arbeit, weiteren darauf Aufbauenden und der späteren realen Umsetzung, aufeinander Aufbauend, verglichen werden. Folgende Gegebenheiten sind hier gegeben: Größere Höhenunterschiede der Fahrbahnen, welche eine Verdeckung verursacht. Zusätzlich stehen dort viele Bäume und Straßenschilder. Der Straßenverlauf der Straße, in die Abgebogen wird hat im relevanten Bereich eine leichte Krümmung, was die Einsicht zusätzlich erschwert. Auf der rechten Seite des Bildes 5.1 ist eine Wand zu sehen, die in Fahrtrichtung eine nicht unerhebliche Steigung verdeutlicht. Dies erschwert unter realen Bedingungen die Generierung des Freiraums, wie in 3 erklärt wurde und ist einer der Gründe für die Wahl der Kreuzung. Neben der Straße sind auch viele Bäume und Straßenschilder zu sehen, die den Freiraum weiter einschränken. Beide Straßen haben je einen Fahrstreifen pro Richtung und das Geschwindigkeitslimit liegt bei 30 km/h. Die Grafik 5.2 zeigt die



Abbildung 5.2: Rviz Bildschirmfoto: Startposition des Fahrzeugs für Szenarien

T-Kreuzung aus Sicht der ADP-Visualisierung. Die weißen Linien zeigen die Straßenbegrenzung der Kartendaten. Unten in der Mitte ist das autonome Fahrzeug zu erkennen. Beginnend mit der grünen Kugel darüber, gefolgt von mehreren blauen und vereinzelt weiteren grünen oder roten, ist die Trajektorie zu sehen. Grün : Fahrzeug beschleunigt,

blau: Geschwindigkeit halten, rot: Geschwindigkeit verringern. An der Trajektorie ist zu sehen, dass das Fahrzeug im Bereich der Kurve verzögert. Der Kurvenradius ist sehr klein, weswegen das Fahrzeug hier nicht mit 30km/h hindurchfahren kann. Auf der Grafik ist die entsprechende Laneletkarte dieses Szenarios zu sehen. Für den Versuch wurde in CarMaker ein entsprechendes Projekt erstellt. Dort startet das Fahrzeug ca. 150m vor der Kreuzung entfernt und kann somit problemlos auf die 30 km/h beschleunigen bevor es die Kreuzung erreicht. Der Abstand ist so gewählt, dass bereits ab dieser Entfernung relevante Daten durch den Freigabealgorithmus generiert werden. - **Versuchsobjekt:** Autonomes Fahrzeug mit 360 Grad LiDAR für Freiraumerkennung und RSU als externe CPM Quelle.

- Annahmen:

Da die RSU für die späteren realen Tests an der Fassade des KMAC befestigt ist, kann diese mit ihrem Sichtfeld nur eine Straße erfassen. Deswegen ist die Fahrtstrecke so gewählt, dass beim Abbiegen nur eine Spur gekreuzt wird, wodurch die Fahrtfreigabe auch nur einen relevanten Sichtbereich für die Freigabe einfordert. Die lokalen synthetischen Freiraumdaten simulieren eine vereinfachte Darstellung eines 360 Grad Lidars, weil dieser später für die realen Tests verwendet wird. Im Rahmen der Simulation wird der Höhenunterschied nicht berücksichtigt, weil die ADP-Software aktuell keine Unterstützung der Höhenunterscheidungen bietet. In der Simulation werden statische Hindernisse, die den Freiraum einschränken würden, nicht simuliert, sondern durch Parametrisierung der lokalen Freiraumdaten umgesetzt. Ob die Sensorreichweite durch ein simuliertes Objekt oder durch einen Parameterwert verringert wird, verhält sich äquivalent. Die Reichweite des resultierenden Freiraumbereiches ist dieselbe und der relevante Bereich kann nicht in Gänze erfasst werden. Sollte das Fahrzeug länger als 20 Sekunden vor der Kreuzung stehen, wird dies als ein TOR gewertet.

5.2 Szenarien

- Allgemeiner Versuchsablauf:

In allen Szenarien startet das Fahrzeug am selben Ort, ca. 150m entfernt. Durch die ADP-Software gesteuert folgt es dem Verlauf der Straße und versucht an der T-Kreuzung rechts abzubiegen. Für jedes Szenario wird jeweils ein zweiter Durchlauf durchgeführt, bei dem zusätzlich die externe CPM gesendet wird. Da diese in ihrer Parametrisierung nicht variieren hier die Erwartung für Szenario 1 bis 3 dem Referenz Szenario: Die Fahrtfreigabe

wird früh genug erteilt und bleibt ab diesem Punkt bestehen, sodass die Kreuzung ohne zusätzliche Verzögerung durchfahren werden kann. Sie verschlechtert das Verhalten in keinem der Fälle.

- **Eingangsgrößen / Variablen:**

Der lokale und externe Freiraum wird in Form einer CPM-ROS2 Nachricht mit statisch gesetzten Werten umgesetzt. Je nach Szenario wird die Reichweite für die lokale CPM über die statisch setzbaren Werte entsprechend angepasst. Die externen CPM hat immer eine Reichweite von 200m. Die Anzahl der "Dreiecke" sowie deren jeweiligen Start- und Endwinkel bleibt bei beiden ungeändert. Für die lokale CPM sind dies zwölf, für die externe nur ein einzelnes. Die Winkel der externen CPM sind so gewählt, dass die gesamte Straße links von der Kreuzung in ganzer Breite erfasst wird. Bei der lokalen CPM sind die Winkel so gewählt, dass sie 360 Grad komplett abdecken. Die Senderate des lokalen Freiraums beträgt immer 10Hz und unterliegt keinen künstlich simulierten Schwankungen. Für den externen Freiraum liegt diese bei nur 1Hz.

Szenario 1: lokaler Sensor erkennt nur Teilbereich des relevanten Bereiches

Parametrisierung: Die Reichweite des lokalen Sensors wird so konfiguriert, dass nur ein Teilbereich des relevanten Bereiches erfasst werden kann. Dies ist äquivalent zu einer Teilverdeckung durch ein Hindernis. Erwartung: Aufgrund der statischen Annahme wird im Fall der Verdeckung auch nach längerer Wartezeit keine Änderung eintreten. Dieses Szenario ist in 5.3 dargestellt. In der Aufnahme steht das Fahrzeug an der Haltelinie und wartet auf Fahrtfreigabe. Analog zu 5.4 müsste der Bereich frei sein, damit eine Fahrtfreigabe erteilt werden könnte. In Abbildung 5.3 werden die verschiedenen Möglichkeiten der Visualisierung für die gleiche Situation dargestellt; a) Kartendaten, relevanten Bereich, Fahrzeugtrajektorie und Kreuzungsmarkierungen; b) a + Fahrspurfelder; c) a + Dreiecksfreiraum; und d) Alle Elemente zusammen. Im weiteren Verlauf der Arbeit wird nur noch eine der vier visualisierten Darstellungen gezeigt.

Szenario 2: Worst-case lokaler Sensor erkennt nichts

Parametrisierung: Die Reichweite des lokalen Sensors wird komplett deaktiviert. Dies ist äquivalent zu einer kompletten Verdeckung durch ein Hindernis oder eines Totalausfalls des Sensors. Erwartung: Der relevante Bereich wird niemals erfasst. Aufgrund der statischen Annahme wird im Fall der Verdeckung auch nach beliebig langer Wartezeit keine Änderung der Sichtverhältnisse eintreten. Dieses Szenario ist in 5.4 dargestellt. In der Abbildung sieht man, wie das Fahrzeug an der Haltelinie auf eine Fahrtfreigabe wartet.

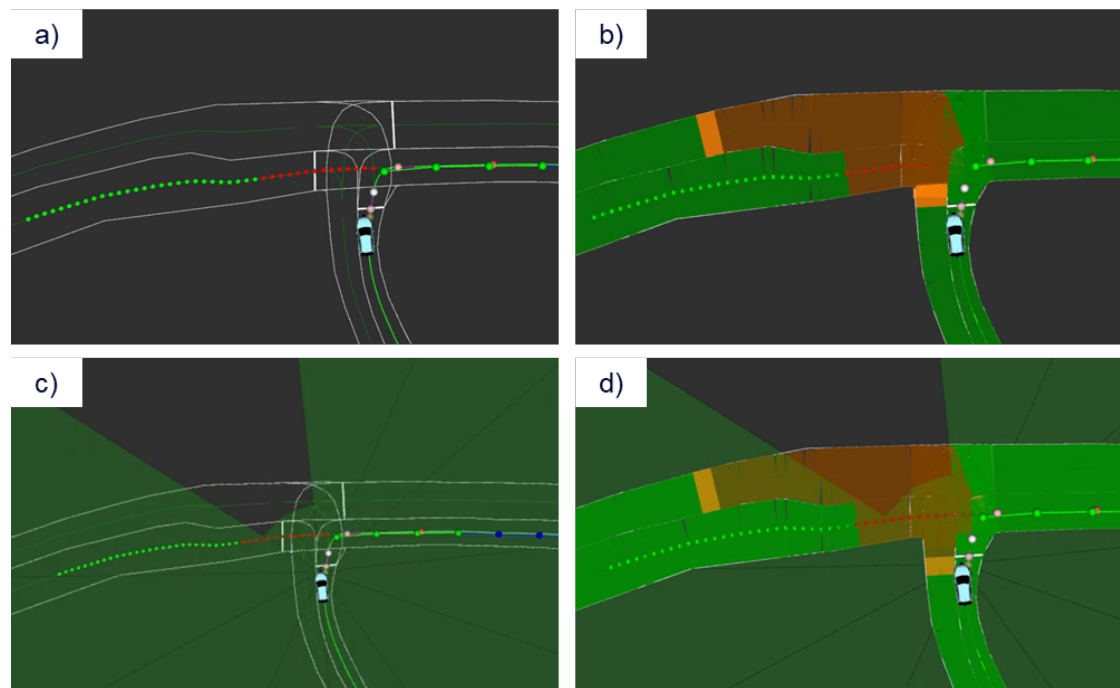


Abbildung 5.3: Rviz Bildschirmfotos: Sensor lokal teilblockiert, Fahrfreigabe nicht erteilt

Dieses Szenario sollte daher zu einem Abbruch der autonomen Fahrt durch einen TOR enden.

Szenario 3: lokaler Sensor erkennt relevanten Bereiches kurz vor Stillstand

Parametrisierung: Die Reichweite des lokalen Sensors wird sehr niedrig eingestellt. Dies ist equivalent zu extrem schlechten schlechten Sichtverhältnissen, wie sie in 30er-Zonen in sehr engen Straßen mit starker Beparkung durchaus häufiger vorkommen. Erwartung: Der relevante Bereich wird erst kurz vor Stillstand erfasst, die Freigabe wird noch gerade Rechtzeitig freigegeben. Das Fahrzeug kann wieder beschleunigen und die Fahrt fortsetzen.

Szenario Referenz: Lokaler Sensor hat freie Sicht

Parametrisierung: Die Reichweite des lokalen Sensors wird auf 150m gesetzt. Die Reichweite ist bei Optimalen Bedingungen realistisch. [30] Erwartung: Die Fahrfreigabe wird früh genug erteilt und bleibt ab diesem Punkt bestehen, sodass die Kreuzung ohne zusätzliche verzögerung durchfahren werden kann.

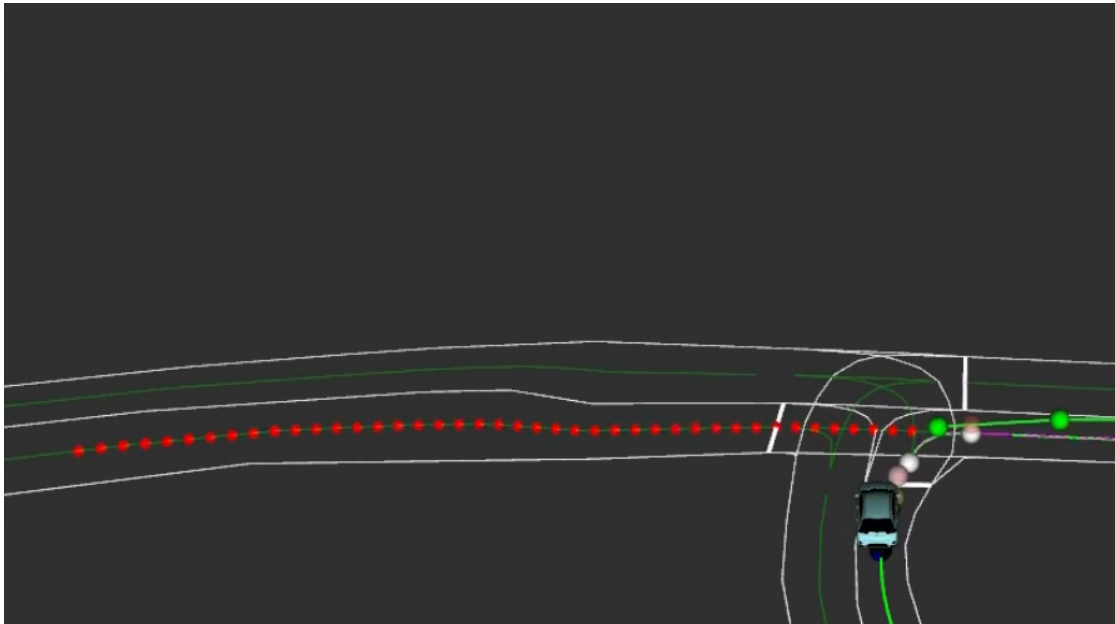


Abbildung 5.4: Rviz Bildschirmfoto: Fahrtfreigabe nicht erteilt

5.3 Metriken

- Messgrößen / Bewertungskriterien:

1. Löst die Fahrtfreigabe eine Bremsreaktion aus? ja oder nein
2. Kommt das Fahrzeug komplett zum Stillstand? ja oder nein
3. Gibt es einen kompletten Fahrabbruch? ja oder nein

5.4 Ergebnisse

Szenario 1:

Die Fahrtfreigabe wird zu keinem Zeitpunkt erteilt. Der Status einiger Felder des relevanten Bereiches ist über die gesamte Zeit “unbekannt”.

Szenario 2:

Die Fahrtfreigabe wird zu keinem Zeitpunkt erteilt. Der Status aller Felder des relevanten Bereiches ist über die gesamte Zeit “unbekannt”.

Szenario 3:

Die Fahrtfreigabe wird nicht erteilt, das Fahrzeug beginnt verzögert kurz vor der Kreuzung. Erst wenige Meter vor der Kreuzung, kurz bevor es zum Stillstand kommt, wird die Freigabe gewährt. Das Fahrzeug beschleunigt wieder und kann seine Fahrt fortsetzen.

Referenz Szenario: Ergebnis: Der relevante Bereich wird ab ca. 60m in gänze als frei erkannt und die Fahrtfreigabe wechelt ab dort auf frei und bleibt es bis zum Ende -> Erwartet und eingetreten.

Ergebnis für alle Szenarien mit externer CPM: Die Fahrtfreigabe ist zu beginn nicht erteilt. Erst ab einer Distanz von ca. 130m zu Kreuzung, wird die Freigabe erstmals erteilt. Diese wird im weiteren Verlauf aber immer wieder widerrufen. Erst ab einer Distanz von ca. 76m wird diese dauerhaft als frei erkannt.

5.5 Diskussion

Diese Arbeit untersucht, inwiefern externe Freiraumdaten die Entscheidungsqualität bei Sichteinschränkung lokaler Sensoren verbessern können, obwohl diese durch Latenz und Paketverlust beeinträchtigt werden.

Die gewählten Szenarien decken alle statischen Fälle der lokalen Sichtfeldeinschränkung ab. Diese sind in Bezug auf die Sichtbarkeit des relevanten Bereiches; Nie sichtbar - Immer nur Teilweise sichtbar - Erst sehr spät komplett sichtbar. Der Referenzfall soll zeigen, dass ohne Sichteinschränkung die Fahrtfreigabe generell funktioniert. Durch den Vergleich mit immer den gleichen Parametern bei den externen CPM Daten, können die Ergebnisse gut verglichen werden. Die große Latenz und die niedrige Senderate der externen CPM haben diese zudem eine starke Aussagekraft, da diese Werte in einer realen Anwendung deutlich geringer wären. [12] [2]

Die Entscheidung der Fahrtfreigabe hat nur zwei Zustände, erlaubt oder nicht erlaubt. Dadurch sehen die resultierenden Ergebnisse in ihrer Rohform recht unbeeindruckend aus, gleichzeitig ist aber zu berücksichtigen, dass genau dies im Kontext autonomer Fahrzeuge gewünscht ist. Sensordaten sind oftmals sehr speicher- und rechenintensiv, dass heißt je schneller man Komplexität reduzieren kann um an die entscheidenden, relevanten Informationen zu gelangen, desto besser. Außerdem liegt dadurch die Verantwortung über die Fahrzeugreaktion nicht im Rahmen dieser Arbeit und liegt auch nicht im Fokus dieser

Arbeit. Deswegen wurde auf Metriken mit Bezug auf das Bremsverhalten oder durch Geschwindigkeiten nicht einbezogen.

Die Ergebnisse zeigen, dass das Fahrzeug mit externer CPM bereits ab ca. 130m eine Fahrtfreigabe erhält, während es diese ohne CPM gar nicht oder erst kurz vor der Kreuzung bekommt. Dies deutet darauf hin, dass externe Freiraumdaten die Reaktionsfähigkeit des Systems bei eingeschränkter lokaler Sicht deutlich verbessern.

Der Ansatz ist bisher nur in der Simulation getestet wurden. Die Aussagekraft der Ergebnisse ist dadurch begrenzt, dass die Freiraumdaten synthetisch gesetzt wurden und keine echten Objekte im relevanten Bereich vorkommen. Zusätzlich unterstützt die aktuelle Implementierung keine spezielle Behandlung dynamischer Objekte, wodurch reale Verkehrssituationen nicht abgebildet werden. Daraus folgt, dass die Ergebnisse nicht direkt auf reale Kreuzungsszenarien übertragbar sind. Der Ansatz ignoriert Positionsunsicherheiten komplett. Es wurden teilweise stark vereinfachende Annahmen getroffen, vor allem im Bereich der zeitlichen Freirauminterpolation. Ihr zugrundeliegendes Bewegungsmodell berücksichtigt keine speziellen Fahrmanöver wie Spurwechsel in Mehrspurigen Straßen oder Objekte, die sich nicht immer entsprechend der Fahrtrichtung innerhalb ihrer Fahrbahnmarkierungen aufhalten. Andere Verkehrsteilnehmern wie Fußgängern oder Fahrradfahrern wurden komplett ignoriert. Die CPM wurden simpel wie möglich umgesetzt. Die zeitliche Interpolation, so wie die Vereinigung der Freiräume speichert keine Informationen über die Herkunft oder Entstehung dieser. Es kann nicht differenziert werden, ob auf alten, aktuellen, externen, lokalen oder vereinigten Daten entschieden wird.

Trotz dieser Einschränkungen zeigt die Arbeit, dass das Konzept der externen Freiraumintegration unter kontrollierten Bedingungen funktioniert und die Entscheidungsqualität messbar verbessert. Dies stellt eine notwendige Grundlage für weiterführende Untersuchungen und Weiterentwicklung.

Zukünftige Arbeiten sollten reale Sensordaten sowie dynamische Objekte integrieren, um die Übertragbarkeit der Ergebnisse auf echten Stadtverkehr zu prüfen. Darüber hinaus wäre eine Evaluation mit variierenden Senderaten der externen CPM sinnvoll, um robustere Aussagen über die Systemgrenzen treffen zu können.

6 Fazit und Ausblick

Die vorgestellte Arbeit zeigt, wie externe Freiraumdaten dazu verwendet werden können um Bereiche, die lokale Sensoren nicht erfassen können, trotz zeitlicher Unsicherheiten in Latenz und senderate für eine Sichere Fahrentscheidung unterstützen können. Dafür würden zwar eine Reihe an Annahmen getroffen, dies ist jedoch im Kontext des Arbeitsrahmens durch zusammenarbeit in einem Forschungsprojekt begründet, dass begrenzte Fähigkeiten besitzt und permanent weiterentwickelt wird. Somit dient diese Arbeit als Grundlage für mehr.

Beantwortung der Forschungsfrage:

Inwiefern verbessern externe Freirauminformation die Entscheidungsqualität autonomer Fahrzeuge bei eingeschränktem Sichtfeld ihrer Sensoren?

Durch die Erweiterung mit externen Freiraum Informationen, selbst wenn diese selten oder stark Verzögert ankommen, kann mit Hilfe der vorgestellten Laufzeitkompensation und der Vereinigung, die Entscheidungsqualität deutlich verbessern. Solange die relevanten Bereiche nach der Umwandlung als noch "frei" vorhanden sind, ermöglichen sie eine klare und sichere Entscheidung, basierend auf der deterministischen Fahrtfreigabe, unter den beschriebenen Annahmen. Dies verhindert konservative Reduktionen der Geschwindigkeit, komplettes stehenbleiben und warten oder einen kompletten Abbruch des autonomen Fahrens.

6.1 Ausblick

Diese Arbeit bietet die Grundlage für folgende Arbeiten:

1. LiDAR Sensor für Messung und Generierung echter Freiraumdaten.
2. OBU-Setup, dass CPM tatsächlich sendet und empfängt.
3. Messungenauigkeiten der Positionen betrachten und entsprechend behandeln.

4. Objektfelder der CPM integrieren und ihre Auswirkungen den Freiraum betrachten und entsprechend behandeln. Dies stellt die Vereinigung dieser Arbeit mit dem in der Motivation vorgestellten Referenzfall, dar.

Literaturverzeichnis

- [1] X. Gao, X. Zhang, Y. Lu, Y. Huang, L. Yang, Y. Xiong, and P. Liu, “A survey of collaborative perception in intelligent vehicles at intersections,” *IEEE Transactions on Intelligent Vehicles*, pp. 1–20, 2024.
- [2] J. Schulte-Tigges, M. Rondinone, B. Coll-Perales, M. Förster, M. Rumpf, J. Wachenfeld, and M. Reke, “Improving automated intersection crossing with collective perception using 5g v2x and mec applications: a prototype-based study,” in *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)*, 2026. submitted for publication.
- [3] B. Mourllion, A. Lambert, D. Gruyer, and D. Aubert, “Collaborative perception for collision avoidance,” in *IEEE International Conference on Networking, Sensing and Control, 2004*, vol. 2, pp. 880–885, 2004.
- [4] A. Rauch, F. Klanner, and K. Dietmayer, “Analysis of V2X communication parameters for the development of a fusion architecture for cooperative perception systems,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*, (Baden-Baden, Germany), pp. 685–690, 2011.
- [5] H.-J. Gunther, O. Trauer, and L. Wolf, “The potential of collective perception in vehicular ad-hoc networks,” in *2015 14th International Conference on ITS Telecommunications (ITST)*, IEEE, 2015. Copenhagen, Denmark.
- [6] H.-J. Günther *et al.*, “Realizing collective perception in a vehicle,” in *2016 IEEE Vehicular Networking Conference (VNC)*, 2016.
- [7] CAR 2 CAR Communication Consortium (C2C-CC), “Guidance for day 2 and beyond roadmap,” White Paper C2CCC_WP_2072, CAR 2 CAR Communication Consortium, 2021.
- [8] 5G Automotive Association (5GAA), “A visionary roadmap for advanced driving use cases, connectivity technologies, and radio spectrum needs.”

- [9] H. Feifel, B. Erdem, M. Menzel, and R. Gee, “Reducing fatalities in road crashes in japan, germany, and USA with V2X-enhanced ADAS,” in *Proceedings of the 27th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, no. 23-0082, 2023.
- [10] M. Gabb, H. Digel, T. Müller, and R.-W. Henn, “Infrastructure-supported perception and track-level fusion using edge computing,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019.
- [11] M. Klöppel-Gersdorf, F. Trauzettel, K. Koslowski, M. Peter, and T. Otto, “The Fraunhofer CCIT smart intersection,” in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021.
- [12] M. Buchholz, J. Müller, M. Herrmann, J. Strohbeck, B. Völz, M. Maier, J. Paczia, O. Stein, H. Rehborn, and R.-W. Henn, “Handling occlusions in automated driving using a multiaccess edge computing server-based environment model from infrastructure sensors,” *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 3, pp. 106–120, 2022. Accepted 2021, published in vol. 14 no. 3 (May 2022).
- [13] O. Knorz, J. Buyer, M. Sautter, J. F. Qureshi, E. Heinrich, M. Kocsis, R. Zöllner, D. Heid, M. Yazgan, S. Orf, S. Ochs, J. Amritzer, M. R. Zofka, J. M. Zöllner, P. Schneider, J. Berger, K. Garlichs, J. C. Müller, H. Freienstein, B. Gernert, and M. M. Motamedi, “Evaluation concepts and test scenarios for V2X-based autonomous driving,” in *2025 IEEE International Automated Vehicle Validation Conference (IAVVC)*, 2025.
- [14] M. Klimke, M. B. Mertens, B. Völz, and M. Buchholz, “A real-world evaluation of two cooperative intersection management approaches,” *IEEE Intelligent Transportation Systems Magazine*, 2026.
- [15] A. Pons, M. Codina, B. Cordero, J. Marias-i Parella, J. Casademont, J. Castells, and F. Vazquez-Gallego, “Real-world implementation of maneuver coordination service for unsignalized intersections,” in *2025 IEEE Vehicular Networking Conference (VNC)*, 2025.
- [16] J. Zimmermann, I. Llatser, M. Scherl, F. Wildschütte, and F. Hofmann, “Benefit evaluation of V2X-enhanced braking in view-obstructed crossing use cases,” in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, 2024.

- [17] J. Zimmermann, J. Mönnich, M. Scherl, I. Llatser, F. Wildschütte, and F. Hofmann, “Analyzing the performance of a V2X-enhanced braking system in real-world crash situations,” in *2025 IEEE Vehicular Networking Conference (VNC)*, 2025.
- [18] I. Llatser, T. Michalke, M. Dolgov, F. Wildschütte, and H. Fuchs, “Cooperative automated driving use cases for 5g v2x communication,” in *2019 IEEE 2nd 5G World Forum (5GWF)*, 2019. Corporate Research, Robert Bosch GmbH.
- [19] K. Moller, L. Schwarzmeier, and J. Betz, “From shadows to safety: Occlusion tracking and risk mitigation for urban autonomous driving,” in *2025 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1883–1890, 2025.
- [20] C. Zhang, F. Steinhauser, G. Hinz, and A. Knoll, “Occlusion-aware planning for autonomous driving with vehicle-to-everything communication,” *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 1229–1242, 2024.
- [21] F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, “Lanelet2: A high-definition map framework for the future of automated driving,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018.
- [22] J. Wang and T. Nordström, “Latency robust cooperative perception using asynchronous feature fusion,” in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–10, 2025.
- [23] S. Ren, Z. Lei, Z. Wang, M. Dianati, Y. Wang, S. Chen, and W. Zhang, “Interruption-aware cooperative perception for V2X communication-aided autonomous driving,” *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 4, pp. 4698–4714, 2024.
- [24] “Ros 2 documentation,” 2026.
- [25] Open Robotics, “Why ros 2?,” 2022.
- [26] Y. Liao, “Roboter-betriebssysteme: Ros2 bügelt schwächen aus,” 2020.
- [27] ETSI, “ETSI TS 103 324 v2.1.1 — Intelligent Transport Systems (ITS); vehicular communications; basic set of applications; collective perception service; release 2,” tech. rep., ETSI. Technical Specification (ETSI TS 103 324 V2.1.1).
- [28] ETSI, “ETSI TR 103 562 v2.1.1 (2019-12) — Intelligent Transport Systems (ITS); vehicular communications; basic set of applications; analysis of the collective perception service (CPS); release 2,” tech. rep., ETSI, 2019. TR V2.1.1 (2019-12).

- [29] J. Godoy, V. Jimenez, A. Artunedo, and J. Villagra, “A grid-based framework for collective perception in autonomous vehicles,” *Sensors*, vol. 21, p. 744, 2021.
- [30] J. Schulte-Tigges, M. Förster, and G. Nikolovski, “Benchmarking of various lidar sensors for use in self-driving vehicles in real-world environments,” *Sensors*, vol. 22, no. 19, p. 7146, 2022.

7 Anhang

7.1 Verwendete Hilfsmittel

In der Tabelle 7.1 sind die im Rahmen der Bearbeitung des Themas der Bachelorarbeit verwendeten Werkzeuge und Hilfsmittel aufgelistet.

Tabelle 7.1: Verwendete Hilfsmittel und Werkzeuge

Tool	Verwendung
Overleaf	Textsatz- und Layout-Werkzeug verwendet zur Erstellung dieses Dokuments

Erklärung zur selbständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort

Datum

Unterschrift im Original