

Automatische Aktualisierung digitaler Karten für die hochgenaue Lokalisierung autonomer Fahrzeuge

Automated Updating of Digital Maps for the Highly Accurate Localization of Autonomous Vehicles

Von der Fakultät für Maschinenwesen der Rheinisch-Westfälischen Technischen
Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der
Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von

Tobias Michael Quack

Berichter: Univ.-Prof. Dr.-Ing. Dirk Abel
Univ.-Prof. Dr.-Ing. Lutz Eckstein

Tag der mündlichen Prüfung: 25. September 2020

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

Zusammenfassung

Eine wesentliche Voraussetzung für autonomes Fahren ist die Verfügbarkeit einer genauen und zuverlässigen Fahrzeugeigenlokalisierung. Aktuell verfügbare Systeme basieren zum großen Teil auf satellitengestützten Verfahren, deren Leistungsfähigkeit insbesondere in anspruchsvollen innerstädtischen Verkehrsszenarien durch zahlreiche prinzipbedingte Störungen beeinträchtigt wird. Seit einigen Jahren wird daher bereits der Abgleich von Daten aus der Fahrzeugumfeldsensorik mit hochgenauen digitalen Karten für die Fahrzeuglokalisierung untersucht. Ein ungelöstes Problem bei diesem Ansatz betrifft die Aktualität des Kartenmaterials: In dynamischen, städtischen Verkehrsräumen ändern sich die vom Fahrzeug wahrnehmbaren Umgebungsmerkmale häufig, sodass auch eine als Referenz genutzte digitale Karte kontinuierlich aktualisiert werden muss.

Der Fokus dieser Dissertation besteht in der Entwicklung eines Systems zur automatischen Aktualisierung digitaler Karten für die hochgenaue Lokalisierung autonomer Fahrzeuge. Das System setzt ein vernetztes Verkehrsumfeld voraus, in dem Fahrzeuge und Infrastruktur miteinander kommunizieren. Dem infrastrukturseitigen Datenverarbeitungssystem kommt hierbei die Aufgabe zu, von Fahrzeugen bereitgestellte Sensordaten automatisiert auszuwerten, um eine digitale Karte des betrachteten Verkehrsraumes kontinuierlich zu aktualisieren. Kartiert werden alle Objekte, die mindestens über mehrere Minuten unbeweglich sind und somit für die Eigenlokalisierung nachfolgender Fahrzeuge genutzt werden können. Als wesentliche Methoden kommen ein graphenbasiertes Optimierungsverfahren und automatische Ausrichtungsalgorithmen für Laserscanner-Punktwolken sowie gitterbasierte probabilistische Kartierungsansätze zum Einsatz.

Neben dem infrastrukturseitigen Teil besteht der zweite Schwerpunkt der Arbeit in der Entwicklung des Fahrzeugsystems, das auf Basis der digitalen Karte eine präzise und robuste Lokalisierung in Echtzeit realisiert. Als wesentlicher Sensor wird fahrzeugseitig ein Laserscanner mit einem horizontalen Sichtfeld von 360° vorausgesetzt. Daneben werden weitere Sensordaten aus Inertial- und Raddrehzahlsensorik fusioniert, um die Schätzung der Fahrzeugposition und -orientierung mit einer Rate von 50 Hz bereitstellen zu können. Ein adaptiertes Partikelfilter mit dynamischem Fahrzeugmodell bildet die methodische Basis für die Fahrzeuglokalisierung.

Die Arbeit beinhaltet nach der Diskussion der Methoden für die Kartierung und Lokalisierung eine experimentelle Validierung des vernetzten Gesamtsystems in einem innerstädtischen Testszenario.

Abstract

A major prerequisite for autonomous driving is precise and reliable vehicle self-localization. Currently available solutions for this problem are mostly based on satellite navigation systems which suffer from several inherent deficiencies especially in challenging urban traffic scenarios. In recent years, research has therefore been conducted on matching sensor data from the vehicle's environment perception systems with high-definition digital maps in order to achieve accurate self-localization. One unsolved problem with this approach concerns the obsolescence of the map data in dynamic traffic environments. Particularly in urban scenarios, environment features detectable by the vehicle's sensor systems are regularly subject to change so that a digital map suitable as a reference for localization must be updated continuously.

The focus of this dissertation lies in the development of a system for automated updating of digital maps for the highly accurate localization of autonomous vehicles. The system requires a connected traffic environment in which vehicles and infrastructure are able to communicate. On the infrastructure side, a data processing system automatically evaluates sensor data provided by the connected vehicles in order to continuously update a digital map of the respective traffic area. The map should include all objects that remain stationary for at least several minutes and which are thus useable for self-localization of following vehicles. The key methods used here are graph-based optimization and automatic alignment of lidar point clouds as well as grid-based probabilistic mapping approaches.

The second emphasis of this work is the development of the vehicle system that achieves accurate and robust real-time localization based on the digital map. The main sensor on the vehicle side is a scanning lidar with a horizontal field of view of 360°. In addition, the system fuses data from an inertial sensor and wheel speed sensors in order to provide estimations for the vehicle position and orientation at a rate of 50 Hz. The key method for the vehicle localization is an adapted particle filter with a dynamic vehicle model.

After discussing the methods for mapping and localization, this dissertation also includes an experimental validation of the connected system in an urban test scenario.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	2
1.2	Ziel der Arbeit	4
1.3	Aufbau der Arbeit	5
2	Grundlagen	7
2.1	Fahrzeuglokalisierung	7
2.1.1	Satellitenbasierte Lokalisierung	7
2.1.2	Filterung und Zustandsschätzung	9
2.1.3	Simultane Lokalisierung und Kartierung	12
2.1.4	Ausrichtung von Punktwolken	15
2.2	Digitale Karten	21
2.2.1	Koordinatensysteme und Transformationen	21
2.2.2	Gitterbasierte Belegungskarten	22
3	Kartengestützte Lokalisierung	25
3.1	Anforderungen	25
3.2	Stand der Forschung	27
3.2.1	Kartierte Merkmale	27
3.2.2	Genutzte Sensorik	29
3.2.3	Lokalisierungsalgorithmen	31
3.3	Konzeption	32
3.4	Umsetzung	34
3.4.1	Prädiktionsmodell	35
3.4.2	Messupdate	36
3.4.3	Parallelisierung	39
3.4.4	Initialisierung	41
3.4.5	Parametrierung	43
3.4.6	Gesamtsystem Fahrzeug	45
4	Infrastrukturbasierter Kartendienst	47
4.1	Anforderungen	47
4.2	Stand der Forschung	49

4.2.1	Ausrichtung der Sensordaten	49
4.2.2	Kontinuierliche Aktualisierung des Kartenmaterials	51
4.3	Konzeption	53
4.3.1	Hardware und Implementierung	53
4.3.2	Digitale Karte	53
4.3.3	Rekonstruktion der Trajektorie	54
4.3.4	Aktualisierung der Karte	55
4.4	Umsetzung	56
4.4.1	GraphSLAM	57
4.4.2	Aktualisierung der Belegungskarte	64
4.4.3	Linienextraktion	71
4.4.4	Gesamtsystem Infrastruktur	76
5	Validierung	79
5.1	Testumfeld	79
5.2	Testfahrzeug	81
5.2.1	Sensorik	82
5.2.2	Prototypencomputer	83
5.3	Methodik	84
5.4	Ergebnisse	86
5.4.1	Kartierung	86
5.4.2	Lokalisierung	105
5.4.3	Gesamtsystem	118
6	Zusammenfassung	121
6.1	Ausblick	123
A	Anhang	125
A.1	Spezifikation der Sensorsysteme	125
A.2	Spezifikation des Echtzeitrechners	126

Abbildungsverzeichnis

2.1	Struktur des Graphen eines SLAM-Problems	14
2.2	Ausrichtung zweier Punktwolken eines LiDAR-Sensors	16
2.3	Aktivitätsdiagramm des ICP-Algorithmus	17
2.4	Abstandsmetriken bei ICP-Algorithmen	18
2.5	Koordinatensysteme für Kartierung und Lokalisierung	21
3.1	Überblick der Komponenten des Lokalisierungsalgorithmus	34
3.2	Kinematisches Fahrzeugmodell zur Bewegungsprädiktion	36
3.3	Geometrische Größen beim Messupdate	38
3.4	Zeitlicher Ablauf des Lokalisierungsalgorithmus	40
3.5	Entwicklung der Punktwolke bei Initialisierung ohne Startpose	42
3.6	Konvergenzverhalten der Lokalisierung	44
3.7	Zustände und Prozesse der Fahrzeugsoftware	45
4.1	Überblick der Komponenten des infrastrukturbasierten Kartendienstes	57
4.2	Geschätzte Fahrzeugtrajektorie nach erstem Optimierungsschritt	59
4.3	Kartenaufbereitung für Scan-Matching	61
4.4	Geschätzte Fahrzeugtrajektorie nach zweitem Optimierungsschritt	62
4.5	Anpassungen verschiedener Bereiche beim Occupancy Grid Mapping	66
4.6	Zeitlicher Gewichtungsfaktor eines Fahrzeugdatensatzes	67
4.7	Simulationsergebnis Belegungskarte nach Durchfahrt eines Fahrzeuges	70
4.8	Logarithmische Wahrscheinlichkeiten entlang eines Laserstrahls	72
4.9	Mittlere Strahlrichtungen bei Vorbeifahrt mit Messfahrzeug	73
4.10	Ablauf der Linienextraktion am Beispiel eines kastenförmigen Clusters	75
4.11	Datenverarbeitung der infrastrukturseitigen Kartierungssoftware	76
5.1	Luftbild des für die Validierung genutzten Testfeldes	80
5.2	IRT-Buggy mit Erweiterung für kartengestützte Lokalisierung	82
5.3	Datensatz der Trajektorien	85
5.4	Euklidische Abstandsfehler bei Trajektorienrekonstruktion	87
5.5	Euklidische Abstandsfehler bei Scan-Matching und ICP	88
5.6	Konvergenz gegen lokales Minimum bei der Anwendung von ICP	89
5.7	Geschätzte Fahrzeugorientierung aus GraphSLAM und GNSS-Referenz	90
5.8	Differenz Fahrzeugorientierung zwischen Scan-Matching und ICP	91

5.9	Fehler bei der Trajektorienrekonstruktion mittels Scan-Matching . . .	92
5.10	Trajektorienrekonstruktion mit erhöhtem Suchradius	93
5.11	Weitgehende Verdeckung eines kartierten statischen Objektes	94
5.12	Scan-Matching bei unkartierten Objekten	95
5.13	Zu kartierende, veränderliche Objekte	96
5.14	Stadien der Belegungskarte bei veränderlichen Hindernissen	98
5.15	Automatische Kartierung und Linienextraktion	99
5.16	Robustheit gegenüber bewegten Objekten	101
5.17	Fahrzeugtrajektorien bei kartengestützter Lokalisierung	106
5.18	Euklidischer Abstandsfehler bei kartengestützter Lokalisierung	107
5.19	Verläufe der euklidischen Abstandsfehler bei mehreren Simulationen .	108
5.20	Histogramm der euklidischen Tracking-Fehler	109
5.21	Positionsschätzung der Monte-Carlo-Lokalisierung	110
5.22	Orientierungsschätzung der Monte-Carlo-Lokalisierung	111
5.23	Histogramm der Konvergenzzeiten bei der Echtzeitlokalisierung	112
5.24	Testszenarien mit unvollständiger digitaler Karte	113
5.25	Verteilung der Partikelwolke bei schlechtem Konvergenzverhalten . . .	115

Tabellenverzeichnis

3.1	Anforderungen Kartengestützte Lokalisierung	27
3.2	Literaturübersicht Kartierte Merkmale	28
3.3	Literaturübersicht Genutzte Sensorik	30
3.4	Literaturübersicht Lokalisierungsalgorithmen	31
3.5	Parameter des Lokalisierungsalgorithmus	43
4.1	Anforderungen Infrastrukturbasierter Kartendienst	48
4.2	Literaturübersicht Ausrichtung Sensordaten	51
4.3	Parameter des Algorithmus zur Rekonstruktion der Trajektorie	63
4.4	Parameter des Kartierungsalgorithmus	69
4.5	Parameter des Algorithmus zur Linienextraktion	75
5.1	Kennwerte zur Genauigkeit der Kartierung eines Referenzobjektes	100
5.2	Gemittelter Rechenzeitbedarf Kartierung	103
5.3	Änderung der mittleren Positionsfehler bei unvollständiger Karte	114
5.4	Datenmengen der auszutauschenden Informationen	118
5.5	Zeitbudgets für Kommunikation und Datenverarbeitung	119
A.1	Spezifikation der Fahrzeugsensorik	125
A.2	Spezifikation des Fahrzeugrechners	126

Abkürzungen

5G	Mobilfunkstandard der 5. Generation
DGPS	Differential Global Positioning System
eCall	Emergency Call
ECEF	Earth Centered, Earth Fixed
EKF	Extended Kalman-Filter
ENU	East, North, Up
FIFO	First In – First Out
GNSS	Globale Navigationssatellitensysteme
GPS	Global Positioning System
GPX	GPS Exchange Format
GSM	Global System for Mobile Communications
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
LiDAR	Light Detection and Ranging
LLH	Latitude, Longitude, Height
LTE	Long Term Evolution
MCL	Monte-Carlo-Lokalisierung
MEC	Mobile Edge Computing
PCL	Point Cloud Library
RADAR	Radio Detection and Ranging
RANSAC	Random Sample Consensus
RTK	Real Time Kinematic
SAPOS	Satellitenpositionierungsdienst der deutschen Landesvermessung
SBAS	Satellite Based Augmentation Systems
SCP	Secure Copy
SLAM	Simultaneous Localization and Mapping
UKF	Unscented Kalman-Filter

Abkürzungen

UMTS	Universal Mobile Telecommunications System
V2X	Vehicle-to-everything-Verkehrsnetzwerk
WGS84	World Geodetic System 1984
WLAN	Wireless Local Area Network
XML	Extensible Markup Language

Symbolverzeichnis

Allgemein

α	Winkel Laserstrahl LiDAR-Sensor
β	Auflösung in translatorischer Richtung
δ	Auflösung in rotatorischer Richtung
μ	Mittelwert Gauss'sche Normalverteilung
σ	Standardabweichung
d_L	Entfernungsmessung eines LiDAR-Messpunktes
e	Fehlermaß
$l(x)$	Logarithmische Wahrscheinlichkeit eines Zustandes x
n_{Fzg}	Anzahl durchfahrender Fahrzeuge
$p(x)$	Wahrscheinlichkeit eines Zustandes x
T	Samplezeit
t	Zeit
w	Gewichtungsfaktor

Ausrichtung von Punktwolken

e_{p2p}, e_{p2e}	Fehlermetrik Punkt-zu-Punkt bzw. Punkt-zu-Ebene
M	Referenz-Punktwolke
R_x, R_y, R_φ	Suchraum in x , y und φ
\vec{t}	Transformation $(\Delta x, \Delta y, \Delta \varphi)$
v	Summierte Wahrscheinlichkeit einer Transformation
W	Wahrscheinlichkeitsfeld
w_j	Wahrscheinlichkeitswert der Zelle j innerhalb des Wahrscheinlichkeitsfeldes
Z	Gemessene Punktwolke

Fahrzeugmodellierung

ω_{hl}, ω_{hr}	Winkelgeschwindigkeiten linkes bzw. rechtes Hinterrad
d_{odo}	Distanz aus Fahrzeugodometrie
$e_d, e_{\dot{\varphi}}$	Fehlerterme für die Odometrie und die Gierrate
r_{hl}, r_{hr}	Abrollradien linkes bzw. rechtes Hinterrad

Filterung

Ω	Linienkarte
Υ	Partikelset
F	Zustandsübergangsmatrix
H	Beobachtungsmatrix
K_k	Kalman-Verstärkung
N	Partikelzahl
P_k	Zustandskovarianzmatrix
Q	Kovarianzmatrix Modellfehler
R	Kovarianzmatrix Messfehler
S_k	Kovarianzmatrix Messresiduum
\vec{u}_k	Eingangsvektor
w	Partikelgewicht
\vec{x}_k	Zustandsvektor
\vec{z}_k	Messvektor

Navigation

ψ	Breitengrad im erdfesten Koordinatensystem
λ	Längengrad im erdfesten Koordinatensystem
φ	Gierwinkel
h	Höhe gegenüber Referenzellipsoid im erdfesten Koordinatensystem
x	Koordinate im kartesischen Koordinatensystem
y	Koordinate im kartesischen Koordinatensystem
z	Koordinate im kartesischen Koordinatensystem

1 Einleitung

Autonomes Fahren ist in der heutigen Automobilentwicklung neben alternativen Antriebskonzepten eines der herausforderndsten technischen Ziele, an dessen Erreichung Automobilhersteller, Zulieferer und zahlreiche weitere Unternehmen mit hohem Forschungs- und Entwicklungseinsatz und umfangreichen Investitionen arbeiten [39]. Der Ersatz des menschlichen Fahrers durch Automatisierungssysteme verspricht zahlreiche gesellschaftliche und wirtschaftliche Vorteile: Zunächst bieten autonome Fahrzeuge das Potenzial, die Sicherheit im Straßenverkehr wesentlich zu erhöhen. Nach Angaben des statistischen Bundesamtes waren 2018 in Deutschland 88.8% der Verkehrsunfälle mit Personenschaden auf Fehlverhalten des menschlichen Fahrers zurückzuführen [79]. Diese Unfallursache entfällt, wenn alle Fahrentscheidungen von automatisierten Systemen getroffen werden. Ein weiterer wesentlicher Vorteil besteht in der Entlastung des Fahrers von der zeitaufwendigen und kognitiv anstrengenden Fahraufgabe. Die hierdurch gewonnene Zeit ermöglicht einen Komfort- und Produktivitätsgewinn. Auch im Hinblick auf die Energieeffizienz und die Reduktion von Treibhausgasemissionen werden positive Einflüsse der Fahrzeugautomatisierung erwartet [44]. Die Vermeidung von Brems- und Beschleunigungsvorgängen, die optimierte Routenwahl und die allgemeine Verbesserung des Verkehrsflusses sind positive Effekte, die von autonomen Fahrzeugen ausgehen können. Da sich diese Fahrzeuge besonders gut für Sharing-Modelle eignen, kann zudem eine Verringerung des Fahrzeugbestandes mit positiven Auswirkungen auf Wirtschaftlichkeit und Nachhaltigkeit der Mobilität erzielt werden.

Aufgrund der Vorteile des autonomen Fahrens ist in Deutschland eine klare politische Unterstützung der Technologieentwicklung für autonome Fahrzeuge festzustellen. Das am 21. Juni 2017 in Kraft getretene Gesetz zum automatisierten Fahren definiert rechtliche Rahmenbedingungen für Hersteller und Fahrzeugführer autonomer Fahrzeuge [63]. Umfangreiche Förderprogramme werden zudem von den Bundesministerien für Bildung und Forschung (BMBF), für Wirtschaft und Energie (BMWi) und für Verkehr und digitale Infrastruktur (BMVI) im Rahmen des „Aktionsplans Forschung für autonomes Fahren“ realisiert [16].

Die vorliegende Arbeit liefert vor diesem Hintergrund einen Beitrag zur Technologie für vernetzte, autonome Fahrzeuge. Der Fokus liegt hierbei insbesondere auf der Realisierung einer sicheren und robusten Fahrzeuglokalisierung in anspruchsvollen, städtischen Verkehrssituationen. Im Folgenden werden die Motivation und das Konzept der Arbeit dargestellt und die wesentlichen Ziele definiert.

1.1 Motivation

Autonome Fahrzeuge benötigen Systeme zur **genauen und zuverlässigen Eigenlokalisierung**, um sowohl auf der Navigations-, als auch auf der Bahnführungs- und Stabilisierungsebene die notwendigen Entscheidungsfindungs- und Regelungsprozesse durchführen zu können. Damit für diese kritischen Positionsinformationen eine hinreichende Sicherheit und Robustheit erzielt werden kann, werden Redundanzen gefordert; es müssen also mehrere unabhängige Systeme existieren, welche die benötigten Informationen bereitstellen können. Eine wichtige Rolle bei der Eigenlokalisierung spielen Globale Navigations satellitensysteme (GNSS). Da diese Systeme Signale nutzen, die von Satelliten in der Erdumlaufbahn emittiert werden, sind sie allerdings anfällig für Störungen aufgrund verschiedenster Mechanismen, darunter Ionosphärenfehler, Abschattungen und Mehrwegeeffekte. Zudem liegen die mit konventionellen GNSS-Systemen erreichbaren Genauigkeiten nicht in den Bereichen, die für die Bahnführungs- und Stabilisierungsaufgaben benötigt werden. Eine weitere wichtige Informationsquelle für die Fahrzeuglokalisierung ist die Detektion von Fahrspurmarkierungen, häufig auf Basis von Kamerasystemen. Entsprechende Lokalisierungsmethoden erreichen im Vergleich zu GNSS höhere Genauigkeiten und ermöglichen bereits die Umsetzung von einfachen Automatisierungsfunktionen wie Spurhaltesystemen. Die Eigenlokalisierung ist in diesem Fall jedoch stets nur relativ zu den Fahrbahnmarkierungen möglich und nicht absolut. Es kann also keine globale Position des Fahrzeuges auf der Erde angegeben werden, die für die Trajektorienplanung höherer Automatisierungsstufen jedoch zwingend notwendig ist. Aus diesen Gründen kommt der kartenbasierten Lokalisierung beim autonomen Fahren eine besondere Bedeutung zu. Durch den Abgleich von über die Fahrzeugsensorik erkannten Umgebungsmerkmalen mit Merkmalen, die in einer digitalen Karte georeferenziert gespeichert sind, kann die eigene Position und Orientierung mit hoher Genauigkeit ermittelt werden. Voraussetzung hierfür ist zum einen eine leistungsfähige Umfeldsensorik des Fahrzeugs, etwa Light Detection and Ranging (LiDAR), Kameras oder Radio Detection and Ranging (RADAR). Zum anderen ergeben sich hohe Anforderungen an die Genauigkeit, Vollständigkeit und Aktualität der genutzten digitalen Karten. [18], [74] Forschungsbedarf besteht daher einerseits hinsichtlich der Erstellung und der kontinuierlichen Aktualisierung digitaler Karten, die für die kartenbasierte Fahrzeuglokalisierung eingesetzt werden können. Andererseits werden die fahrzeugseitigen Algorithmen benötigt, die unter Verwendung dieser Karten eine hochgenaue und zuverlässige Lokalisierung in Echtzeit erzielen können.

Neben der Lokalisierung besteht eine weitere wichtige Voraussetzung für autonomes Fahren in einem semantischen Verständnis des Fahrzeugumfeldes. Hierunter wird verstanden, die Fahrzeugumgebung nicht nur mittels geeigneter Sensoren detektieren zu können, also sensorspezifische Rohdaten zu generieren, sondern aus diesen Daten zudem eine Zuordnung der Objekte zu diskreten Klassen zu erreichen. Diese semantischen Klassen beinhalten einerseits bewegte Verkehrsteilnehmer wie PKW, LKW, Fußgänger oder Rad-

fahrer, andererseits müssen auch Teile der Verkehrsinfrastruktur wie Lichtsignalanlagen, Verkehrsschilder und Fahrbahnmarkierungen als solche erkannt und klassifiziert werden. In Betracht kommt für diese Aufgaben sowohl die Auswertung der Sensorrohdaten des Fahrzeuges in Echtzeit, als auch die Nutzung kartierter Informationen in semantischen Ebenen digitaler Karten. [48] Auch bezüglich des semantischen Verständnisses des Fahrzeugumfeldes ergibt sich somit die Herausforderung, den Fahrzeugen Informationen über Änderungen im Verkehrsumfeld zur Verfügung zu stellen – etwa aufgrund der Einrichtung temporärer Hindernisse wie Baustellen. Bei der Nutzung digitaler Karten muss somit eine **kontinuierliche Aktualisierung der Datenbasis** sichergestellt werden.

Bei der Entwicklung und der rechtlichen Absicherung autonomer Fahrfunktionen spielen die Systemgrenzen eine große Rolle. Sie grenzen die Situationen, in denen das System bei bestimmungsgemäßem Gebrauch sicher genutzt werden kann von den Situationen ab, die vom System nicht sicher beherrscht werden können und in denen es nicht verwendet werden darf. [31] Bei aktuell verfügbaren oder in Entwicklung befindlichen Systemen der Autonomiestufen 2 (teilautomatisiert) und höher nach SAE J3016 [70] ist häufig eine Festlegung der Systemgrenzen auf Autobahnfahrten festzustellen [84]. Autobahnen bieten ein Verkehrsumfeld, das im Vergleich zu ländlichen oder städtischen Verkehrsräumen eine wesentlich geringere Komplexität aufweist. Im Hinblick auf die Fahrzeuglokalisierung treten hier viele der für GNSS problematischen Effekte – etwa durch Abschattung und Mehrwegeausbreitung – nur in deutlich verminderter Form auf. Zudem existiert nur eine vergleichsweise geringe Varianz an Fahrspurkonfigurationen, sodass auf Basis von GNSS und Fahrspurerkennung bereits eine für einfache Anwendungen hinreichende Eigenlokalisierung realisiert werden kann. In städtischen Verkehrsräumen führen dagegen die häufig dichte und hohe Bebauung, die Vielzahl an relevanten Objektklassen bei der Umfeldsegmentierung und die allgemein höhere Komplexität der Bewegungsmuster und Verkehrsregeln zu wesentlich größeren Herausforderungen bei der Umsetzung von Fahrzeugautonomie. Zugleich bieten die Bebauung und andere nicht bewegliche Objekte allerdings auch ein großes Potenzial für kartenbasierte Lokalisierungsansätze. Im Vergleich zu Autobahnanwendungen existieren wesentlich mehr für die Lokalisierung geeignete Objekte mit einer großen Variabilität an Formen und Geometrien. Die Entwicklung von Algorithmen für die kartenbasierte Lokalisierung ist also speziell für **städtische Szenarien** besonders vielversprechend, wo entsprechende Systeme einen Beitrag zur Ausweitung der Systemgrenzen autonomer Fahrfunktionen leisten können.

Eine weitere wesentliche Rolle bei der aktuellen Technologieentwicklung für autonome Fahrzeuge spielt die **Vernetzung zwischen Fahrzeugen und Infrastruktur**. Mit dem im Ausbau befindlichen Mobilfunkstandard der 5. Generation (5G) steht in naher Zukunft Kommunikationsinfrastruktur zur Verfügung, die hohe Datenraten, hohe Verfügbarkeiten und geringe Latenzen erzielt. Vorgesehen sind zudem spezielle Standards für Anwendungen in der Mobilität, insbesondere zur Vernetzung von Fahrzeugen und Infrastruktursystemen. Der Einsatz von sogenannten Micro- und Picozellen mit sehr geringen Reichweiten von wenigen 100 Metern ermöglicht unter Verwendung spezieller

Trägerfrequenzen die leistungsfähige Abdeckung von stark ausgelasteten innerstädtischen Bereichen. [15] Mit der Technologie Mobile Edge Computing (MEC) wird darüber hinaus eine Netzarchitektur etabliert, bei der direkt an den Mobilfunkzellen angebrachte Recheneinheiten in dezentraler Struktur Softwaredienste zur Verfügung stellen können [45] [52]. Das deutschlandweit erste Testfeld für diese Technologien wurde mit dem „Vodafone 5G Mobility Lab“ in Aldenhoven realisiert [85]. Vor diesem Hintergrund bieten sich enorme Chancen für die Auslagerung von rechenaufwendigen Algorithmen vom Fahrzeugsteuergerät auf infrastrukturbasierte Dienste. Die Bereitstellung und kontinuierliche Aktualisierung von digitalen Karten stellt dabei eine sehr vielversprechende Anwendung dar. Fahrzeuge können vom lokalen MEC-Rechner hochaktuelle Kartendaten für die nahe Umgebung erhalten und zugleich durch die Bereitstellung ihrer Sensordaten Informationen für die automatische Erkennung von Änderungen im Verkehrsumfeld an die Infrastruktur liefern.

1.2 Ziel der Arbeit

Die vorliegende Arbeit widmet sich den beschriebenen Herausforderungen und bestehenden Potenzialen durch die Entwicklung eines Systems zur Bereitstellung von automatisch aktualisierten digitalen Karten in städtischen Umgebungen. Das System baut auf einem mit moderner Kommunikationsinfrastruktur ausgestatteten Verkehrsumfeld auf, das Datenaustausch zwischen Fahrzeugen und intelligenter Verkehrsinfrastruktur mit hohen Datenraten und hoher Zuverlässigkeit ermöglicht und MEC-Funktionalitäten bereitstellt. Der wesentliche Anwendungsfall liegt in der Realisierung eines hochgenauen und robusten Eigenlokalisierungssystems für automatisierte Fahrzeuge, das auch in Bereichen mit mangelnder Zuverlässigkeit satellitenbasierter Lokalisierungssysteme die Position und Orientierung des Fahrzeuges bereitstellen kann. Als wesentliche Grundlage für die Lokalisierung wird ein Abgleich der digitalen Karte mit Echtzeitdaten der Fahrzeugsensoren, insbesondere eines LiDAR-Systems, genutzt. Die vom MEC-System verwaltete digitale Karte wird auf diese Aufgabe hin konzipiert und enthält charakteristische geometrische Objekte in zwei Layern. Ein statisches Layer beinhaltet Geometrien, die über lange Zeiträume unveränderlich sind und von Kartenanbietern zum Teil bereits heute angeboten werden [38], [81]. Daneben wird ein zusätzliches Layer umgesetzt, das für die Lokalisierung geeignete Objekte enthält, welche in kürzeren Zeiträumen verändert, bewegt und teils entfernt werden; Beispiele hierfür sind parkende Fahrzeuge. Dieses Layer wird auf Basis der Sensordaten von im kartierten Bereich fahrenden Fahrzeugen kontinuierlich aktualisiert.

Die Arbeit umfasst hiermit zwei wesentliche Säulen: Der fahrzeugseitige Teil des Systems nutzt die bereitgestellte digitale Karte für eine kartengestützte Echtzeitlokalisierung. Die Software fusioniert hierfür alle relevanten Sensordaten von in aktuellen Fahrzeugen

verbreiteten Sensorsystemen wie Odometriesensoren und einer Inertial Measurement Unit (IMU) mit Punktwolken eines 360° LiDAR-Systems und erreicht durch den Abgleich mit der georeferenzierten digitalen Karte eine globale Lokalisierung. Ein Forschungsschwerpunkt liegt hier in der Umsetzung von Positionsupdates mit einer Frequenz von 50 Hz auf einem eingebetteten Prototypensteuergerät mit begrenzter Rechenkapazität.

Die zweite wesentliche Säule bildet der infrastrukturseitige Teil des Systems. Hier erfolgt die Filterung und Auswertung der von den Fahrzeugen bereitgestellten Sensordaten mit dem Ziel, Informationen über die zu kartierenden Objekte und deren geometrische Eigenschaften abzuleiten. Es kommen probabilistische Algorithmen zum Einsatz, um eine Robustheit gegenüber ungenauen Daten und unklaren Interpretationen der Daten zu erzielen. Durch die sukzessive Auswertung großer Datenmengen wird zudem eine statistische Mittelung und somit eine Verbesserung der Genauigkeit angestrebt.

Der Umfang der Arbeit beinhaltet die Konzeption und Implementierung der benötigten Algorithmen für die Kartierung und die kartengestützte Lokalisierung, die simulative Analyse und Parametrierung der Software und die experimentelle Validierung auf einem Testfeld für vernetzte, innerstädtische Mobilität.

1.3 Aufbau der Arbeit

In Kapitel 2 werden zunächst die Grundlagen der Fahrzeugeigenlokalisierung und der Erstellung digitaler Karten für autonome Fahrzeuge eingeführt. Es werden die grundlegenden Definitionen und Algorithmen erklärt, die im Rahmen des entwickelten Systems genutzt werden und weiterführende Literaturhinweise gegeben. Die nachfolgende Grundstruktur orientiert sich an den beiden wesentlichen Säulen der Arbeit: Kapitel 3 hat den fahrzeugseitigen Lokalisierungsteil zum Thema, während Kapitel 4 den infrastrukturbasierten Kartierungsteil beschreibt. Beide Kapitel verfügen über eine identische Unterstruktur, die mit einer Definition der Anforderungen an das jeweilige System beginnt. Anschließend wird ein umfassender Überblick über den relevanten Stand der Forschung gegeben und auf dieser Basis die grundlegende Konzeption des zu entwickelnden Systems abgeleitet. Schlussendlich wird im Detail auf die Umsetzung der einzelnen Komponenten des entsprechenden Systems eingegangen.

Die Validierung der Kartierungs- und Lokisierungsalgorithmen erfolgt auf Basis experimenteller Untersuchungen mit einem prototypischen Testfahrzeug auf einem Testgelände für urbane Mobilität. Das Testumfeld, das Fahrzeug und die für die Validierung genutzte Methodik werden in Kapitel 5 zunächst eingeführt, bevor die wesentlichen Ergebnisse im Detail dargestellt und diskutiert werden. Kapitel 6 liefert eine abschließende Zusammenfassung der Resultate und Schlussfolgerungen der Arbeit und einen Ausblick auf zukünftige Forschungsaktivitäten.

2 Grundlagen

Eine wichtige Basis für die vorliegende Arbeit bilden grundlegende Definitionen und Verfahren zur Lokalisierung von Fahrzeugen sowie zur Erstellung digitaler Karten. Die entsprechenden Technologien, Algorithmen und Konventionen werden daher im Folgenden eingeführt und erklärt.

2.1 Fahrzeuglokalisierung

Unter Fahrzeuglokalisierung wird die Aufgabe verstanden, die Lage eines Fahrzeuges relativ zu einem festgelegten Bezugssystem zu bestimmen. Je nach Anwendungsfall kann dabei im Fahrzeugkontext unterschieden werden zwischen einer globalen Lokalisierung, welche eine eindeutige Position auf der Erdoberfläche liefert und einer relativen Lokalisierung, welche die Fahrzeugposition im Verhältnis zu anderen im Allgemeinen nicht georeferenzierten Objekten wie etwa geparkten Fahrzeugen oder Fahrspurmarkierungen angibt. Für viele Aufgabenstellungen aus dem Bereich des automatisierten Fahrens ist die Bestimmung der zweidimensionalen kartesischen Koordinaten (x, y) des Fahrzeugschwerpunktes nicht ausreichend, sondern es wird auch der Orientierungswinkel φ benötigt. Der Fahrzeugzustand (x, y, φ) wird dann häufig als Pose bezeichnet.

Im nachfolgenden Abschnitt werden Grundlagen der Fahrzeuglokalisierung, die der vorliegenden Arbeit zugrunde liegen, eingeführt und die einschlägige Fachliteratur angegeben. Ein Überblick über verschiedene Lokalisierungsverfahren sowie eine Klassifizierung der Methoden kann [88] und [12] entnommen werden.

2.1.1 Satellitenbasierte Lokalisierung

Eine weit verbreitete Technologie zur globalen Positionsbestimmung auf der Erde sind satellitenbasierte Navigationssysteme, kurz GNSS. Zurzeit befinden sich verschiedene Systeme im Einsatz, darunter das US-amerikanische NAVSTAR Global Positioning System (GPS), das russische GLONASS, das System Galileo der Europäischen Union sowie das chinesische Beidou. Die grundlegenden Funktionsweisen der Systeme sind identisch: Die Positionsbestimmung des Nutzers erfolgt stets durch den simultanen Empfang der Signale

von mindestens vier Satelliten, wobei aus den sich ergebenden Signallaufzeiten auf die zugehörigen Distanzen, die sogenannten Pseudoranges, geschlossen werden kann. Durch die genaue Kenntnis der Satellitenbahnen kann der Empfänger aus der Triangulation mit den ermittelten Pseudoranges die eigene Position ermitteln. Da der Empfänger im Gegensatz zu den Satelliten, die mit hochpräzisen Atomuhren ausgestattet sind, nicht über eine hinreichend genaue interne Uhr zur korrekten Bestimmung der Signallaufzeiten verfügt, enthält das zu lösende Gleichungssystem für eine dreidimensionale Position vier Unbekannte, von denen eine den Uhrenfehler des Empfängers bezeichnet. [87] [90]

Die Genauigkeit und Zuverlässigkeit der Positionsbestimmung mittels GNSS wird von einigen Effekten negativ beeinflusst: Ionosphärenfehler treten auf, wenn die von den Satelliten ausgesendeten elektromagnetischen Wellen die Ionosphäre in der Höhe zwischen 50 und 1000 km oberhalb der Erdoberfläche durchqueren, wobei die Ausbreitungsgeschwindigkeit der Signale durch die hier auftretenden freien Elektronen verändert wird. Die so veränderten Signallaufzeiten sind abhängig von der Sonneneinstrahlung und somit von der Tageszeit, aber auch der aktuellen solaren Aktivität, sodass die Prädiktion der Fehler nur schwierig und unvollständig möglich ist. Zusätzliche Fehler treten auch in der unteren Schicht der Erdatmosphäre, der Troposphäre, auf, wo die veränderliche Zusammensetzung der Gase sowie die Konzentration von Wasserdampf Unterschiede in den Signallaufzeiten hervorrufen. Ein weiterer, unerwünschter Effekt kann durch Objekte in der Nähe des Empfängers hervorgerufen werden, welche die Satellitensignale reflektieren. Die vom Empfänger ermittelte Signallaufzeit ist bei diesem sogenannten Mehrwegeeffekt im Vergleich zum direkten Empfang des Signals erhöht, woraus deutliche Positionsfehler resultieren können. Weitere Quellen für Ungenauigkeiten bei der Positionsbestimmung resultieren aus Abweichungen bei der Bahnberechnung der Satelliten oder Ungenauigkeiten der Satellitenuhren. In Summe muss bei der satellitenbasierten Positionsbestimmung daher mit einem Positionsfehler von über 10 m gerechnet werden. [33]

Zur Verbesserung der erreichbaren Genauigkeit von GNSS werden verschiedene Ansätze verfolgt. Sogenannte Satellite Based Augmentation Systems (SBAS), darunter das europäische EGNOS, liefern Korrekturdaten zu den Fehlern der Satellitenuhren, den Satellitenbahnen sowie den Ionosphärenfehlern, sodass die Positionsgenauigkeit auf einen Bereich von etwa 3 m verbessert werden kann. Noch genauere Korrekturen können durch das Verfahren Differential Global Positioning System (DGPS) erzielt werden, bei dem eine Referenzstation mit exakt bekannter, stationärer Position die Abweichungen zwischen den gemessenen und den wahren Pseudoranges bestimmt und in Echtzeit an den bewegten Empfänger überträgt. Je nach Abstand zwischen Referenzstation und Empfänger lassen sich hiermit Genauigkeiten im Dezimeterbereich realisieren; unter Einbeziehung von Trägerphasenmesswerten sind sogar Positionsgenauigkeiten im Millimeterbereich möglich. [87]

Da trotz aufwendiger Korrekturverfahren die Positionsbestimmung mit GNSS nur dort möglich ist, wo eine ausreichende Anzahl von Satelliten direkt empfangen werden kann,

sind satellitenbasierte Systeme allein für die zuverlässige Lokalisierung automatisierter Fahrzeuge nicht hinreichend. Insbesondere in innerstädtischen Szenarien führt dichte Bebauung nicht selten zu einer Abschattung großer Bereiche des Himmels und Mehrwegeeffekte verschlechtern die Genauigkeit, sodass zusätzliche, von Satelliten unabhängige Lokalisierungsverfahren erforderlich sind.

2.1.2 Filterung und Zustandsschätzung

Die Lokalisierung von Fahrzeugen auf Basis von Daten verschiedener Sensorsysteme kann als spezielle Form eines Problems der Zustandsschätzung betrachtet werden. Die gesuchten, aber im Allgemeinen nicht direkt bzw. nicht fehlerfrei messbaren Zustände entsprechen dabei der Position bzw. Pose des Fahrzeuges. Der Lokalisierungsalgorithmus hat zum Ziel, Schätzungen für die Zahlenwerte sowie die zugehörigen Unsicherheiten dieser Zustände zu berechnen.

Im Folgenden werden die Grundlagen wesentlicher Algorithmen zur Zustandsschätzung eingeführt. Aufgrund seiner Einfachheit sowie der sehr weiten Verbreitung wird dabei zunächst auf das Kalman-Filter eingegangen. Im Anschluss werden daraus die Grundlagen des im Rahmen der vorliegenden Arbeit implementierten Partikelfilters abgeleitet.

2.1.2.1 Kalman-Filter

Das Kalman-Filter ist ein Schätzalgorithmus, welcher die auf Wahrscheinlichkeiten beruhende Systematik des rekursiven Bayes-Filters praktisch umsetzt. Dem Bayes'schen Verfahren entsprechend kann der Algorithmus in zwei wesentliche Schritte unterteilt werden: Der Prädiktionsschritt bestimmt eine Schätzung für den aktuellen Zustand \vec{x}_k auf Basis des Zustandes im vorherigen Zeitschritt \vec{x}_{k-1} sowie der Eingangsgröße \vec{u}_k , wobei ein Modell des Prozesses verwendet wird. Die durch die Prädiktion bestimmte Schätzung wird beim anschließenden Messupdate mit der Sensormessung \vec{z}_k korrigiert.

Eine wesentliche Eigenschaft des Kalman-Filters ist die Abbildung der im Systemmodell, in den Messdaten und in den geschätzten Zuständen auftretenden Unsicherheiten durch mehrdimensionale, Gaußsche Normalverteilungen, welche jeweils durch die zwei Parameter Mittelwert und Kovarianz definiert sind. In seiner einfachsten Form als lineares Kalman-Filter liegt dem Algorithmus zudem die Annahme zugrunde, dass sowohl das im Prädiktionsschritt genutzte Systemmodell, als auch das beim Messupdate eingesetzte Messmodell linear sind.

Für den Bereich der Fahrzeuglokalisierung stellen die Annahmen des linearen Kalman-Filters häufig eine recht starke Einschränkung dar, da sowohl die Kinematik von Kraftfahrzeugen, welche im Bewegungsmodell beschrieben wird, als auch die üblichen Messverfahren

häufig starke Nichtlinearitäten aufweisen. Daher wird in der Praxis häufig das erweiterte Kalman-Filter, auch Extended Kalman-Filter (EKF) genutzt, bei dem sowohl das Prozessmodell, als auch das Messmodell durch eine nichtlineare Gleichung beschrieben wird. Beide werden anschließend in jedem Zeitschritt linearisiert. (vgl. [80], [43], [76])

Im zeitdiskreten Fall berechnet sich die A-priori-Schätzung des Zustandes \vec{x}_k zum Zeitschritt k auf Basis des vorherigen Zustandes \vec{x}_{k-1} und der Eingangsgröße \vec{u}_{k-1} beim EKF somit direkt aus der nichtlinearen Funktion f :

$$\vec{x}_k = f(\vec{x}_{k-1}, \vec{u}_{k-1}). \quad (2.1)$$

Durch Linearisierung von f wird die Systemmatrix F ermittelt:

$$F = \left. \frac{\partial f}{\partial x} \right|_{\vec{x}_{k-1}, \vec{u}_{k-1}}. \quad (2.2)$$

Hiermit kann die A-priori-Schätzung der Kovarianzmatrix P_k wie folgt bestimmt werden:

$$P_k = F \cdot P_{k-1} \cdot F^T + Q. \quad (2.3)$$

Q bezeichnet dabei die Kovarianzmatrix des Prozessmodells.

Basierend auf der Messung \vec{z}_k wird anschließend der Korrekturschritt des Kalman-Filters durchgeführt, bei dem auf Basis der A-priori-Schätzungen \vec{x}_k und P_k die korrigierten A-posteriori-Schätzungen $\hat{\vec{x}}_k$ und \hat{P}_k berechnet werden.

$$\vec{y}_k = \vec{z}_k - h(\vec{x}_k) \quad (2.4)$$

$$S_k = H \cdot P_k \cdot H^T + R \quad (2.5)$$

$$K_k = P_k \cdot H^T \cdot S^{-1} \quad (2.6)$$

$$\hat{\vec{x}}_k = \vec{x}_k + K_k \cdot \vec{y}_k \quad (2.7)$$

$$\hat{P}_k = (I - K_k \cdot H) \cdot P_k. \quad (2.8)$$

Hierbei bezeichnet \vec{y}_k das Messresiduum, S_k ist die Kovarianz des Residuums, K_k ist als Kalman-Verstärkung bekannt und R enthält die Kovarianz des Messmodells. Analog zu F im Prädiktionsschritt wird auch H durch die Linearisierung der Messfunktion h bestimmt:

$$H = \left. \frac{\partial h}{\partial x} \right|_{\vec{x}_k}. \quad (2.9)$$

2.1.2.2 Partikelfilter

Ebenso wie das Kalman-Filter basiert auch das Partikelfilter auf dem rekursiven Bayes-Filter. Es nutzt also Wahrscheinlichkeiten zur Darstellung der auftretenden Unsicherheiten und verfügt über eine Komponente zur Prädiktion der Zustände anhand eines Systemmodells und eine Komponente zur Korrektur der Prädiktion auf Basis von Messdaten. Anders als das Kalman-Filter ist das Partikelfilter allerdings nicht auf Gaußsche Wahrscheinlichkeitsverteilungen und deren Parameter Mittelwert und Kovarianz festgelegt. Die Wahrscheinlichkeitsverteilungen werden hier stattdessen durch eine finite Anzahl an Samples, den sogenannten Partikeln, approximiert. Auf diese Weise können beliebige, insbesondere auch multimodale Wahrscheinlichkeitsverteilungen realisiert werden. Zudem können nichtlineare Prozess- und Messmodelle direkt genutzt werden, ohne durch die anschließende Linearisierung einen Genauigkeitsverlust in Kauf nehmen zu müssen. Beim Partikelfilter ist die Genauigkeit der durch die finite Anzahl an Partikeln realisierten Approximation abhängig von der Partikelzahl. Einer verbesserten Genauigkeit bei größerer Partikelzahl steht der steigende Rechenaufwand gegenüber, der daher bei praktischen Implementierungen im Allgemeinen deutlich über dem des Kalman-Filters liegt. (vgl. [25], [24], [36], [80], [43])

Algorithmus 1 zeigt eine Pseudocode-Darstellung der dem Partikelfilter zugrunde liegenden Rechenschritte. Erfasst sind die Operationen, die in einem diskreten Zeitschritt, hier beim Übergang vom Zeitschritt $k - 1$ zum Zeitschritt k , ausgeführt werden müssen.

Algorithmus 1 Zeitschritt Partikelfilter (Partikelset Υ_{k-1} , Input \vec{u}_k , Messgröße \vec{z}_k)

```

1: Erzeuge temporäres Partikelset  $\Upsilon_{TEMP}$  mit  $N$  Partikeln
2: for  $n = 1$  to  $N$  do
3:   Berechne mit Prädiktionsmodell und  $\vec{u}_k$  neuen Zustand  $\vec{x}_k^n$  für Partikel  $n$ 
4:   Überlagere  $\vec{x}_k^n$  mit zufälligem Modellfehler gemäß Wahrscheinlichkeitsverteilung
5:   Berechne Gewicht  $w_k^n$  des Partikels  $n$  aus Übereinstimmung mit Messung  $\vec{z}_k$ 
6:   Füge das Partikel  $\langle \vec{x}_k^n, w_k^n \rangle$  dem temporären Partikelset  $\Upsilon_{TEMP}$  hinzu
7: end for
8: for  $n = 1$  to  $N$  do
9:   Ziehe Partikel  $n$  aus  $\Upsilon_{TEMP}$ ; Wahrscheinlichkeit,  $n$  zu ziehen, entspricht  $w_k^n$ 
10:  Füge  $\vec{x}_k^n$  dem neuen Partikelset  $\Upsilon_k$  hinzu
11: end for
12: return Partikelset  $\Upsilon_k$ 

```

Ein wesentlicher Datentyp des Algorithmus ist das Partikelset Υ , das sich aus N einzelnen Partikeln zusammensetzt. Die Zahl N kann dabei zunächst als konstant angenommen werden, prinzipiell sind aber auch Implementierungen des Algorithmus mit über der Laufzeit variablen Partikelzahlen möglich und üblich. Jedes Partikel besitzt zum einen

eine Hypothese über den Zustand \vec{x} des Systems, zum anderen verfügt es über ein Gewicht w .

Jeder Zeitschritt des Partikelfilters hat nun zum Ziel, aus dem Partikelset Υ_{k-1} des vorherigen Zeitschrittes das neue Partikelset Υ_k zu ermitteln, wobei der Systeminput \vec{u}_k und die Messgröße \vec{z}_k des aktuellen Zeitschrittes verarbeitet werden. In Zeile 1 des Algorithmus wird hierfür zunächst ein neues temporäres Partikelset Υ_{TEMP} erzeugt, das die selbe Partikelzahl wie Υ_{k-1} aufweist. Anschließend werden in den Zeilen 2 bis 7 für jedes Partikel aus Υ_{k-1} das Prädiktions- und das Messmodell angewendet. Nach der Berechnung des Prozessmodells in Zeile 3 wird in Zeile 4 der Einfluss des Modellfehlers berücksichtigt: Für jedes Partikel wird ein Modellfehler addiert, dessen Wert sich aus einem zufälligen Ziehen aus der entsprechenden Wahrscheinlichkeitsverteilung ergibt. Zeile 5 enthält die sich anschließende Berechnung des zugehörigen Gewichts w_k^n des Partikels n , wobei die Wahrscheinlichkeitsverteilung des Messmodells benötigt wird. w_k^n ist hierbei definiert als die Wahrscheinlichkeit, die Messung \vec{z}_k zu erhalten unter der Voraussetzung, dass der Zustand \vec{x}_k^n des entsprechenden Partikels vorliegt: $w_k^n = p(\vec{z}_k | \vec{x}_k^n)$. Jedes auf diese Weise neu berechnete Partikel wird in Zeile 6 dem temporären Partikelset Υ_{TEMP} hinzugefügt.

Der zweite wesentliche Mechanismus des Partikelfilters, das sog. „Resampling“ erfolgt in den Zeilen 8 bis 11. Der Mechanismus kann als ein Aussortieren von Partikeln aufgefasst werden, wobei die Partikel entfernt werden, die den Zustand des Systems nicht gut abbilden. Als Merkmal für die Güte eines Partikels wird das Gewicht w verwendet. In den Zeilen 9 und 10 werden also aus Υ_{TEMP} zufällig Partikel gezogen und dem neuen Partikelset Υ_k hinzugefügt, wobei die Wahrscheinlichkeit, ein bestimmtes Partikel zu ziehen, durch sein Gewicht gegeben ist. Partikel mit großen Gewichten werden hierbei also möglicherweise mehrfach gezogen; Partikel, für die kleine Gewichte bestimmt wurden, werden mit größerer Wahrscheinlichkeit nicht gezogen. Das Resampling stellt somit sicher, dass im Partikelset vor allem solche Partikel verbleiben, deren Zustand gut mit dem Modell und der Messung übereinstimmt. Auf diese Weise wird der durch das Partikelset erzeugte Rechenaufwand auf die Bereiche hoher Wahrscheinlichkeiten konzentriert. (vgl. [36], [80])

2.1.3 Simultane Lokalisierung und Kartierung

In Ergänzung zu satellitenbasierten Lokalisierungsmethoden sowie Sensorfusionsansätzen, die zusätzliche Bewegungsdaten eines Fahrzeuges einbeziehen, kann eine weitere Verbesserung der Fahrzeuglokalisierung erreicht werden, wenn auch Informationen der Umfeldsensorik genutzt werden. Durch den Abgleich von Umgebungsmerkmalen aus einer digitalen Karte mit aktuellen Daten der Bordsensorik stehen zusätzliche Anhaltspunkte für die Positionsbestimmung zur Verfügung. Ist die digitale Karte im Vorhinein nicht

bekannt, so ergibt sich das sogenannte Simultaneous Localization and Mapping (SLAM)-Problem. Dieses bezeichnet die Aufgabe, zugleich eine Karte des Fahrzeugumfeldes zu erstellen und diese für die Verbesserung der Lokalisierungsgenauigkeit zu nutzen. Da die Genauigkeit der Lokalisierung die Qualität der erzeugten Karte beeinflusst und gleichzeitig die Genauigkeit der digitalen Karte wiederum einen wesentlichen Einfluss auf die Genauigkeit der Lokalisierung hat, liegt eine große Herausforderung bei der Konzeption von Algorithmen für das SLAM-Problem in der bestmöglichen Kombination und Ausnutzung der zur Verfügung stehenden Informationen.

Die aus der Literatur bekannten SLAM-Ansätze können vor diesem Hintergrund in zwei wesentliche Kategorien eingeteilt werden: Sogenannte Online-SLAM-Verfahren verfolgen das Ziel, die aktuelle Pose eines Fahrzeuges zusammen mit der aktuellen Karte kontinuierlich zu schätzen und auf Basis neu eingehender Sensordaten inkrementell zu aktualisieren. Übliche Algorithmen für Online-SLAM sind Kalman-Filter [20] oder Partikelfilter [34]. Beim Offline-SLAM-Ansatz wird hingegen ein vollständiger Satz aus Sensordaten im Nachhinein verarbeitet, wobei das Ziel ist, die vollständige Sequenz aus Fahrzeugposen sowie die vollständige Karte zu rekonstruieren. Ein wesentlicher Unterschied besteht somit darin, dass beim Offline-SLAM für die Bestimmung einer Fahrzeugpose nicht nur die vorangegangenen Sensorinformationen, sondern auch später aufgezeichnete Sensordaten einbezogen werden können. Die gesuchten Zustände – Fahrzeugtrajektorie sowie Kartenelemente – werden hierbei in der Regel auf Basis eines Optimierungsproblems ermittelt, das alle Sensordaten als Randbedingungen beinhaltet. Für die Formulierung dieses Optimierungsproblems hat sich die Strukturierung als Graph durchgesetzt, weshalb entsprechende Ansätze häufig unter der Bezeichnung GraphSLAM zu finden sind. [35] Ein umfassender Überblick über verschiedene SLAM-Verfahren ist in [19] gegeben. Für die vorliegende Arbeit ist insbesondere das Offline-SLAM-Problem von Bedeutung, weshalb im Folgenden vertiefend auf die Grundlagen von GraphSLAM-Verfahren eingegangen wird.

Graphenbasierte SLAM-Algorithmen können in zwei wesentliche Funktionsmodule unterteilt werden. Das sogenannte **Frontend** ist spezifisch für das genutzte Sensorsetup sowie für die Definition der digitalen Karte und erfüllt die Aufgabe, das Optimierungsproblem in Form eines Graphen aufzustellen. Abbildung 2.1 visualisiert beispielhaft einen solchen Graphen, der sich aus Knoten und Kanten zusammensetzt. Die Knoten bezeichnen hierbei die zu optimierenden Zustände. Einerseits sind dies die Fahrzeugposen zu allen betrachteten diskreten Zeitpunkten; andererseits werden so auch die für die Orientierung genutzten Kartenelemente, z.B. vom Fahrzeug detektierte Landmarken, modelliert. Die Kanten verbinden bestimmte Knoten miteinander, zu deren relativer Lage Sensordaten zur Verfügung stehen. Jede Kante enthält neben der gemessenen Lagebeziehung auch Informationen über die Unsicherheit der Messung in Form einer Informationsmatrix, die als Gewichtungsfaktor für den Einfluss der Kante auf den zu minimierenden Gesamtfehler interpretiert werden kann. Beispiele für Typen von Kanten eines SLAM-Graphen sind:

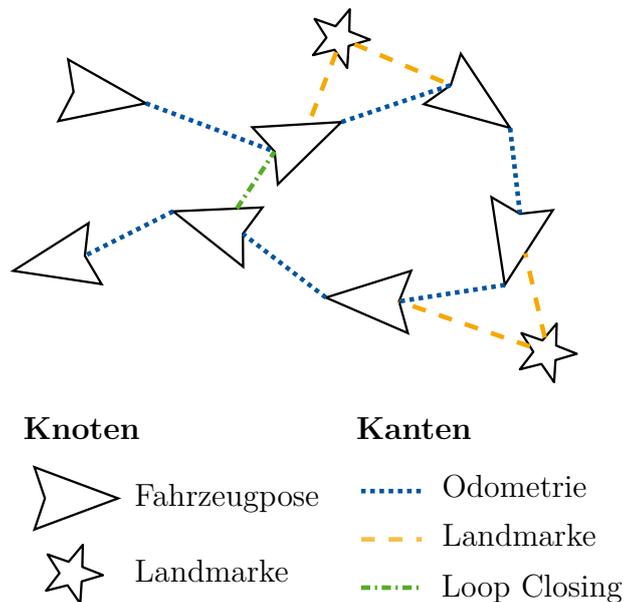


Abb. 2.1: Beispiel für die Struktur des Graphen eines SLAM-Problems

- **Odometrie:** Verknüpfung aufeinander folgender Knoten durch die Auswertung von Raddrehzahlsensoren
- **Inertialmessdaten:** Auswertung von translatorischen Beschleunigungs- und rotatorischen Drehrateninformationen ebenfalls zur Verknüpfung aufeinander folgender Knoten
- **GNSS:** Positionsschätzung des Fahrzeuges zu einem definierten Zeitpunkt
- **Landmarken:** Abbildung der Lage einer detektierten Landmarke relativ zur Fahrzeugpose zum Zeitpunkt der Beobachtung
- **Loop Closing:** Erfassung der relativen Transformation zwischen zwei Posen mit größerem zeitlichen Abstand bei Wiederkehr des Fahrzeuges in einen zuvor bereits befahrenen Bereich
- **Paarweiser Vergleich:** Ermittlung der Transformation zwischen zwei nicht notwendigerweise aufeinander folgenden Fahrzeugposes etwa durch das relative Ausrichten von Laserscans

Zur Vervollständigung des Optimierungsproblems wird nach der Definition der Knoten und Kanten des Graphen eine Initialisierung aller Zustände vorgenommen. Diese kann etwa auf der Prädiktion aller Posen mittels Odometrie-Messungen ausgehend von einer Startbedingung oder auf GNSS-Informationen beruhen.

Dem **Backend** wird der vom Frontend erzeugte Graph übergeben, sodass dort das zugrunde liegende Optimierungsproblem gelöst werden kann. Da die oben beschriebenen Randbedingungen des Graphen im Allgemeinen durch nichtlineare Funktionen beschrieben werden, erfolgt die Lösung üblicherweise durch iterative, numerische Verfahren. Ein weit verbreiteter Ansatz zur Lösung des nichtlinearen Optimierungsproblems besteht in der Anwendung des Gauß-Newton-Verfahrens. Ausgangspunkt für das Verfahren ist die Fehlerfunktion, welche durch die Summe der gewichteten Fehler aller Kanten definiert ist. Diese wird zunächst mittels einer Taylorentwicklung 1. Ordnung um den initialen Zustand linearisiert. Das so erzeugte lineare Least-Squares-Problem kann dann direkt gelöst und das sich ergebende Inkrement zur initialen Schätzung addiert werden. Diese drei Schritte – Linearisierung im Arbeitspunkt, Lösung des linearen Systems, Aktualisierung des Arbeitspunktes – werden bis zum Erreichen eines Abbruchkriteriums iterativ wiederholt. [47]

In praktischen Anwendungen kann die Anzahl an Knoten und Kanten und damit die Dimension des Optimierungsproblems sehr groß werden. Eine wesentliche Herausforderung liegt daher in der Realisierung von für die Anwendung hinreichend kleinen Rechenzeiten. Hierbei kann ausgenutzt werden, dass die sich ergebenden Matrizen des Optimierungsproblems aufgrund der speziellen Struktur des Graphen im Allgemeinen dünn besetzt sind. Üblicherweise beziehen sich die meisten Verknüpfungen innerhalb des Graphen auf aufeinanderfolgende oder nah beieinander liegende Zustände. Zudem erzeugen Kanten stets ausschließlich Abhängigkeiten zwischen genau zwei Knoten. Durch die Anwendung der Cholesky-Zerlegung können daher sehr effiziente Solver umgesetzt werden, die der dünn besetzten Systemstruktur Rechnung tragen. Ein Beispiel für eine frei verfügbare Softwarebibliothek für diese Aufgabe ist CHOLMOD [21]. Mit g2o existiert zudem ein vollständiges Framework zur Lösung von GraphSLAM-Problemen, das die vorgenannte Bibliothek integriert [47].

2.1.4 Ausrichtung von Punktwolken

Werden zur Unterstützung der Fahrzeuglokalisierung Daten eines Laserscanners oder Radarsensors eingesetzt, so liegen die Sensordaten üblicherweise in Form von Punktwolken vor. Jeder Punkt der Punktwolke besitzt dabei eine eindeutige zwei- oder dreidimensionale Koordinate und repräsentiert eine einzelne Abstandsmessung des Sensors. Um die Information der Punktwolken für die Lokalisierung nutzbar zu machen, kann durch paarweisen Vergleich zweier Datensätze diejenige Transformation \vec{t} gefunden werden, mit der beide Punktwolken optimal aneinander ausgerichtet werden können. Die so ermittelte Transformation lässt dann einen Rückschluss auf die Änderung der Sensorposition, also die relative Bewegung des Fahrzeuges zwischen beiden Zeitpunkten zu. Abbildung 2.2 zeigt beispielhaft zwei Punktwolken eines LiDAR-Sensors, die aus zwei verschiedenen Perspektiven in einem Parkhaus aufgezeichnet und anschließend ausgerichtet wurden.

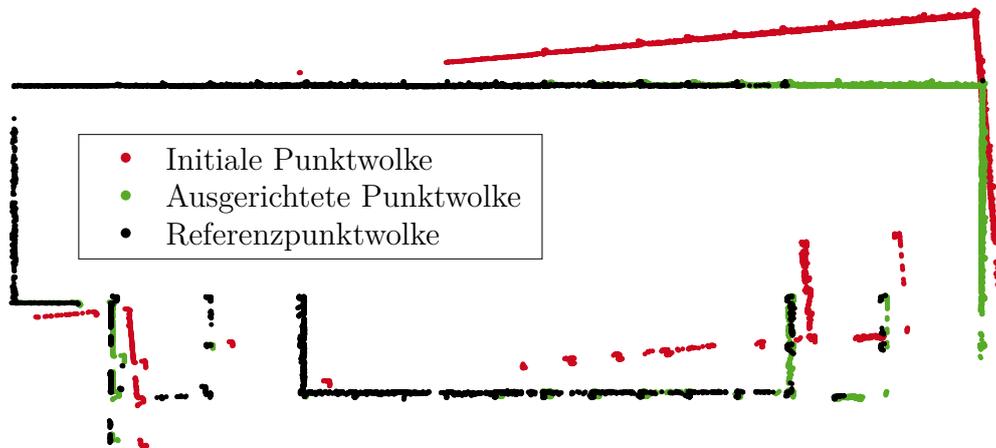


Abb. 2.2: Ausrichtung zweier Punktwolken eines LiDAR-Sensors

In der Literatur finden sich zahlreiche Ansätze zur rechenzeiteffizienten und präzisen Ausrichtung von Punktwolken. [11] liefert eine Übersicht der üblichen Verfahren und stellt spezifische Eigenschaften im Hinblick auf Genauigkeit und Rechenzeit gegenüber. Für die vorliegende Arbeit sind insbesondere der Iterative Closest Point (ICP)-Algorithmus sowie das direkte Scan-Matching von Bedeutung, weshalb im Folgenden die Grundzüge dieser Verfahren vorgestellt werden.

2.1.4.1 Iterative Closest Point-Algorithmus

ICP wurde ursprünglich in [13] eingeführt und beschreibt einen iterativen Algorithmus zur bestmöglichen Ausrichtung zweier Punktwolken einer als statisch angenommenen Umgebung. Die Qualität der Ausrichtung kann durch die Summe der Abstandsfehler zwischen korrespondierenden Punkten beider Punktwolken beschrieben werden. Sind die Korrespondenzen zwischen allen Punkten beider Punktwolken bekannt, so kann die Transformation zur optimalen Ausrichtung beider Punktwolken als Lösung eines linearen Least-Squares-Problems mittels einer Singulärwertzerlegung direkt berechnet werden. Da die optimalen Korrespondenzen allerdings im Allgemeinen nicht bekannt sind, nutzt der ICP-Ansatz ein iteratives Schema, das in Abbildung 2.3 in Form eines Aktivitätsdiagrammes dargestellt ist.

Ausgangspunkt für den Algorithmus sind die beiden auszurichtenden Punktwolken in ihrer initialen, nicht ausgerichteten Lage. Der Algorithmus ermittelt nun zunächst auf Basis einer einfachen Metrik Punktkorrespondenzen. Häufig wird hierfür ein „nächster Nachbar“-Ansatz genutzt; zu jedem Punkt der auszurichtenden Punktwolke wird also in der Referenzpunktwolke der Punkt mit der geringsten euklidischen Distanz ausgewählt.

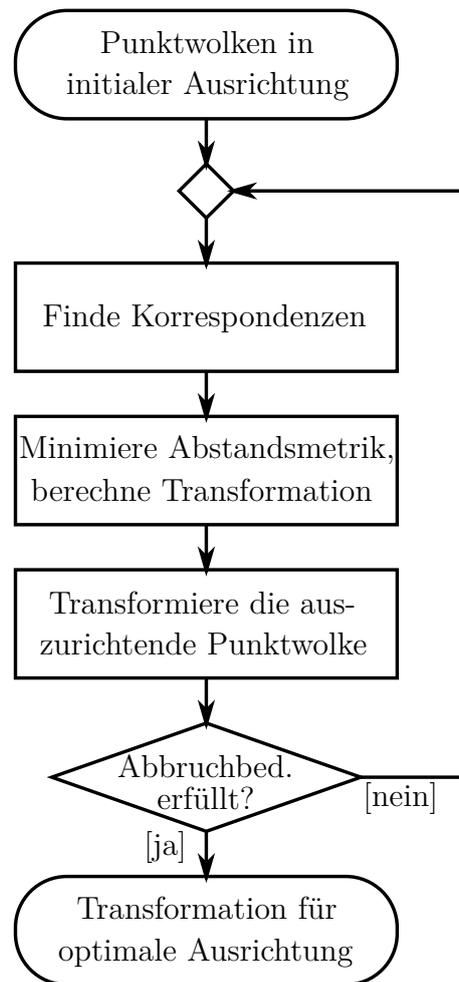


Abb. 2.3: Aktivitätsdiagramm des ICP-Algorithmus

Anschließend wird die Transformation berechnet, die die Summe der entsprechenden Abstandsfehler minimiert und somit für die angenommenen Korrespondenzen die bestmögliche Lösung darstellt. Die auszurichtende Punktwolke wird daraufhin mit dieser Transformation transformiert. Da die auf Basis der initialen Ausrichtung ausgewählten Korrespondenzen im Allgemeinen nicht optimal sind, erfolgt eine iterative Wiederholung der drei beschriebenen Hauptschritte: Zunächst werden also neue Korrespondenzen auf Basis der transformierten Punktwolken ermittelt; anschließend wird eine erneute Optimierung und Transformation der auszurichtenden Punktwolke vorgenommen. Ein Abbruchkriterium definiert schließlich das Ende des Algorithmus, etwa auf Basis einer maximalen Anzahl an Iterationsschritten oder der Unterschreitung einer Schwelle für die durch den letzten Iterationsschritt hervorgerufene Änderung. [68], [69]

Eine grundlegende Voraussetzung für die Anwendbarkeit von ICP ist die Verfügbarkeit einer näherungsweise korrekten initialen Ausrichtung. Ist diese zu ungenau, besteht die Gefahr, dass der ICP-Algorithmus nicht die optimale Ausrichtung ermitteln kann, sondern gegen ein lokales Minimum konvergiert.

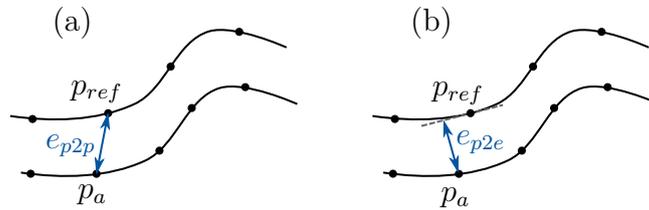


Abb. 2.4: Verschiedene Abstandsmetriken bei ICP-Algorithmmen. (a) Punkt-zu-Punkt, (b) Punkt-zu-Ebene

Verschiedene Varianten existieren bei ICP-Ansätzen für die Definition der zu minimierenden Abstandsfehler korrespondierender Punkte. Abbildung 2.4 zeigt schematisch zwei übliche Varianten: Bei der einfachsten „Punkt-zu-Punkt-Metrik“ wird für den Abstandsfehler die euklidische Distanz e_{p2p} zwischen den korrespondierenden Punkten p_a und p_{ref} gewählt. Die „Punkt-zu-Ebene-Metrik“ ermittelt für eine kleine Umgebung jedes Punktes der Referenzpunktwolke p_{ref} eine lokale Tangentialebene und berechnet sodann die senkrechte Distanz e_{p2e} des korrespondierenden Punktes der auszurichtenden Punktwolke p_a zu dieser Tangentialebene. Trotz geringfügig höheren Rechenaufwandes für die Lösung des sich ergebenden Optimierungsproblems wird hiermit häufig eine schnellere und robustere Konvergenz erzielt. [40]

Die freie Softwarebibliothek Point Cloud Library (PCL) enthält rechenzeiteffiziente Implementierungen aller üblichen ICP-Varianten [69].

2.1.4.2 Scan-Matching

Neben dem iterativen ICP-Ansatz kommt in der Literatur auch eine direkte Methode zur Ausrichtung von Punktwolken häufig zum Einsatz, die auf [60] basiert. Die grundlegende Idee besteht darin, die Transformation $\vec{t} = (\Delta x, \Delta y, \Delta \varphi)$ für die optimale Ausrichtung einer Punktwolke an der Referenz durch eine systematische Suche zu finden. Hierfür wird der Raum der möglichen Transformationen in den drei Koordinaten Δx , Δy und $\Delta \varphi$ diskretisiert und mittels der Brute-Force-Methode abgesucht. Das Grundprinzip besteht also darin, alle möglichen Transformationen zu prüfen und diejenige Transformation zu übernehmen, die zu der besten Übereinstimmung der Punktwolken führt. Der prinzipielle Ablauf dieses Scan-Matching-Verfahrens ist in Algorithmus 2 dargestellt.

Algorithmus 2 Scan-Matching (Referenz-Punktwolke M , gemessene Punktwolke Z , Suchraum R_φ, R_x, R_y)

```

1: Erzeuge Matrix  $W$  für das Wahrscheinlichkeitsfeld, Initialisiere mit 0
2: Erzeuge Vektor  $\vec{t}_{opt}$  für optimale Transformation, Initialisiere mit 0
3: Erzeuge Variable  $v_{opt}$  für optimale summierte Wahrscheinlichkeit, Initialisiere mit 0
4: for all Punkte  $m_i$  in der Referenz-Punktwolke  $M$  do
5:   for all Zellen  $j$  im Wahrscheinlichkeitsfeld  $W$  in der Umgebung von  $m_i$  do
6:     Berechne Wahrscheinlichkeit  $w_{j,TEMP}$  für Messecho in Zelle  $j$  aufgrund von  $m_i$ 
7:     if  $w_{j,TEMP} > w_j$  then
8:       Aktualisiere  $w_j$  mit  $w_{j,TEMP}$ 
9:     end if
10:  end for
11: end for
12: for all Rotationen  $\Delta\varphi_i$  im Suchraum  $R_\varphi$  do
13:   Transformiere Punktwolke  $Z$  mit  $\Delta\varphi_i$ 
14:   Projiziere transformierte Punktwolke  $\tilde{Z}$  auf Zellen von  $W$ 
15:   for all Verschiebungen  $\Delta x$  im Suchraum  $R_x$  do
16:     for all Verschiebungen  $\Delta y$  im Suchraum  $R_y$  do
17:       Transformiere Punktwolke  $\tilde{Z}$  um  $\Delta x$  und  $\Delta y$  durch Indexverschiebung
18:       Berechne summ. Wahrscheinlichkeit  $v_{TEMP}$  der akt. Transf. mittels  $W$ 
19:       if  $v_{TEMP} > v_{opt}$  then
20:         Aktualisiere  $v_{opt}$  mit  $v_{TEMP}$ 
21:         Aktualisiere  $\vec{t}_{opt}$  mit aktueller Transformation
22:       end if
23:     end for
24:   end for
25: end for
26: return optimale Transformation  $\vec{t}_{opt}$ 

```

Ein wesentliches Problem des Brute-Force-Ansatzes besteht darin, dass ein Kompromiss zwischen einer möglichst kleinen Diskretisierung des Suchraumes im Interesse einer präzisen Lösung und der Vermeidung einer übermäßigen Rechenzeit aufgrund zu vieler zu berechnender Transformationen gefunden werden muss. Hierfür sieht [60] mehrere Mechanismen vor, die das Verfahren prinzipiell für Anwendungen der echtzeitfähigen Lokalisierung nutzbar machen. Zunächst wird die Metrik betrachtet, anhand derer jede Transformation im Hinblick auf ihre Qualität bewertet wird. Optimal wäre die Berechnung der bedingten Wahrscheinlichkeit $p(\vec{z} | \vec{t}, M)$, die Messung \vec{z} zu erhalten, wenn die zu testende Transformation \vec{t} sowie die Referenzkarte M gegeben sind. Die Bestimmung dieser Wahrscheinlichkeit würde allerdings die Auswertung eines inversen Sensormodells für jeden Punkt der Punktwolke innerhalb der Brute-Force-Schleife über alle Transforma-

tionen bedingen. Wie Algorithmus 2 zeigt, wird dieser Rechenaufwand vermieden, indem die Vorabberechnung eines diskretisierten Wahrscheinlichkeitsfeldes (engl. „likelihood field“) in den zweidimensionalen, kartesischen Koordinaten der Referenzkarte erfolgt. Die Wahrscheinlichkeiten der Zellen des Wahrscheinlichkeitsfeldes werden dabei ausgehend von jedem belegten Punkt m_i der Referenzkarte ermittelt. Anhand der angenommenen Unsicherheit des genutzten Sensors wird bestimmt, wie hoch die Wahrscheinlichkeit ist, eine Messung in einer zu m_i benachbarten Zelle des Wahrscheinlichkeitsfeldes zu erhalten, wenn der Sensor eigentlich m_i detektiert hat. Ist das Wahrscheinlichkeitsfeld vollständig berechnet, so kann darin der Wahrscheinlichkeitswert jedes Messpunktes einer transformierten Punktwolke direkt abgelesen werden. Innerhalb der Brute-Force-Schleife muss damit lediglich das Auslesen aus dem Wahrscheinlichkeitsfeld als relativ kostengünstige Rechenoperation sehr oft durchgeführt werden.

Neben der Vorberechnung des Wahrscheinlichkeitsfeldes besteht ein weiterer, die Effizienz steigernder Mechanismus darin, die rotatorische Transformation $\Delta\varphi$ in der äußersten Schleife des Suchraumes durchzuführen, sodass diese Operation nur vergleichsweise selten anfällt. Ist eine Punktwolke bezüglich der Rotation in das Koordinatensystem der Referenzkarte transformiert, so kann der translatorische Suchraum mit wesentlich geringerem Rechenaufwand durchlaufen werden: Indem für das Wahrscheinlichkeitsfeld und die Diskretisierung des Suchraumes in Δx und Δy die gleiche Schrittweite definiert wird, entspricht das Absuchen der translatorischen Koordinaten einfachen Indexverschiebungen.

Ein letzter, wesentlicher Beitrag zur Echtzeitfähigkeit des Scan-Matching-Verfahrens besteht in der Wahl zweier unterschiedlich genauer Diskretisierungsstufen. Es wird demnach zunächst eine Brute-Force-Suche mit grober Diskretisierung durchgeführt, bei der Bereiche der Transformation \vec{t} identifiziert werden, in denen hohe Wahrscheinlichkeitswerte erreicht werden. Diese Bereiche werden anschließend mit der feineren Diskretisierung abgesucht, sodass zugleich eine hohe Genauigkeit des Matchings erzielt werden kann. Aus Gründen der Übersichtlichkeit ist dieser Mechanismus in Algorithmus 2 nicht erfasst.

Im Vergleich zum ICP-Verfahren bietet das direkte Scan-Matching mehrere Vorteile: Es ist nicht für die Konvergenz in lokale Minima anfällig und benötigt keine gute initiale Ausrichtung der Punktwolken. Nachteilig ist die durch die Diskretisierung prinzipbedingt begrenzte Genauigkeit sowie der trotz aller Optimierungen vergleichsweise hohe Rechenaufwand. Soll dieser reduziert werden, so darf der Suchraum nicht übermäßig groß sein, sodass in der Praxis ebenso wie beim ICP-Algorithmus auf eine grobe initiale Ausrichtung vertraut werden muss.

2.2 Digitale Karten

Ein wesentlicher Schwerpunkt der vorliegenden Arbeit liegt in der Nutzung digitaler Karten zur Verbesserung der Lokalisierung automatisierter Fahrzeuge. In den folgenden Abschnitten werden daher grundlegende Definitionen, Transformationen und Algorithmen eingeführt, auf denen das hier entwickelte System aufbaut.

2.2.1 Koordinatensysteme und Transformationen

Eine wichtige Grundlage für die Kartierung und Navigation sind Konventionen über die verwendeten Koordinatensysteme. In der vorliegenden Arbeit wird auf drei wesentliche Systeme Bezug genommen, deren Definition im Folgenden erläutert wird.

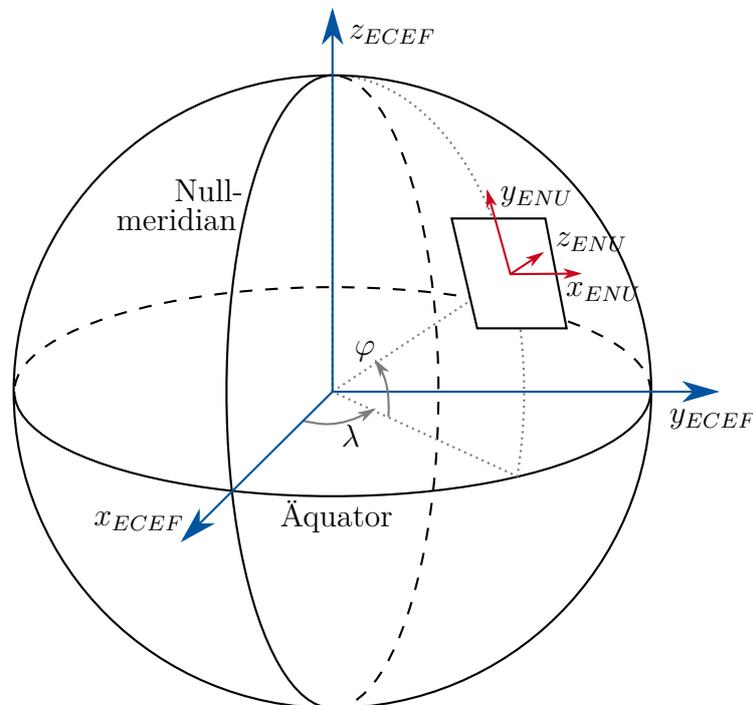


Abb. 2.5: Koordinatensysteme für Kartierung und Lokalisierung

Zur eindeutigen Beschreibung von Positionen auf der Erde wird ein Koordinatensystem verwendet, dessen Achsen im Bezug zur Erde fest sind. Das Earth Centered, Earth Fixed (ECEF)-Koordinatensystem ist ein kartesisches System, dessen Ursprung im Mittelpunkt des die Erde abbildenden Rotationsellipsoids liegt. Die x -Achse schneidet den Nullmeridian, x - und y -Achse liegen in der Äquatorialebene. Die z -Achse entspricht der Drehachse der Erde. Um Positionen in Latitude, Longitude, Height (LLH), also

in Form von Breitengrad ψ , Längengrad λ und Höhe h angeben zu können, ist ein Erdmodell erforderlich. In dieser Arbeit wird aufgrund seiner weiten Verbreitung sowie der Verwendung in GNSS das World Geodetic System 1984 (WGS84) genutzt. Das System definiert die Parameter eines Referenzellipsoiden mit dem Ziel, die Erdoberfläche möglichst genau abzubilden. Für die Umrechnung von LLH- in ECEF-Koordinaten werden die Parameter aus [23] verwendet. Die entsprechenden Formeln sind in [87] gegeben.

Für die lokale Navigation wird ein kartesisches Koordinatensystem genutzt, bei dem die x - und die y -Achse eine Ebene aufspannen, welche tangential zur Erdoberfläche in einem definierten Referenzpunkt liegt. Beim East, North, Up (ENU)-System ist hierbei die x -Achse in Richtung Osten definiert, die y -Achse weist nach Norden und die z -Achse weist senkrecht zur Erdoberfläche nach oben. Die Transformationen zwischen ECEF- und ENU-Koordinaten sind in [90] gegeben.

Abbildung 2.5 zeigt die Lage des ECEF-, des WGS84- sowie eines ENU-Koordinatensystems im Bezug zur Erde.

2.2.2 Gitterbasierte Belegungskarten

Eine verbreitete Methode zur Aufbereitung von Kartierungsinformationen, die auch in der vorliegenden Arbeit eingesetzt wird, ist die Erstellung gitterbasierter Belegungskarten. Diese auch als „Occupancy Grid Maps“ bekannten Karten unterteilen die kartierte Region mittels eines in der Regel äquidistant gewählten Rasters in diskrete Gitterzellen, von denen jede eine Belegungswahrscheinlichkeit p mit $p \in [0, 1]$ aufweist. Als „belegt“ gilt eine Zelle dann, wenn sich an der entsprechenden Stelle ein unpassierbares Objekt, etwa eine Häuserwand oder ein geparktes Fahrzeug, befindet. Zellen mit der Belegungswahrscheinlichkeit 0 werden als hindernisfrei und somit passierbar angesehen. Die probabilistische Betrachtung ermöglicht es, die Informationen mehrerer einzelner Beobachtungen – etwa mehrerer Kamerabilder oder mehrerer LiDAR-Abstandsmessungen – in die Berechnung des Belegungswertes einzubeziehen und gegebenenfalls variabel zu gewichten.

Um bei der Aktualisierung der Belegungswahrscheinlichkeit einer Zelle numerische Probleme im Fall von Wahrscheinlichkeitswerten nahe 0 oder 1 zu vermeiden und zugleich eine rechenzeiteffiziente Formulierung zu nutzen, werden für gitterbasierte Belegungskarten üblicherweise logarithmische Wahrscheinlichkeiten eingesetzt. Die Wahrscheinlichkeit $p(x)$ eines Zustandes x wird hierfür durch Logarithmieren in eine logarithmische Wahrscheinlichkeit $l(x)$ umgerechnet:

$$l(x) = \ln \frac{p(x)}{1 - p(x)}. \quad (2.10)$$

Es gilt nun $l(x) \in [-\infty, \infty]$.

In umgekehrter Richtung kann aus der logarithmischen Wahrscheinlichkeit $l(x)$ die Wahrscheinlichkeit $p(x)$ wieder wie folgt gewonnen werden:

$$p(x) = 1 - \frac{1}{1 + \exp(l(x))}. \quad (2.11)$$

Die Verarbeitung einer neuen Messung z_t zum Zeitpunkt t erfolgt bei Verwendung logarithmischer Wahrscheinlichkeiten als rechentechnisch schnell ausführbare Addition:

$$l_t(x) = l_{t-1}(x) + \log \frac{p(x | z_t)}{1 - p(x | z_t)} - \log \frac{p(x)}{1 - p(x)}. \quad (2.12)$$

Hierbei bezeichnet $p(x | z_t)$ die Wahrscheinlichkeit des Zustandes x – also etwa des „belegt“-Zustandes einer Gitterzelle – auf Basis der aktuellen Messung z_t . Das $p(x | z_t)$ zugrunde liegende Messmodell kann anwendungsspezifisch gewählt werden. Der Ausdruck $\log \frac{p(x)}{1 - p(x)}$ wird oft als l_0 abgekürzt und bezeichnet die initiale Belegungswahrscheinlichkeit. [80]

3 Kartengestützte Lokalisierung

Als wesentlicher Baustein des vorliegenden Konzeptes zur hochgenauen Fahrzeuglokalisierung mittels infrastrukturbasiert verwalteter digitaler Karten wird im folgenden Kapitel die fahrzeugseitige Komponente des Systems vorgestellt. Diese Komponente nutzt die von der Infrastrukturseite bereitgestellte digitale Karte, um beim Durchfahren des kartierten Bereiches eine hochgenaue Positionsbestimmung in Echtzeit unter Verwendung von Daten eines LiDAR-Sensors zu realisieren.

Im Folgenden werden zunächst die Anforderungen an das Verfahren zur kartenbasierten Lokalisierung im vorliegenden Anwendungsfall spezifiziert. Anschließend wird ein Überblick über den Stand der Forschung zu entsprechenden Echtzeitlokalisierungsverfahren gegeben, woraus sich die Systemspezifikation des hier eingesetzten Lokalisierungsverfahrens ableiten lässt. Die Umsetzung erklärt schließlich im Detail die Funktionsweise, die Implementierung auf dem Testträger sowie die Parametrierung des Systems.

3.1 Anforderungen

Wesentliche Anforderungen an das Fahrzeugsystem zur kartengestützten Lokalisierung lassen sich zunächst aus der verfügbaren Hardware zur Datenverarbeitung und der vorgesehenen Sensorik ableiten. Als Zielsystem wird in der vorliegenden Arbeit ein Linux-basierter eingebetteter Prototypencomputer verwendet, der mit einem Intel x86-Mehrkernprozessor ausgestattet ist. Der Lokalisierungsalgorithmus muss auf diesem System zuverlässig in Echtzeit lauffähig sein. Zudem soll die Software den aktuellen Entwicklungen im Fahrzeugbereich entsprechend zur parallelisierten Nutzung der vier zur Verfügung stehenden Rechenkerne geeignet sein [42][30]. Im Hinblick auf die Sensorik wird für die Umfelderkennung auf einen 360° LiDAR-Sensor zurückgegriffen, der geometrische Information aus der Fahrzeugumgebung in Form von Punktwolken zur Verfügung stellt. Prinzipiell kämen für diese Aufgabe auch andere Sensortechnologien wie etwa RADAR oder Kameras infrage; aufgrund der weiten Verbreitung von LiDAR im Bereich des autonomen Fahrens und der besonders guten Nutzbarkeit von LiDAR-Daten zur präzisen Abbildung von komplexen geometrischen Strukturen beschränkt sich die vorliegende Arbeit auf Laser-basierte Umfeldsensorik.

Neben LiDAR steht für die näherungsweise globale Lokalisierung ein kostengünstiges Einfrequenz-GNSS-System zur Verfügung und es können Raddrehzahlen sowie Daten einer Inertialmesseinheit genutzt werden. Aufgrund der verpflichtenden Verfügbarkeit des Europäischen Notrufsystems Emergency Call (eCall) [27] sowie eines elektronischen Stabilitätsprogramms [26] in allen in Europa zugelassenen Fahrzeugen sind diese Sensorsysteme in aktuellen PKW grundsätzlich verfügbar.

Das vorgesehene Anwendungsszenario betrifft Fahrten im innerstädtischen Umfeld, insbesondere in Kreuzungsbereichen. Hieraus lassen sich für automatisierte Fahrzeuge Geschwindigkeiten bis maximal 50 km h^{-1} ableiten. Im Hinblick auf die Samplerate ergibt sich die Anforderung, die Positionsinformation mit mindestens 50 Hz bereitzustellen, was zu einer maximalen Wegstrecke von 28 cm zwischen den Positionsdaten führt. Es wird erwartet, dass die erreichbare Genauigkeit des Lokalisierungssystems in dieser Größenordnung liegt, sodass sich mit der geforderten Samplerate eine realistische örtliche Auflösung ergibt.

Bezüglich der Verwendung von GNSS birgt das innerstädtische Szenario aufgrund von potenziell enger Bebauung mit zahlreichen Störfaktoren auf den Satellitenempfang ein besonders großes Risiko von Signalausfällen oder großen Fehlern in der Positionslösung. Das Lokalisierungssystem muss gegen diese Fehler robust sein, also auch dann eine zuverlässige Position liefern, wenn keine oder nur gestörte GNSS-Signale verfügbar sind. Eine weitere Robustheitsanforderung betrifft Abweichungen zwischen der digitalen Karte und der realen Situation im Fahrzeugumfeld. Zwar sind durch die kontinuierliche Aktualisierung des Kartenmaterials fundamentale Unterschiede unwahrscheinlich, trotzdem sind im dynamischen innerstädtischen Umfeld kurzfristige Änderungen der geometrischen Konstellation zu erwarten, die nicht immer bereits in der Karte repräsentiert sein müssen. Die Funktionsfähigkeit der Lokalisierung muss also für nicht perfektes Kartenmaterial sichergestellt werden.

Beim Einfahren in den kartierten Bereich stehen dem Fahrzeug üblicherweise noch keine hochgenauen Positionsdaten zur Verfügung, sodass eine Initialisierung des kartenbasierten Lokalisierungsverfahrens auch mit einer ungenauen Ausgangsposition gewährleistet sein muss. Im Anschluss wird eine schnelle Konvergenz zur korrekten Positionslösung gefordert. Das Kartenformat für die kartengestützte Lokalisierung ergibt sich aus den in Kapitel 4.3 definierten Randbedingungen. Als Ausgangsdaten werden georeferenzierte Positionen sowie die Fahrzeugorientierung im WGS84-Koordinatensystem gefordert.

Für einige der vorgenannten Anforderungen können auf Basis des Anwendungsszenarios keine sinnvollen quantitativen Grenzwerte vorgegeben werden. In Kapitel 5 werden die mit dem vorgeschlagenen Verfahren erreichbaren Kennwerte quantitativ untersucht und diskutiert. Eine Zusammenfassung der für die kartengestützte Lokalisierung definierten Anforderungen zeigt Tabelle 3.1.

Tabelle 3.1: Anforderungen Kartengestützte Lokalisierung

Merkmal	Anforderung
Zielhardware	Protoypensteuergerät mit Mehrkernarchitektur
Sensorik	360° LiDAR, Einfrequenz-GNSS, Raddrehzahlsignale, Inertialmesseinheit
Samplerate	50 Hz
GNSS-Verfügbarkeit	Robustheit gegen gestörten GNSS-Empfang und vollständigen Signalausfall
Datenformat digitale Karte	georeferenzierte Linienkarte gemäß Kapitel 4.3
Unsicherheit digitale Karte	Robustheit gegen Abweichungen von digitaler Karte und realem Umfeld
Datenformat Positionslösung	Ausgabe von Position und Orientierung im WGS84-Koordinatensystem

3.2 Stand der Forschung

Die Lokalisierung von Fahrzeugen mittels digitaler Karten ist seit vielen Jahren Gegenstand der Forschung. Zahlreiche Publikationen beschreiben vielfältige Verfahren zur Lösung dieses Problems, wobei ein breites Spektrum an Karten, Sensoren und Algorithmen zum Einsatz kommt. Im Folgenden werden daher die relevantesten Veröffentlichungen analysiert und es wird eine Klassifikation nach den wesentlichen Unterscheidungsmerkmalen vorgenommen.

3.2.1 Kartierte Merkmale

Eine sehr wesentliche Unterscheidung der Lokalisierungsverfahren kann nach den für die Lokalisierung genutzten Merkmalen erfolgen. Bei der Konzeption des Lokalisierungsverfahrens muss festgelegt werden, welche Umgebungsmerkmale in der digitalen Karte gespeichert und damit auch beim Durchfahren des kartierten Bereiches von den Fahrzeugsensoren detektiert werden sollen. Prinzipiell sind für die Konzeption der digitalen Karte zwei grundsätzliche Ansätze möglich: Dichte / kontinuierliche Karten enthalten alle Informationen, die bei einer ununterbrochenen Abtastung der Umgebung mit einem definierten Messverfahren gewonnen wurden. Landmarkenbasierte Karten können dagegen als eine Liste spezifischer Landmarken mit ihrer jeweiligen Geoposition betrachtet werden – die Umgebungsinformation ist hier also in diskreten Punkten konzentriert. Tabelle 3.2 zeigt eine Übersicht der analysierten Quellen mit einer Einordnung entsprechend der kartierten Merkmale.

Tabelle 3.2: Literaturübersicht Kartierte Merkmale

	Kartentypen	Publikationen
Landmarken- basiert	Fahrbahnmarkierungen	[54], [71], [46], [62], [51]
	Straßenbegrenzungen	[71], [62]
	Bordsteine	[54], [64]
	Vertikale, zylindrische Objekte	[71], [46], [78]
	Gebäudefassaden	[46]
	RADAR-Landmarken	[51]
	LiDAR-Landmarken	[72]
Dicht / kontinuierlich	Gitterbasierte 2D-Belegungskarte	[7], [66]
	Gaußprozess-Belegungskarte	[37], [59]
	Reflektivitätskarte der Straßenoberfläche	[89], [49], [50]
	Vertikale 2D-Rasterkarte	[81]
	Gitter der Fahrbahntopologie	[54]

Eine weit verbreitete Methode zur kontinuierlichen Kartierung besteht in der Nutzung von zweidimensionalen gitterbasierten Belegungskarten („Occupancy Grid Maps“, siehe Kapitel 2.2.2). Hierbei wird davon ausgegangen, dass jeder Ort einen von zwei Zuständen aufweisen kann – belegt oder nicht belegt. Durch die Diskretisierung der Umgebung mit einem üblicherweise äquidistanten Gitter entsteht ein Netz aus Gitterzellen, die diese Zustände widerspiegeln. Aufgrund von Messunsicherheiten treten bei der Kartierung im Allgemeinen widersprüchliche Informationen bezüglich des Belegungszustandes der Zellen auf, sodass zumeist kontinuierliche Belegungswahrscheinlichkeiten im Intervall $[0, 1]$ betrachtet werden, die anhand der zur Verfügung stehenden Informationen probabilistisch angepasst werden. Beispiele für entsprechende Lokalisierungssysteme auf Basis von LiDAR-Daten liefern [7] und [66].

Neben geometrischen Belegungsinformationen können mittels gitterbasierten Karten auch Helligkeitsunterschiede der Bodenebene, speziell der Fahrbahnoberfläche abgebildet werden. Entsprechende Arbeiten in [49], [50] und [89] verwenden hierfür Messungen zur Infrarotreflektivität der Fahrbahnoberfläche, sodass insbesondere Fahrbahnmarkierungen in der Karte erfasst werden. Als Erweiterung zu diskreten Gittern kommen in diesem Kontext auch Gaußsche Prozessmodelle zum Einsatz, die es ermöglichen, die räumlichen Belegungswahrscheinlichkeiten durch kontinuierliche Funktionen zu beschreiben [59], [37].

Neben den o.g. weit verbreiteten Methoden zur dichten bzw. kontinuierlichen Kartierung werden in der Literatur auch einige spezielle Verfahren beschrieben, die ebenfalls dieser Klasse zugeordnet werden können. In [81] wird eine zweidimensionale Rasterkarte vorgestellt, die vertikal an beiden Straßenrändern angeordnet ist und deren Zellen die

Abstände zu den nächsten Hindernissen in horizontaler Richtung als Tiefenwerte codieren. Ein gleichmäßiges Gitter von geometrischen Punkten auf der Fahrbahnoberfläche, die somit die Topologie der Straße abbilden, ist Teil der Lokalisierungslösung in [54].

Für die Klasse der landmarkenbasierten Verfahren finden sich in der Literatur vielfältige Implementierungen, wobei der Auswahl der genutzten Landmarken eine wichtige Bedeutung zukommt. Aufgrund ihrer großen Verfügbarkeit, der mit verschiedenen Sensoren recht zuverlässigen Erkennbarkeit sowie der weitgehenden Unveränderlichkeit spielen Fahrbahnmarkierungen im Bereich der landmarkenbasierten Verfahren eine große Rolle. Da solche Markierungen zudem zum Großteil aus einfachen linienförmigen Primitiven aufgebaut sind, können sie speicherplatzeffizient in digitalen Karten abgelegt werden. Einige Beispiele für diese Technologie finden sich unter anderem in [51], [62] und [46].

Alternativ oder zusätzlich zu Fahrbahnmarkierungen werden auch geometrische Informationen zu Bordsteinen ([54], [64]) oder Straßenbegrenzungen ([71], [62]) genutzt. Vertikale, pfahlförmige Strukturen wie Bäume, Ampel- oder Schildermasten werden aus verschiedenen Blickrichtungen identisch wahrgenommen und können anhand ihrer Mittelachse eindeutig und speicherplatzeffizient lokalisiert werden. Sie werden daher in [71], [78] und [46] kartiert und für die Lokalisierung genutzt. [46] integriert zudem geometrische Informationen von Gebäudefassaden.

Die vorgenannten Landmarkentypen haben die Eigenschaft, mit gewissen Einschränkungen sensoragnostisch nutzbar zu sein. Karten, die diese Informationen enthalten, können also im Allgemeinen unabhängig von der konkreten Sensorausstattung des Fahrzeuges genutzt werden. So lassen sich Fahrbahnmarkierungen sowohl mit LiDAR, als auch mit Kameradaten detektieren und für zylindrische Masten kommt auch ein RADAR-System infrage. Daneben wurden in der Literatur aber auch solche Lokalisierungsverfahren untersucht, die sensorspezifische Schlüsselpunkte nutzen. So erzeugen bestimmte Materialien und Geometrien in einer zu kartierenden Umgebung charakteristische Reflektionen von RADAR-Wellen, die eine gute Wiedererkennbarkeit versprechen [51]. Automatisch erkannte Schlüsselpunkte in LiDAR-Scans werden in [72] zur Erzeugung einer speicherplatzeffizienten Karte genutzt.

3.2.2 Genutzte Sensorik

Für die Kategorisierung der kartengestützten Lokalisierungsverfahren kann auch eine Unterscheidung nach den genutzten Sensorsystemen vorgenommen werden. Einerseits sind hierunter die Sensorsysteme zu verstehen, welche die in der Karte verzeichneten Umgebungsmerkmale aus Sicht des zu lokalisierenden Fahrzeuges detektieren. Andererseits verwenden die meisten Lokalisierungsverfahren auch zusätzliche Informationen über den Fahrzeugzustand aus weiteren Sensorsystemen, um im Sinne einer Sensorfusion die

Tabelle 3.3: Literaturübersicht Genutzte Sensorik

Genutzte Sensoren		Publikationen
Umgebungserkennung	LiDAR	[37], [66], [72], [59], [64], [71], [49], [46], [73], [50], [7]
	Kamera	[54], [71], [61], [89], [46], [51]
	Stereokamera	[62], [78], [73]
	RADAR	[51]
Fahrzeugzustand	GNSS	[37], [54], [71], [49], [46], [62], [78], [51], [50]
	Odometrie	[37], [66], [54], [64], [71], [89], [46], [62], [78], [73], [51]
	Inertiale Messeinheit	[59], [54], [64], [71], [89], [49], [51], [50]

Genauigkeit der Lokalisierung zu erhöhen. Welche Sensorsysteme in den verschiedenen Literaturquellen eingesetzt wurden, ist in Tabelle 3.3 zusammengefasst.

Bei den Sensorsystemen zur Umgebungserfassung spielt in der hier analysierten Literatur die LiDAR-Technologie eine besonders große Rolle. Wie Tabelle 3.3 zeigt, werden bei einer großen Zahl an Publikationen Laserscanner eingesetzt, darunter die seit 2017 veröffentlichten Arbeiten von [37], [72], [46] und [73]. Mit einer sehr guten geometrischen Genauigkeit, einer vergleichsweise geringen Empfindlichkeit gegenüber sich ändernden Umgebungsbedingungen und einer großen Sichtweite bei einem möglichen horizontalen Sichtfeld von 360° vereint diese Technologie zahlreiche für die Lokalisierung vorteilhafte Eigenschaften. Nachteilig sind vor allem die hohen Kosten von LiDAR-Sensoren, die eine weite Verbreitung in aktuell üblichen Serienfahrzeugen bislang verhindert haben [88].

Im Gegensatz zu LiDAR-Sensorik sind Kamerasysteme in aktuellen Fahrzeugen häufig bereits serienmäßig verbaut, da sie für Komfort- und Sicherheitsfunktionen wie die Verkehrszeichenerkennung oder Fußgängerschutzsysteme zum Einsatz kommen [88]. Ein prinzipieller Vorteil im Vergleich zu anderen Sensorsystemen besteht in der großen Auflösung des Sensors und der damit sehr feinen Abtastung der Umgebung. Vergleichsweise niedrigen Kosten für Kameras stehen höhere Aufwände für Recheneinheiten zur Verarbeitung der großen Datenmengen gegenüber. Durch die Kombination mehrerer Optiken zu sogenannten Stereokameras können auch dreidimensionale Umgebungsinformationen gewonnen werden. Beispiele für Lokalisierungssysteme auf Basis von Kameradaten finden sich unter anderem in [89], [71] und [51]. Stereokameras werden in [62], [78] und [73] eingesetzt.

Mit der Ausnahme von [51] kommt RADAR-Sensorik in der untersuchten Literatur für die kartengestützte Lokalisierung nicht zum Einsatz. Nachteile der Technologie liegen in der vergleichsweise geringen Auflösung. Die hohe Reichweite, die gegenüber LiDAR

Tabelle 3.4: Literaturübersicht Lokalisierungsalgorithmen

Lokalisierungsalgorithmen	Publikationen
Monte-Carlo-Lokalisierung / Partikelfilter	[37], [66], [64], [49], [78], [54], [71], [73], [51]
Extended / Unscented Kalman-Filter	[89], [78], [62]
Histogrammfilter	[50]
Direktes Scan Matching	[71], [7]
Optimierungsbasierte Methoden	[46]

kostengünstigere Produktion und die weitgehende Unempfindlichkeit gegenüber Schmutz und schwierigen optischen Bedingungen in Verbindung mit aktuellen Weiterentwicklungen der Technologie lassen jedoch vermuten, dass auch RADAR für künftige Anwendungen in der Umgebungserfassung eine relevante Rolle spielen wird [88].

Zur Unterstützung der kartengestützten Lokalisierung setzen zahlreiche Publikationen satellitenbasierte Positionsinformationen aus GNSS ein. Vielfach wird jedoch auf die mangelnde Zuverlässigkeit der GNSS-Positionen insbesondere in städtischen Szenarien verwiesen. [71] und [89] nutzen GNSS daher lediglich für die Initialisierung ihres Lokalisierungsverfahrens mit einer georeferenzierten Startposition. [49] setzt eine nachgelagerte Plausibilisierungsstufe ein, die eine Divergenz des eigentlichen Lokalisierungsverfahrens anhand von GNSS-Positionen verhindert.

Ein in der Mehrzahl der untersuchten Publikationen (siehe Tabelle 3.3) eingesetztes Sensorsystem ist die Raddrehzahlsensorik bzw. Odometrie. Die hochfrequent zur Verfügung stehenden Raddrehzahlsignale liefern zuverlässige Informationen über die zeitliche Bewegung des Fahrzeuges und kommen damit häufig in Prädiktionsritten der verwendeten Filterungsalgorithmen (siehe Kapitel 3.2.3) zum Einsatz. Auf ähnliche Weise werden vielfach auch Daten aus der fahrzeugeigenen Inertialsensorik / IMU integriert. Hier werden sowohl translatorische Beschleunigungen, als auch Drehrateninformationen verarbeitet.

3.2.3 Lokalisierungsalgorithmen

Eine sehr wesentliche Kategorie zur Unterscheidung der verschiedenen kartenbasierten Lokalisierungsverfahren besteht in den verwendeten Algorithmen zur Positionsbestimmung. Tabelle 3.4 stellt die Zuordnung der analysierten Literaturquellen zu den verschiedenen Algorithmen dar.

Umfangreiche Verwendung finden in der Literatur Lokalisierungsalgorithmen auf Basis von Partikelfiltern, die häufig als Monte-Carlo-Lokalisierung (MCL) bezeichnet werden

[29]. Für die grundlegenden Prinzipien des Verfahrens sei auf Kapitel 2.1.2.2 verwiesen. Die entsprechenden Lokalisierungssysteme werden sowohl mit Feature-basierten Kartentypen (z.B. [78], [71]), als auch mit dichten Karten (z.B. [37], [49]) kombiniert. Im Falle der Feature-basierten Karten muss neben dem Problem der Positionsschätzung auch das Datenzuordnungsproblem behandelt werden, welches die Assoziation zwischen den vom Fahrzeug wahrgenommenen Features und den in der Karte vorhandenen Features leistet. Hierfür wird das Partikelfilter üblicherweise um ein explizites Zuordnungsverfahren ergänzt, z.B. mittels einer „nächster Nachbar“-Methode [73] oder mittels eines Poisson-Prozesses [51]. Dem Partikelfilter ähnlich ist das Histogramm-Filter, bei dem zur Abbildung der Wahrscheinlichkeitsverteilung anstelle einer Vielzahl an Partikeln ein Histogramm verwendet wird. Es kommt in [50] zum Einsatz.

Eine weitere wichtige Klasse der Lokisierungsalgorithmen sind die verschiedenen Varianten des Kalman-Filters. Wie bereits in Kapitel 2.1.2.1 eingeführt, nutzt der klassische Algorithmus lineare Modelle für die Prädiktion des Systemverhaltens sowie die Zusammenhänge zwischen Messsignalen und Systemzuständen. Da diese Zusammenhänge im Falle der Fahrzeuglokalisierung häufig wesentlich genauer durch nichtlineare Modelle beschrieben werden können, finden sich in der Literatur hauptsächlich Anwendungen von EKF ([89], [78]) oder Unscented Kalman-Filter (UKF) [62].

Der direkte Abgleich von Sensordaten mit der digitalen Karte kann ebenfalls zur Positionsbestimmung eingesetzt werden. Dieses Vorgehen ist vor allem bei dichten Karten in Kombination mit LiDAR-Daten sinnvoll. Iterative Verfahren wie ICP (siehe 2.1.4.1) korrigieren die Fahrzeugposition ausgehend von einer initialen Schätzung so, dass sich eine möglichst gute Überlappung von Karte und Sensorausgabe ergibt. Beispiele für das direkte Scan Matching finden sich in [71] und [7].

Die auf Basis aller aktuellen Sensorinformationen wahrscheinlichste Position kann auch als Ergebnis eines Optimierungsproblems ähnlich dem GraphSLAM-Verfahren formuliert werden. Ein solcher optimierungsbasierter Ansatz unter Verwendung eines nichtlinearen Least-Squares-Verfahrens ist in [46] beschrieben.

3.3 Konzeption

Basierend auf den in Kapitel 3.1 dargestellten Anforderungen an das Lokalisierungssystem und dem in Kapitel 3.2 erarbeiteten Überblick über den Stand der Forschung wird im Folgenden die Konzeption des Lokalisierungssystems im vorliegenden Anwendungsfall vorgenommen.

Im Hinblick auf die eingesetzte Sensorik wird aufgrund der guten geometrischen Genauigkeit sowie der ausreichenden Reichweite und Messfrequenz ein 360°-Laserscanner als einziges System zur Umfelderkennung verwendet. Daneben kommen nur solche Sensoren

zum Einsatz, die in aktuell üblichen Serienfahrzeugen bereits verfügbar sind – konkret werden das Gierratensignal einer IMU sowie Raddrehzahlinformationen der beiden Hinterräder des Fahrzeuges genutzt.

Der zugrunde liegende Algorithmus basiert auf einem Partikelfilter, dessen Zustände durch die Fahrzeugpose im zweidimensionalen Raum bestimmt sind. Die Zustände sind also die Position (x, y) sowie der Gierwinkel φ , wobei als Koordinatensystem ein lokales, kartesisches ENU-System gewählt wird. Durch die Festlegung auf den zweidimensionalen Raum kann die Komplexität und somit der Rechenaufwand des Algorithmus wesentlich reduziert werden, da anstelle von mindestens sechs Zustandsvariablen im dreidimensionalen Raum hier nur drei Zustände ausreichen, um die Pose des Fahrzeuges zu beschreiben. Zugleich gehen kaum relevante Informationen verloren, da Nick- und Wankwinkel des Fahrzeuges im Kontext der automatisierten Navigation in der Regel nicht von Relevanz sind. Bezüglich der Höheninformation können durch die zweidimensionale Betrachtung etwa bei Über- oder Unterführungen prinzipiell Doppeldeutigkeiten auftreten. Diese lassen sich jedoch durch ein Matching auf entsprechend definierte Kartenebenen auflösen, ohne eine zusätzliche, kontinuierliche Höhenkoordinate einführen zu müssen.

Die Wahl des Partikelfilters als Basisalgorithmus für die Lokalisierung beruht auf einigen spezifischen Vorteilen dieses Prinzips gegenüber anderen Ansätzen. Zunächst kann das Partikelfilter der Klasse der Bayes'schen Zustandsschätzer zugeordnet werden (siehe Kapitel 2.1.2.2), es berücksichtigt also auftretende Unsicherheiten bei Sensordaten und beim Mess- und Bewegungsmodell explizit. Auch die ermittelte Pose liegt in Form einer Wahrscheinlichkeitsverteilung vor; neben der eigentlichen Posenschätzung kann also auch eine Information über die zugehörige Unsicherheit entnommen werden. Im Vergleich zu Methoden, die auf dem direkten Abgleich der Daten von Umgebungssensorik und Karte beruhen, bietet die Bayes'sche Wahrscheinlichkeitstheorie somit großes Potenzial für eine robuste und zuverlässige Lokalisierung trotz komplexer, unmodellierter Systemdynamiken, möglicherweise ungenauer digitaler Karten und Sensorunsicherheiten. In Abgrenzung zum Kalman-Filter sowie den Erweiterungen EKF und UKF müssen beim Partikelfilter die auftretenden Wahrscheinlichkeitsverteilungen nicht als Gaußsche Normalverteilungen approximiert werden. Relevant ist dies im vorliegenden Anwendungsfall insbesondere für das Messupdate, bei dem der Abgleich zwischen den Daten des Laserscanners und der Karte erfolgt. Wenn mehrere Regionen von Fahrzeugposen existieren, die zu einer guten Übereinstimmung zwischen Sensordaten und digitaler Karte führen, können diese unabhängigen Hypothesen im Partikelfilter durch multimodale Wahrscheinlichkeitsverteilungen zutreffend abgebildet und zeitlich nachverfolgt werden. Eine Gaußsche Approximation würde hier zu einer starken Verfälschung führen, bei der mit den zwei Parametern Mittelwert und Standardabweichung keine der Hypothesen adäquat abgebildet werden könnte.

Das für die Lokalisierung gewählte Kartenformat approximiert die geometrischen Merkmale der Umgebung durch diskrete Kartensegmente. Jedes Segment bezeichnet eine gerade

Linie in zwei Dimensionen, die jeweils durch ihre Anfangs- und Endpunkte definiert ist. Bezüglich der genaueren Definition der digitalen Karte wird an dieser Stelle auf die Konzeption des Kartendienstes in Kapitel 4.3 verwiesen.

3.4 Umsetzung

Die sich aus der Konzeption des Lokalisierungsalgorithmus (siehe Kapitel 3.3) ergebende Systemstruktur ist in Abbildung 3.1 visualisiert. Entsprechend den wesentlichen Komponenten des auf dem Fahrzeug zu implementierenden Gesamtsystems sind auch die folgenden Unterkapitel strukturiert. Nach der Darstellung der verwendeten Modelle und Algorithmen für die Prädiktion und das Messupdate in Kapitel 3.4.1 und 3.4.2 wird in Kapitel 3.4.3 auf die zeitliche Ablaufstruktur der erforderlichen Berechnungen eingegangen. Kapitel 3.4.4 und 3.4.5 behandeln schließlich die Initialisierung des Partikelfilters sowie die Ermittlung der für die vorliegende Anwendung optimalen Parametrierung. Details zur Umsetzung des Lokalisierungsalgorithmus sowie das Konzept der Kartierung und Lokalisierung sind zudem Inhalte der eigenen Veröffentlichungen [3] und [2].

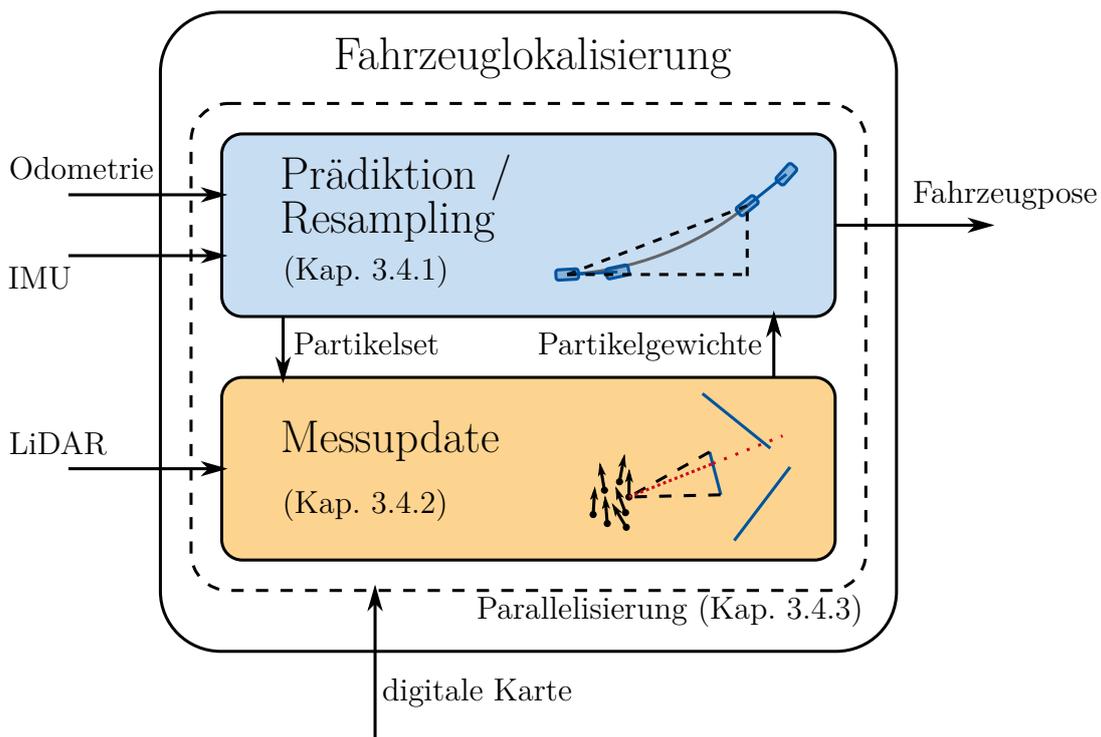


Abb. 3.1: Überblick der Komponenten des Lokalisierungsalgorithmus

3.4.1 Prädiktionsmodell

Innerhalb des Partikelfilters wird das Prädiktionsmodell genutzt, um für jedes Partikel des Partikelsets Υ ausgehend von der Fahrzeugpose des vorherigen Zeitschrittes $(x, y, \varphi)|_{k-1}$ den Zustand des aktuellen Zeitschrittes $(x, y, \varphi)|_k$ unter Verwendung der Sensorinformationen aus Hinterraddometrie und inertialer Messeinheit zu präzisieren. In jedem Prädiktionsschritt wird dafür zunächst die zurückgelegte Distanz d_{odo} aus den Winkelgeschwindigkeiten des linken und rechten Hinterrades ω_{hl} bzw. ω_{hr} bestimmt:

$$d_{odo} = 0.5 \cdot (r_{hl} \cdot \omega_{hl} + r_{hr} \cdot \omega_{hr}) \cdot T. \quad (3.1)$$

Das Modell nutzt hierbei somit die Annahme einer konstanten Geschwindigkeit über den Prädiktionszeitraum. Die Bildung des arithmetischen Mittels der beiden Drehzahlsignale entspricht einer Reduktion der an einem zweisepurigen Fahrzeug gemessenen Bewegungsinformationen auf das entsprechende Einspurmodell. r_{hl} und r_{hr} bezeichnen die Abrollradien des linken und rechten Hinterrades, T steht für die Samplezeit des zeitdiskreten Algorithmus.

Die inertielle Messeinheit misst die Drehrate des Fahrzeuges $\dot{\varphi}$ um die Hochachse, sodass die Gierwinkeldifferenz $\Delta\varphi$ direkt bestimmt werden kann:

$$\Delta\varphi = T \cdot \dot{\varphi}. \quad (3.2)$$

In Abbildung 3.2 sind die geometrischen Zusammenhänge des kinematischen Fahrzeugmodells dargestellt. Aufgrund des durch die Kinematik des Fahrzeuges nach unten beschränkten Kurvenradius sowie der hohen Samplerate kann der sich aus der Kurvenfahrt ergebende Kreisbogen durch die entsprechende Kreissehne angenähert werden. Der hierdurch zu erwartende Fehler liegt deutlich unter der Unsicherheit, die durch den variablen Abrollradius sowie Schlupf an den Rädern verursacht wird. Die neuen Zustände $(x, y, \varphi)|_k$ werden anhand des Modells wie folgt berechnet:

$$x_k = x_{k-1} + (d_{odo} + e_d) \cdot \cos(\varphi_{k-1} + 0.5 \cdot (\Delta\varphi + e_{\dot{\varphi}})) \quad (3.3)$$

$$y_k = y_{k-1} + (d_{odo} + e_d) \cdot \sin(\varphi_{k-1} + 0.5 \cdot (\Delta\varphi + e_{\dot{\varphi}})) \quad (3.4)$$

$$\varphi_k = \varphi_{k-1} + (\Delta\varphi + e_{\dot{\varphi}}). \quad (3.5)$$

Das Modell enthält die zwei Fehlerterme $e_d \sim \mathcal{N}(0, \sigma_{Odo}^2)$ für die Unsicherheit der durch die Radodometrie gemessenen Distanz und $e_{\dot{\varphi}} \sim \mathcal{N}(\mu, \sigma_{IMU}^2)$ für den durch die IMU ermittelten Gierwinkel. e_d und $e_{\dot{\varphi}}$ werden in jedem Zeitschritt zufällig aus einer Gaußschen Normalverteilung mit Mittelwert $\mu = 0$ gezogen.

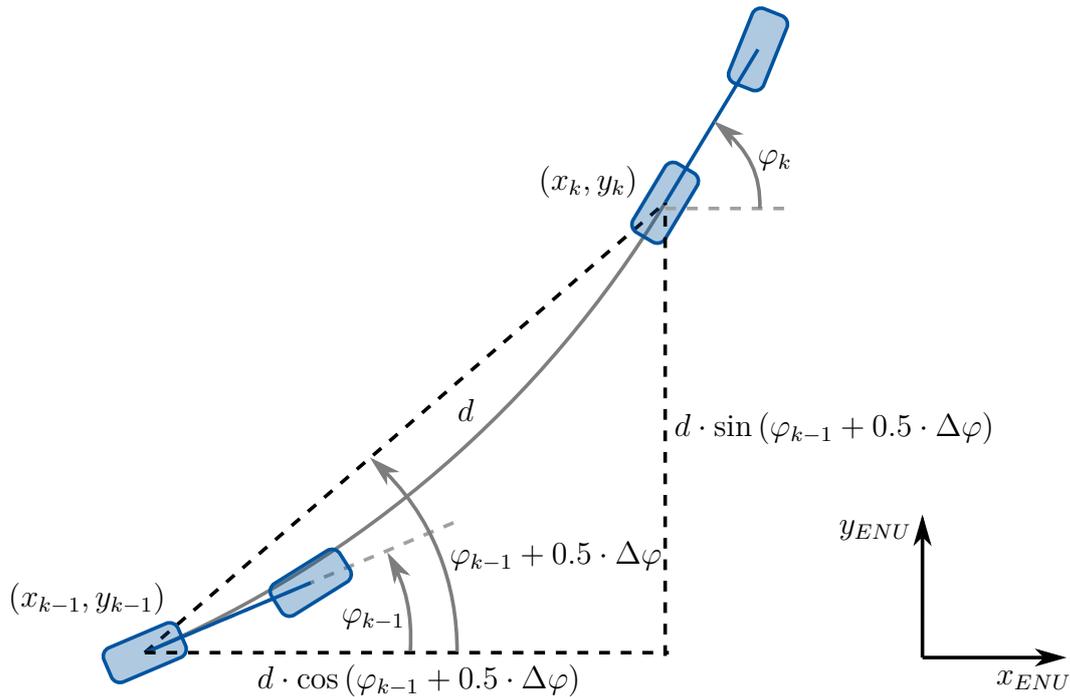


Abb. 3.2: Kinematisches Fahrzeugmodell zur Bewegungsprädiktion

Als Ergebnis des Prädiktionsschrittes steht somit für jedes Partikel des aktuellen Partikelsets Υ eine neue Posenschätzung zur Verfügung. Die Partikelgewichte werden hierbei nicht verändert – sie werden mit dem Wert $w = 0$ initialisiert und erst im Rahmen des nächsten Messupdates neu bestimmt. Das aktualisierte Partikelset bildet auch die Grundlage für die Bestimmung der neuen Posenschätzung des Fahrzeuges am Ende jedes Prädiktionsschrittes. Hierfür wird das arithmetische Mittel der Posenhypothesen aller Partikel gebildet.

3.4.2 Messupdate

Das Ziel des Messupdates innerhalb des Partikelfilters liegt darin, für ein gegebenes, ungewichtetes Partikelset Υ die individuellen Partikelgewichte zu berechnen. Diese ergeben sich aus dem Abgleich der in jedem Partikel gespeicherten Posenhypothese mit der aktuellen LiDAR-Messung und der digitalen Karte. Die dem Messupdate zugrunde liegenden Rechenschritte sind in Algorithmus 3 dargestellt. Aus Gründen der Übersichtlichkeit wird im Folgenden auf die Darstellung der Indizes k für den aktuellen Zeitschritt verzichtet.

Grundlegend kann Algorithmus 3 in zwei wesentliche Abschnitte aufgeteilt werden: In den Zeilen 2 bis 6 erfolgt die Berechnung von Hilfsgrößen zu den einzelnen Segmenten der

Algorithmus 3 Messupdate (Partikelset Υ , LiDAR-Scan Z , Karte Ω)

```

1: for all Partikel  $n$  des Partikelsets  $\Upsilon$  do
2:   for all Segmente  $m$  der Karte  $\Omega$  do
3:     Berechne Startwinkel  $\alpha_{min}^m$  des Kartensegmentes, ausgehend von Partikel  $n$ 
4:     Berechne Endwinkel  $\alpha_{max}^m$  des Kartensegmentes, ausgehend von Partikel  $n$ 
5:     Abschätzung für min. Abstand  $d_{min}^m$  des Kartensegmentes zu Partikel  $n$ 
6:   end for
7:   for all LiDAR-Punkte  $p$  des LiDAR-Scans  $Z$  do
8:     Initialisiere Variable für kürzestes Laserecho  $d_{L,best} = 0$ 
9:     for all Segmente  $m$  der Karte  $\Omega$  do
10:      if  $d_{min}^m < d_{L,max}$  and  $\alpha_{min}^m < \alpha_p < \alpha_{max}^m$  then
11:        Berechne Distanz  $d_L^{p,m}$  zu Schnittp. Laserstrahl  $p$  mit Kartensegment  $m$ 
12:        if  $d_L^{p,m} < d_{L,max}$  and ( $d_L^{p,m} < d_{L,best}$  or  $d_{L,best} == 0$ ) then
13:          Setze  $d_{L,best} = d_L^{p,m}$ 
14:        end if
15:      end if
16:    end for
17:    Berechne Punktgewicht  $w^p$  aus LiDAR-Punkt  $p$  und  $d_{L,best}$ 
18:  end for
19:  Berechne Partikelgewicht  $w^n$  als Summe aller Punktgewichte  $w^p$ 
20: end for
21: return gewichtetes Partikelset  $\Upsilon_w$ 

```

digitalen Karte Ω , die der Steigerung der Effizienz des Algorithmus dienen. Anschließend wird unter Verwendung dieser Hilfsgrößen in den Zeilen 7 bis 18 die eigentliche Berechnung der gesuchten Gewichte durch den Abgleich von Sensormessung und digitaler Karte durchgeführt. Die Berechnung des Partikelgewichtes wird einzeln für jedes Partikel des Partikelsets Υ durchgeführt; die äußerste Schleife des Algorithmus von Zeile 1 bis Zeile 19 läuft demnach über das gesamte Partikelset. Aufgrund der Unabhängigkeit der für die einzelnen Partikel erforderlichen Berechnungen kann diese Schleife aufgeteilt und in parallelen Threads auf einem Mehrkernprozessor verarbeitet werden.

Das Gewicht eines Partikels ist definiert als die bedingte Wahrscheinlichkeit, die Messung z zu erhalten, wenn der im Partikel gespeicherte Zustand vorliegt (siehe Kapitel 2.1.2.2). Diese Wahrscheinlichkeit kann aus dem Vergleich der realen LiDAR-Messung mit einer virtuellen Messung ermittelt werden. Die virtuelle Messung bestimmt dabei, welches Messergebnis zu erwarten wäre, wenn der Laserscanner sich in der im betrachteten Partikel gespeicherten Pose befinden würde. Zur Realisierung dieser virtuellen Messung wird also ein Modell des Messvorganges benötigt, das die Ausgabe des entsprechend positionierten Laserscanners auf Basis der digitalen Karte simuliert.

Da sich eine Messung des Laserscanners als Punktwolke aus einer großen Anzahl an

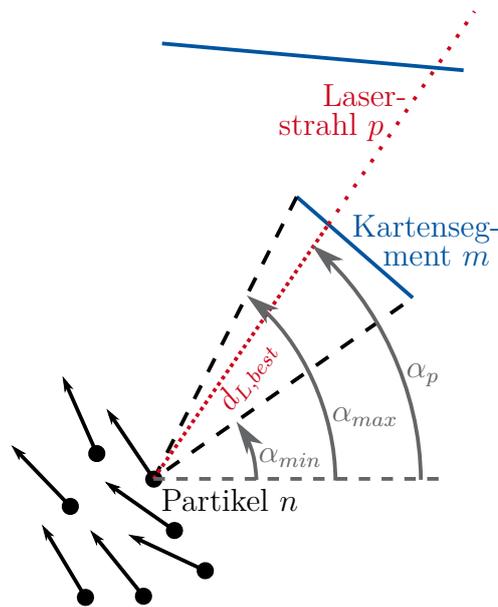


Abb. 3.3: Geometrische Größen beim Messupdate

einzelnen Entfernungsmessungen zusammensetzt, ist die Simulation dieses Messprinzips sehr rechenaufwendig. Um die virtuelle Punktwolke ausgehend von der Pose eines Partikels ermitteln zu können, muss für jeden vom Laserscanner ausgesendeten Laserstrahl ermittelt werden, auf welchem kartierten Liniensegment der Strahl auftrifft und welche Entfernung zwischen dem Sensor und dem Reflexionspunkt vorliegt. Die in den Zeilen 3, 4 und 5 des Algorithmus 3 für jedes Liniensegment berechneten Hilfsgrößen α_{min} , α_{max} und d_{min} erleichtern diesen Schritt, da mit ihnen schnell zu prüfende Vorabbedingungen formuliert werden können, welche die rechenaufwendige Schnittpunktberechnung aus Laserstrahl und Liniensegment in vielen Fällen unnötig machen. Die Definition von α_{min} und α_{max} ist in Abbildung 3.3 dargestellt. Für d_{min} wird als grobe Abschätzung zur sicheren Seite die folgende Formel genutzt, bei der d_1 und d_2 die Distanz von der Partikelpose zu den beiden Endpunkten des Liniensegmentes und d_{12} die Länge des Liniensegmentes bezeichnen:

$$d_{min} = \max(0, \max(d_1, d_2) - d_{12}). \quad (3.6)$$

In Zeile 10 des Algorithmus 3 können nun vor der Schnittpunktberechnung aus Laserstrahl und Liniensegment anhand der beiden Vorabbedingungen alle Segmente ausgeschlossen werden, bei denen der Winkel α_p des betrachteten Laserstrahls nicht innerhalb der Winkelgrenzen des Segmentes liegt bzw. das Segment sicher außerhalb der maximalen Messreichweite $d_{L, max}$ des Laserscanners liegt.

Für alle Kartensegmente, die nicht im Vorhinein ausgeschlossen werden können, erfolgt in

Zeile 11 die Berechnung der vom LiDAR-Sensor als Messung zu erwartenden Distanz d_L , wobei (x_1^m, y_1^m) und (x_2^m, y_2^m) die beiden Endpunkte des Liniensegmentes m bezeichnen und (x^n, y^n) als Symbole für die im Partikel n gespeicherte Position stehen:

$$d_L = \frac{(y_1^m - y^n) \cdot (x_2^m - x_1^m) + (x^n - x_1^m) \cdot (y_2^m - y_1^m)}{(x_2^m - x_1^m) \cdot \sin(\alpha_p) - (y_2^m - y_1^m) \cdot \cos(\alpha_p)}. \quad (3.7)$$

Wie Abbildung 3.3 verdeutlicht, können grundsätzlich mehrere Schnittpunkte des Laserstrahls mit Kartensegmenten auftreten. Mit $d_{L,best}$ wird aus diesen möglichen Kandidaten derjenige ausgewählt, der dem Laserscanner am nächsten liegt, also die kleinste Distanzmessung zur Folge hat. Dies ist konsistent mit der Behandlung von mehrfachen Echos des realen Laserscanners, die insbesondere bei halbtransparenten Objekten auftreten können: Auch hier wird stets der kleinste Abstandswert übernommen.

In Zeile 17 erfolgt die Berechnung des Gewichtes w_p für den betrachteten Laserstrahl p . Hierbei wird die ermittelte virtuelle Distanzmessung $d_{L,best}$ mit der realen Messung des Sensors z^p verglichen. Als Fehlermodell kommt eine Gaußsche Normalverteilung mit Standardabweichung σ_L zum Einsatz, sodass sich der folgende formelmäßige Zusammenhang ergibt:

$$w^p = \frac{1}{\sigma_L \cdot \sqrt{2\pi}} \cdot \exp\left(-\frac{(d_{L,best} - z^p)^2}{2\sigma_L^2}\right). \quad (3.8)$$

Als finaler Schritt des Messupdates wird in Zeile 19 von Algorithmus 3 das Gewicht w^n des Partikels n berechnet. Es ergibt sich als Summe der Einzelgewichte w^p jedes Laserstrahls. Mit dem gewichteten Partikelset stehen am Ende des Messupdates die erforderlichen Informationen zur Verfügung, um den Resampling-Schritt des Partikelfilters gemäß Kapitel 2.1.2.2 durchzuführen.

3.4.3 Parallelisierung

Wie in Kapitel 3.1 dargestellt, besteht ein Ziel bei der Implementierung des Algorithmus zur kartenbasierten Lokalisierung darin, eine Samplerate von 50 Hz für die Ausgabe der Poseninformationen zu erreichen. Aufgrund des hohen Rechenaufwandes bei der Berechnung des LiDAR-Messmodells für jedes Partikel ist diese Anforderung auch auf aktueller Rechenhardware nicht unkritisch. Zwar kann im Zuge der Parametrierung des Algorithmus etwa durch die Wahl einer kleinen Partikelzahl der Rechenaufwand stark reduziert werden. Solche Maßnahmen fordern aber in der Regel zugleich Kompromisse bei der Genauigkeit oder Robustheit der ermittelten Positionsinformation.

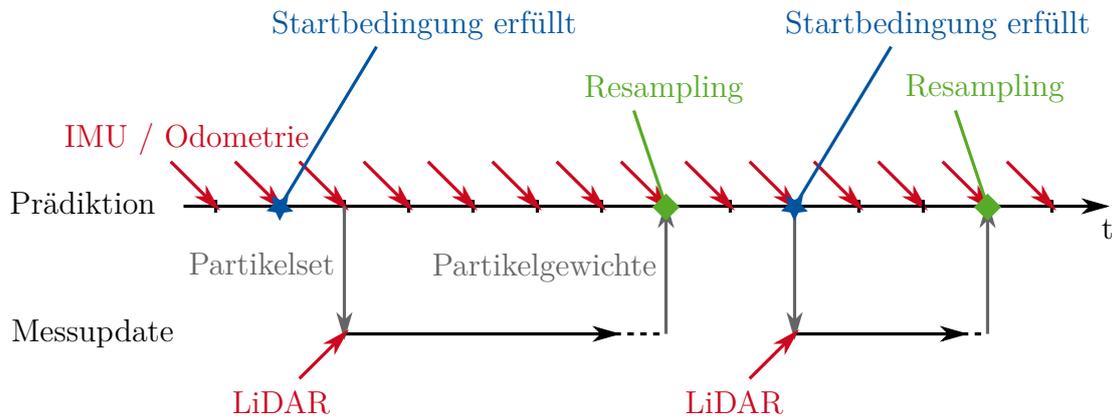


Abb. 3.4: Zeitlicher Ablauf des Lokalisierungsalgorithmus

Bezüglich der Effektivität des Messupdates kann gleichzeitig festgestellt werden, dass die berechneten Gewichte vor allem dann aussagekräftig sind, wenn das Fahrzeug seit dem letzten Update eine relevante Änderung der Pose vollzogen hat. Neue Informationen für das Aussortieren bzw. Vervielfältigen von Partikeln ergeben sich hauptsächlich daraus, dass neue Bereiche der digitalen Karte sichtbar werden oder die Perspektive des Sensors auf die kartierten Objekte variiert. Wird das Messupdate – etwa bei stehendem Fahrzeug – mit identischen Informationen des LiDAR-Sensors mehrfach hintereinander ausgeführt, so wird die entsprechende Beobachtung im Partikelset überrepräsentiert und Partikel, die abweichende Posenhypthesen beinhalten, gehen verloren.

Unter diesen Randbedingungen kann eine Auflösung des Zielkonfliktes zwischen Rechenzeit und Genauigkeit also dadurch erreicht werden, dass für das Messupdate keine feste Wiederholungsrate von 50 Hz gefordert wird. Das Messupdate wird stattdessen nur dann ausgeführt, wenn eine minimal erforderliche Fahrzeugbewegung seit dem letzten Update-Schritt als Startbedingung erfüllt ist. Durch die Parallelisierung der Prozesse von Prädiktion und Messupdate kann somit erreicht werden, dass die jeweils aktuellste prädizierte Pose mit einer festen Rate von 50 Hz ausgegeben werden kann und zugleich das Messupdate bedarfsgerecht und über mehrere Prädiktionsschritte hinweg berechnet werden kann. Die sich ergebende zeitliche Ablaufstruktur des Algorithmus ist in Abbildung 3.4 visualisiert.

In jedem Zeitschritt der Prädiktion wird neben der Berechnung der neuen Pose aller Partikel auch eine Prüfung der Startbedingung für ein neues Messupdate vorgenommen. Ist diese Bedingung erfüllt, so wird ein neuer paralleler Thread für das Messupdate gestartet und das aktuelle Partikelset übergeben. Aufgrund der niedrigeren Messfrequenz des verwendeten Laserscanners von 20 Hz kann es sein, dass zum aktuellen Zeitschritt keine neue LiDAR-Messung zur Verfügung steht. In diesem Fall wird das Messupdate bei der nächsten Verfügbarkeit eines Laserscans gestartet. Die folgende Berechnung des

Messupdates kann abhängig von der Komplexität der innerhalb der Sensorreichweite liegenden Kartensegmente eine variable Zeit in Anspruch nehmen. Sobald die Gewichte bestimmt wurden, erfolgt vor dem Beginn des nächsten Prädiktionsschrittes das Resampling. Zwar wurden die dem Messupdate zugrunde liegenden Partikel zwischenzeitlich im Allgemeinen durch mehrere Prädiktionsschritte verändert. Da die Zuordnung der Gewichte zu den Partikeln allerdings in unveränderter Weise bekannt ist, können durch das Resampling trotzdem die passenden Partikel aussortiert bzw. vervielfältigt werden.

3.4.4 Initialisierung

Die vom Partikelfilter bei der Prädiktion und beim Messupdate durchgeführten Operationen setzen die Existenz eines Partikelsets voraus, das die aktuelle Zustandsschätzung des Systems repräsentiert. Wird der Lokalisierungsalgorithmus also – etwa beim Einfahren von einem nicht kartierten Areal in einen kartierten innerstädtischen Kreuzungsbereich – neu gestartet, so muss auf Basis des zu diesem Zeitpunkt verfügbaren Wissens eine entsprechende Initialisierung des Partikelsets vorgenommen werden. In der Regel kann in diesem Fall davon ausgegangen werden, dass Positionsinformationen des fahrzeugeigenen GNSS-Systems verfügbar sind, welche für die Generierung dieses Partikelsets genutzt werden können.

Die x - und y -Positionen der Partikel werden demnach bei der GNSS-basierten Initialisierung durch zufälliges Ziehen aus einer Gaußschen Normalverteilung $\mathcal{N}(\mu, \sigma_G^2)$ gewonnen. Der Mittelwert μ der Normalverteilung entspricht der vom GNSS bestimmten Position; die Standardabweichung σ_G wird dem Fehlermodell des GNSS-Empfängers entsprechend parametrisiert. Da jedes Partikel neben den kartesischen Positionen auch einen Zustand für die Fahrzeugorientierung enthält, muss dieser ebenfalls initialisiert werden. Eine einzelne GNSS-Messung kann diese Information nicht liefern, daher wird für die Ermittlung der Bewegungsrichtung zusätzlich eine zurückliegende GNSS-Position herangezogen, wobei eine minimale gefahrene Distanz zwischen beiden Messungen gefordert und auf Basis der Hinterradodometrie überprüft wird.

Ist keine GNSS-basierte Position verfügbar, so kann die Initialisierung des Partikelfilters prinzipiell auch davon unabhängig durchgeführt werden. Voraussetzung ist lediglich, dass sich das Fahrzeug innerhalb des kartierten Bereiches befindet. Als initiale Zustandsschätzung wird in diesem Fall eine Zufallsverteilung gewählt, bei der jede Pose innerhalb der Kartengrenzen als gleich wahrscheinlich angenommen wird. Abbildung 3.5 (a) zeigt beispielhaft eine solche Verteilung am Beispiel eines Parkdecks in einem mehrgeschosigen Parkhaus. Wie in Abbildung 3.5 (b) und (c) zu sehen ist, erreicht das sukzessive Resampling des Lokalisierungsalgorithmus parallel zur Fortbewegung des Fahrzeuges auch bei einer solchen Initialisierung die Konvergenz des Partikelsets. Im Zustand (b), der 4 s nach der ersten Ausführung des Partikelfilters vorliegt, bildet das Partikelset eine multimodale Wahrscheinlichkeitsverteilung ab, bei der aufgrund von Symmetrien

der digitalen Karte verschiedene Posenhypothesen verfolgt werden. Nach 8 s liegen alle Partikel in einer Umgebung des realen Zustandes.

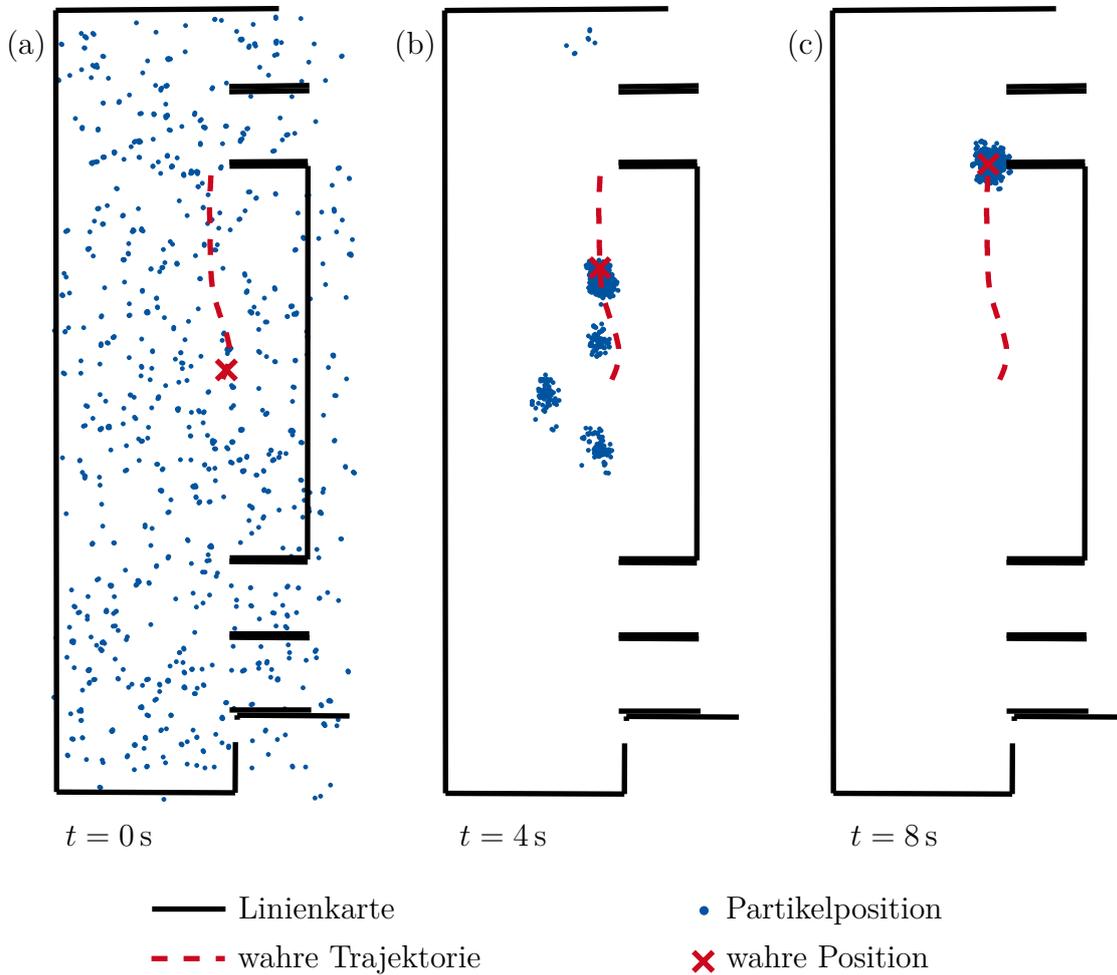


Abb. 3.5: Entwicklung der Punktwolke bei Initialisierung ohne Startpose

Mögliche Anwendungen der Initialisierung ohne GNSS-Informationen können sich ergeben, wenn etwa durch die Triangulation aus Sendeanlagen von Mobilfunk oder Vehicle-to-everything-Verkehrsvernetzung (V2X) eine sehr grobe Positionsbestimmung möglich ist, zugleich jedoch keine satellitenbasierte Positionslösung vorliegt. In der vorliegenden Arbeit werden diese alternativen Lokalisierungstechnologien jedoch nicht implementiert, sodass im Folgenden von einer satellitenbasierten Initialisierung ausgegangen wird.

3.4.5 Parametrierung

Der Lokalisierungsalgorithmus nutzt verschiedene Parameter, die wesentlichen Einfluss auf die Funktion und die Genauigkeit der Lokalisierung haben. Die Festlegung der Parameter ist spezifisch für die Definition der digitalen Karte und die Eigenschaften der genutzten Sensorsysteme. Tabelle 3.5 zeigt die für die vorliegende Anwendung gewählte Parametrierung. Die Werte wurden auf Basis von Experimenten im Testfeld iterativ festgelegt.

Tabelle 3.5: Parameter des Lokalisierungsalgorithmus

Symbol	Bezeichnung	Wert
T	Samplezeit	0.02 s
N	Anzahl Partikel	1000
$d_{min,Korr}$	Minimale Distanz zwischen zwei Messupdate-Schritten	0.3 m
σ_{Odo}	Standardabweichung Odometriefehler	0.0035 m
σ_{IMU}	Standardabweichung Gierratenfehler IMU	0.002 rad s ⁻¹
σ_L	Standardabweichung Distanzfehler LiDAR	1.8 m
σ_G	Standardabweichung initialer GNSS-Fehler in x und y	5 m
$\sigma_{G,\varphi}$	Standardabweichung initialer GNSS-Fehler in φ	0.5 rad

Bei der Anzahl der Partikel muss eine Abwägung zwischen Rechenzeit und Genauigkeit bzw. Robustheit getroffen werden. Grundsätzlich führt eine größere Partikelzahl zu einer Verbesserung der Ergebnisse der Lokalisierung, da hiermit eine dichtere Abbildung der Wahrscheinlichkeitsverteilungen der Positionen x und y und der Orientierung φ des Fahrzeuges gewährleistet wird. Jedes Partikel kann hierbei als eine diskrete Stützstelle der entsprechenden Wahrscheinlichkeitsverteilung angesehen werden. Zudem bieten mehr Partikel die Möglichkeit, die Stützstellen über einen breiteren Bereich um die geschätzte Pose zu verteilen, ohne dabei Einbußen bei der Genauigkeit in Kauf nehmen zu müssen, sodass die Robustheit erhöht wird. Da die Rechenzeit des Algorithmus in etwa linear mit der Anzahl an Partikeln steigt, wird die Partikelzahl allerdings durch die zur Verfügung stehende Rechenleistung begrenzt. Im vorliegenden Fall kann mit einer Zahl von 1000 Partikeln sicher die geforderte Samplezeit T eingehalten werden.

Die in Kapitel 3.4.3 beschriebene Startbedingung für die Ausführung eines neuen Korrekturschrittes des Partikelfilters wird durch eine minimale gefahrene Distanz seit dem letzten Korrekturschritt definiert. Mit dem Zahlenwert von $d_{min,Korr} = 0.3$ m lässt sich ein zu starkes Komprimieren der Partikelwolke bei langsamer Fahrt bzw. bei Stillstand des Fahrzeuges effektiv verhindern.

Bei der Parametrierung der Unsicherheiten von LiDAR-, IMU- und Odometriesensoren spielt das Verhältnis der entsprechenden Zahlenwerte eine große Rolle. σ_{IMU} und σ_{Odo}

haben zur Folge, dass bei jedem Prädiktionsschritt des Partikelfilters ein Fehler addiert wird, der aus einer Gaußschen Wahrscheinlichkeitsverteilung mit der entsprechenden Standardabweichung gezogen wird. Große Werte führen somit dazu, dass die Partikelwolke bezüglich der Fahrzeugorientierung bzw. der Position in Bewegungsrichtung stärker aufgeweitet wird. Im Gegensatz dazu kann σ_L als Gewichtungsfaktor für die Korrekturen auf Basis des Laserscanners angesehen werden. Die Annahme kleinerer Unsicherheiten des Laserscanners führt dazu, dass Abweichungen zwischen den virtuellen Distanzmessungen der Partikel und den realen LiDAR-Messungen zu einer stärkeren Reduktion der Partikelgewichte führen. Aufgrund der kleineren Gewichte werden Partikel mit größeren Abweichungen zur digitalen Karte im Zuge des Resamplings somit schneller aussortiert und es kommt zu einer Konzentration der Partikelwolke. Der letztere Mechanismus führt tendenziell zu einer Verbesserung der Genauigkeit und zu einer Beschleunigung der Konvergenz. Eine Verbesserung der Robustheit wird dagegen durch eine breitere Partikelwolke und ein langsames Aussortieren von Partikeln erzielt.

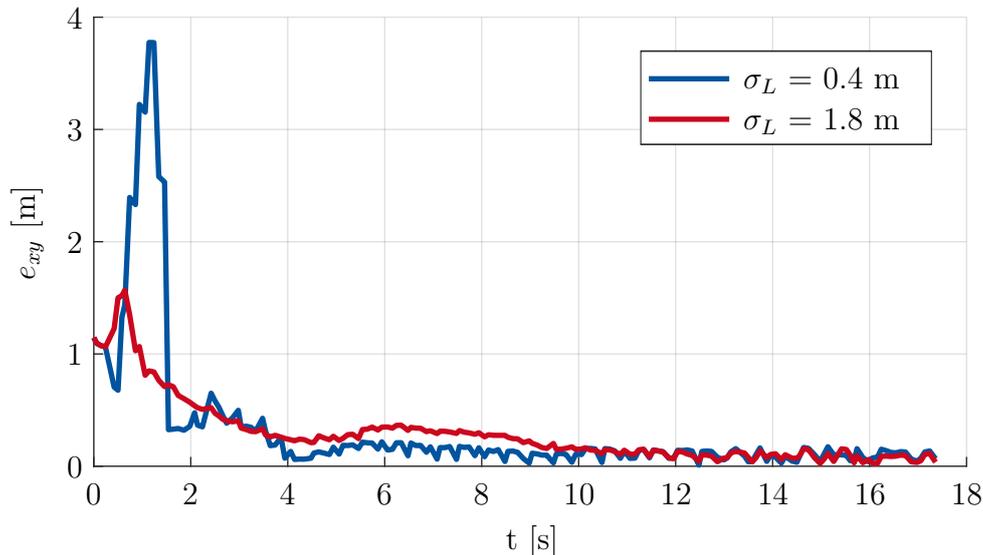


Abb. 3.6: Konvergenzverhalten der Lokalisierung in Abhängigkeit der LiDAR-Unsicherheit

Abbildung 3.6 verdeutlicht diesen Zusammenhang anhand der simulativen Auswertung einer Testfahrt mit verschiedenen Werten für σ_L . Gezeigt ist der euklidische Abstandsfehler e_{xy} zwischen der vom Lokalisierungsalgorithmus bestimmten Position und der Referenzposition. Es ist erkennbar, dass der initiale Fehler beim Start des Algorithmus etwa 1.1 m beträgt. Die LiDAR-Unsicherheit von $\sigma_L = 1.8$ m führt zu einer vergleichsweise langsamen, kontinuierlichen Konvergenz des Algorithmus zu einem Fehler im Bereich von 10–15 cm. Wird mit $\sigma_L = 0.4$ m eine stärkere Gewichtung der LiDAR-Korrekturen vorgenommen, so sind deutlich dynamischere Änderungen der Positionsschätzung festzustellen. Der Fehler erreicht im Vergleich schneller den niedrigen Bereich von 10–15 cm,

dafür entstehen durch das schnelle Aussortieren der Partikel in den ersten Sekunden des Testlaufes kurzzeitig wesentlich größere Abweichungen. Dieses Verhalten bietet somit das Potenzial präziserer und schnellerer Lokalisierungsergebnisse. Es besteht allerdings die Gefahr einer Divergenz, wenn aufgrund kurzfristiger Abweichungen zwischen Karte und Umfeldmessung zu viele Partikel fälschlicherweise eliminiert werden. Im vorliegenden Fall wird bei der Parametrierung der Fokus auf robustes Verhalten gelegt und somit mittels einer größeren LiDAR-Unsicherheit eine stärkere zeitliche Dämpfung des Korrekturschrittes vorgenommen.

3.4.6 Gesamtsystem Fahrzeug

Das auf dem Fahrzeug implementierte Softwaresystem umfasst neben dem Lokalisierungsalgorithmus auch weitere Prozesse, die insbesondere für die Vorgänge beim Einfahren in den kartierten Bereich und beim Verlassen desselben von Bedeutung sind. Abbildung 3.7 zeigt einen Überblick über die Zustände und Prozesse des Fahrzeugsystems.

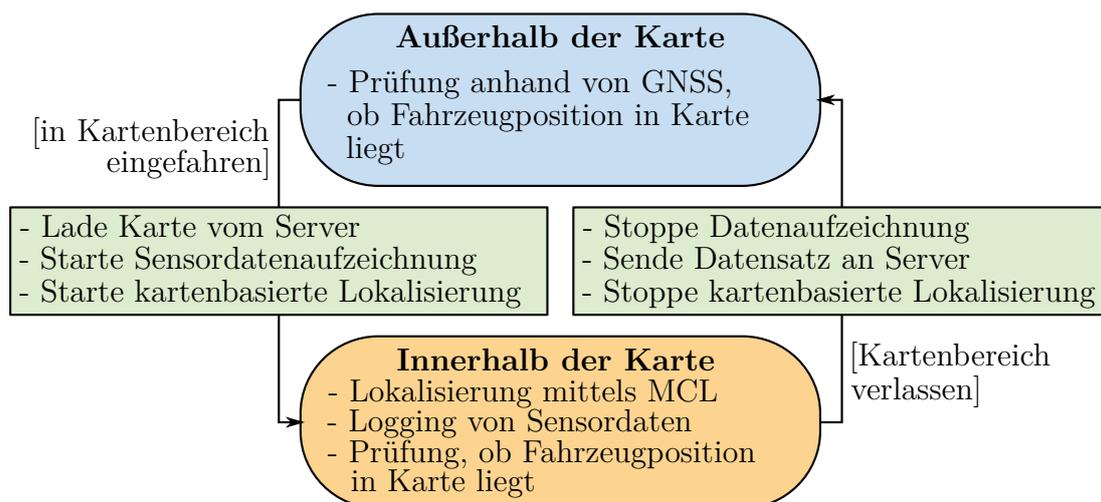


Abb. 3.7: Zustände und Prozesse der Fahrzeugsoftware

Die zwei Systemzustände betreffen die Fahrt außerhalb und innerhalb kartierter Areale. Befindet sich das Fahrzeug in einem Bereich, für den keine digitale Karte zur Verfügung steht, so wird ausschließlich der GNSS-Empfänger für die Lokalisierung genutzt. Anhand dieser Positionsinformation wird kontinuierlich überprüft, ob das Fahrzeug in einen kartierten Bereich einfährt. Die Kartengrenzen müssen dem Fahrzeug im Vorhinein bekannt sein oder sie müssen – etwa per Broadcast – vom Infrastruktursystem bereitgestellt werden. Sie sind als Polygone mit einer variablen Anzahl an georeferenzierten Eckpunkten

definiert. Beim Einfahren in ein kartiertes Areal lädt das Fahrzeug die entsprechende digitale Karte über die Fahrzeugschnittstelle des Infrastruktursystems (siehe Kapitel 4.4.4) herunter und startet die kartengestützte Lokalisierung. Zugleich wird auch mit der Aufzeichnung der für die Kartierung benötigten Sensordaten begonnen. Ergibt der kontinuierliche Abgleich der Fahrzeugposition mit den Kartengrenzen ein Verlassen des kartierten Bereiches, so wird die Datenaufzeichnung beendet, der gewonnene Datensatz an das Infrastruktursystem versendet und die kartengestützte Lokalisierung gestoppt. Es steht dann wieder lediglich die GNSS-basierte Positionslösung zur Verfügung.

4 Infrastrukturbasierter Kartendienst

Ein wesentlicher Gegenstand der vorliegenden Arbeit ist die Entwicklung eines infrastrukturbasierten Kartendienstes. Die für die Lokalisierung genutzten digitalen Karten müssen hier nicht im Vorhinein auf dem Lokalisierungsrechner des Fahrzeuges abgespeichert werden, sondern es werden mittels mobiler Datenkommunikation die jeweils erforderlichen, stets aktuellen Kartendaten bei Bedarf auf das Fahrzeug übertragen.

In diesem Kapitel wird die infrastrukturseitige Komponente betrachtet, die neben der Bereitstellung der Kartendaten vor allem die Aufgabe der kontinuierlichen Aktualisierung des Kartenmaterials auf Basis von übermittelten Fahrzeugdaten übernimmt. Es werden zunächst die Anforderungen an den infrastrukturbasierten Kartendienst im vorliegenden Anwendungsfall diskutiert, anschließend der Stand der wissenschaftlichen Forschung zusammengefasst und schließlich die Konzeption und Umsetzung des im Rahmen der vorliegenden Arbeit entwickelten Kartendienstes vorgestellt.

4.1 Anforderungen

Als Randbedingung für die Konzeption des infrastrukturbasierten Kartendienstes muss zunächst die im vorgesehenen Entwicklungsumfeld verfügbare Rechenhardware berücksichtigt werden. Ziel ist die Implementierung des Kartendienstes auf einer MEC-Plattform der Firma MECSware. Die Plattform nutzt Linux-basierte virtuelle Maschinen für die simultane Realisierung verschiedener Infrastrukturdienste. Für die Kommunikation werden virtuelle TCP/IP-kompatible Netzwerkschnittstellen eingesetzt. Die MEC-Architektur beinhaltet eine dezentrale Organisation des Gesamtsystems. Jede Recheneinheit verwaltet also die Karte für eine definierte und eindeutig abgegrenzte Region. Durch Überdeckungen mit benachbarten Regionen lassen sich die Übergangsszenarien zwischen den Zellen realisieren und es lässt sich eine große Sicherheit gegen einen Ausfall des Gesamtsystems darstellen, da der Ausfall einer Zelle keine Auswirkung auf die benachbarten Recheneinheiten hat.

Hinsichtlich der für die Kartierung zu nutzenden Sensordaten werden in Übereinstimmung mit Kapitel 3.1 neben Punktwolken aus einem LiDAR-Sensor lediglich solche Daten einbezogen, die in aktuellen Serienfahrzeugen vorausgesetzt werden können. Konkret sind dies Positionsdaten eines Einfrequenz-GNSS-Empfängers, Raddrehzahlsignale sowie Drehraten

Tabelle 4.1: Anforderungen Infrastrukturbasierter Kartendienst

Merkmal	Anforderung
Zielhardware	MECSware Mobile Edge Computing Plattform [55]
Eingehende Daten	Rohdaten der Fahrzeugsensoren: LiDAR, Einfrequenz-GNSS, Inertialmesseinheit und Rad-drehzahlsignale
Systemstruktur	dezentral
Datenformat digitale Karte	georeferenzierte Karte statischer und halbstatistischer geometrischer Merkmale
Reaktionszeit für Kartenanforderung	0.2 s
Reaktionszeit für Kartenupdate	1 min

aus einer inertialen Messeinheit. Da die Umgebung mittels eines LiDAR-Sensors abgetastet wird, der geometrische Umgebungsinformationen liefert, sind die zu kartierenden Informationen ebenfalls geometrische Merkmale. Darunter fallen insbesondere Hauswände, parkende Fahrzeuge und Elemente der Stadtmöblierung wie Bushaltestellen, Werbetafeln etc. Zu kartieren sind lediglich solche Merkmale, die ihre Position über mehrere Minuten nicht ändern, also als Referenz für die Lokalisierung nachfolgender Fahrzeuge geeignet sind. Bewegte Objekte sollen nicht in die Karte übernommen werden.

Bei den Reaktionszeiten des Gesamtsystems werden zwei Fälle unterschieden. Für die Anforderung einer Karte durch ein in den kartierten Bereich einfahrendes Fahrzeug wird eine Reaktionszeit von 0.2 s definiert. Ein mit der in Innenstädten üblichen Geschwindigkeit von 50 km h^{-1} fahrendes Fahrzeug legt damit bei der Einfahrt in die Kartenzelle weniger als 3 m zurück, bevor die Karte zur Verfügung steht. Für den Fall des Kartenupdates wird eine wesentlich längere Reaktionszeit von 1 min definiert – ein geändertes geometrisches Merkmal innerhalb des kartierten Bereiches kann also mit größerem zeitlichen Verzug verarbeitet werden. Da die Karte keine dynamischen Objekte beinhaltet und Updates ohnehin nur erfolgen können, wenn Daten von durchfahrenden Fahrzeugen zur Verfügung stehen, lässt eine schnellere Verarbeitung der Daten keine wesentliche Erhöhung der Leistungsfähigkeit des Systems erwarten.

In Tabelle 4.1 ist eine Zusammenfassung der beschriebenen Anforderungen an den infrastrukturbasierten Kartendienst gegeben.

4.2 Stand der Forschung

Im Bezug auf den infrastrukturbasierten Kartendienst können zwei wesentliche methodische Fragestellungen identifiziert werden. Zunächst ist zur Erstellung einer konsistenten Karte die korrekte Verknüpfung der von der Umfeldsensorik aufgezeichneten Daten erforderlich. Bei der Fahrt durch den kartierten Bereich befindet sich der Sensor in Bewegung, sodass Umgebungsmerkmale von unterschiedlichen Positionen aus erfasst werden. Für die Transformation der Messdaten in ein globales Koordinatensystem ist somit die Kenntnis über die Lage des Sensorkoordinatensystems zu jedem Messzeitpunkt erforderlich. Aus der Literatur bekannte Verfahren zur Rekonstruktion dieser Trajektorie werden in Kapitel 4.2.1 vorgestellt.

Die zweite wesentliche Fragestellung betrifft die Methodik zur kontinuierlichen Aktualisierung einer bestehenden digitalen Karte auf Basis neuer Sensorinformationen. Entsprechende Publikationen werden in Kapitel 4.2.2 vorgestellt und analysiert.

Selbstverständlich besteht ein wesentliches Merkmal zur Klassifikation der in der Literatur beschriebenen Kartierungsverfahren in der Definition der kartierten Merkmale. Da hier allerdings die gleichen Merkmale infrage kommen, die auch für die kartengestützte Lokalisierung genutzt werden, wird an dieser Stelle auf Kapitel 3.2.1 verwiesen.

4.2.1 Ausrichtung der Sensordaten

Bei der Ausrichtung der Sensordaten kann zunächst unterschieden werden zwischen solchen Verfahren, bei denen die erzeugte Karte georeferenziert ist, also Informationen über die Position der kartierten Merkmale in einem Weltkoordinatensystem enthalten sind und anderen Verfahren, die eine lokale Karte mit beliebig festgelegtem Koordinatenursprung nutzen. Erstere Verfahren nutzen in der Regel satellitenbasierte Positionsinformationen bei der Kartenerstellung; die so erzeugte Karte eignet sich gut für die Lokalisierung im regulären Straßenverkehr. Beispiele für die georeferenzierte Kartierung finden sich in [51], [78] und [53]. Lokale Karten werden häufig in Innenräumen oder in abgegrenzten Bereichen eingesetzt, wo eine Geoposition nicht unbedingt notwendig ist und kein Satellitenempfang zur Verfügung steht. In [41] werden solche Karten für eine Parkhausumgebung verwendet, [22] beschreibt einen entsprechenden Ansatz für Innenräume.

Bei der Ausrichtung der aus unterschiedlichen Positionen aufgenommenen Sensordaten spielen graphenbasierte Optimierungsverfahren eine große Rolle. Hierbei werden alle aufgenommenen Sensorinformationen in Form eines Posengraphen strukturiert. Dieser wird im Anschluss optimiert, wodurch die geometrischen Transformationen der aufgezeichneten Umgebungsdaten ermittelt werden können, die zur bestmöglichen Übereinstimmung aller gesammelten Informationen führen (siehe Kapitel 2.1.3). Graphenbasierte Verfahren ermöglichen die Integration vielfältiger Randbedingungen, darunter die Berücksichtigung

von sogenannten „Loop Closing“-Situationen. Hierbei kehrt das Messsystem nach längerer kontinuierlicher Bewegung wieder an einen bereits zuvor vermessenen Ort zurück, sodass eine Korrektur der über den dazwischen liegenden Messzeitraum kumulierten Fehler erfolgen kann. Aufgrund dieser Flexibilität sowie der Möglichkeit, eine globale Optimierung unter Berücksichtigung aller Sensorinformationen durchzuführen, werden graphenbasierte Verfahren in der Literatur sehr umfangreich eingesetzt. Beispiele finden sich unter anderem in [49], [77] und [86].

Satellitenbasierte Positionsdaten können sowohl in Posengraphen integriert, als auch direkt zur Ausrichtung von Sensordaten mit dem Ziel der Kartenerstellung genutzt werden. Insbesondere die Daten hochgenauer GNSS-Systeme, die mittels Echtzeitkinematik die Unsicherheiten der Satellitennavigation stark reduzieren, eignen sich für die direkte Transformation von Umgebungsinformationen ohne zusätzlichen Optimierungsschritt. Aus einer Sequenz von Satellitenpositionen kann zudem die Bewegungsrichtung und damit je nach kinematischer Konfiguration des Messsystems auch die Rotation des Umgebungssensors ermittelt werden. [78], [73] und [51] beschreiben einen entsprechenden Ansatz zur Generierung ihres Kartenmaterials.

Im Bereich der kartengestützten Lokalisierung auf Basis von Bilddaten spielt das Verfahren der Bündelblockausgleichung eine wichtige Rolle. Anhand von korrespondierenden Punkten überlappender photographischer Aufnahmen werden die dreidimensionalen Transformationen ermittelt, die eine bestmögliche Übereinstimmung der Merkmale zur Folge haben. Literaturquellen zu Anwendungen des Verfahrens sind [62], [46] und [77].

Bei der Ausrichtung von Punktwolken, die beispielsweise von LiDAR-Sensoren ausgegeben werden, können iterative Verfahren eingesetzt werden, die auf der Minimierung der Abstände zwischen assoziierten Punkten der beiden Punktwolken beruhen (siehe auch Kapitel 2.1.4.1). Eine einfache Möglichkeit zur Erzeugung dieser Assoziationen besteht in der Zuordnung der jeweils am nächsten liegenden Punkte („Nearest Neighbor“, „Closest Point“). Ist initial eine näherungsweise passende Transformation bekannt, kann so durch die iterative Wiederholung der Punktzuordnung sowie der Optimierung eine gute Ausrichtung erzielt werden. Entsprechende Literaturquellen sind [66], [22] und [57].

Einige in der Literatur dokumentierte Arbeiten verzichten auf eine vollautomatische Kartenerstellung. Stattdessen definiert ein menschlicher Anwender charakteristische Punkte, durch die Überschneidungen zwischen verschiedenen Messungen definiert werden können oder registriert manuell eingemessene GNSS-Referenzkoordinaten. Vorteilhaft ist hierbei die implizite Plausibilitätsprüfung durch den Menschen, die Fehler durch unzutreffende automatische Zuordnungen verhindert. Der manuelle bzw. teilautomatische Ansatz kommt in [37], [41] und [54] zum Einsatz.

Ein Überblick über die analysierten Literaturquellen mit den entsprechenden Kartierungsverfahren ist in Tabelle 4.2 gegeben.

Tabelle 4.2: Literaturübersicht Ausrichtung Sensordaten

Verfahren	Publikationen
GraphSLAM / graphenbasierte Optimierung	[89], [49], [50], [77], [67], [57], [86]
Geolokalisierung mittels GNSS RTK	[78], [73], [51]
Bündelblockausgleichung	[62], [46], [77]
Iterative Closest Point / Nearest-Neighbor-Optimierung	[66], [22], [57]
Manuelle / teilautomatisierte Kartierung	[37], [41], [54]

4.2.2 Kontinuierliche Aktualisierung des Kartenmaterials

Eine sehr häufig vorausgesetzte Grundannahme bei der Erstellung und Nutzung digitaler Karten geht von einer statischen, also im Hinblick auf die kartierten Merkmale unveränderlichen Welt aus. Für die kartengestützte Lokalisierung in städtischen Umgebungen bildet diese Annahme eine starke Einschränkung, da relevante Teile des von der Fahrzeugsensorik wahrnehmbaren Umfeldes Änderungen unterworfen sind. Um diese Änderungen in der Karte abbilden zu können, stellt sich die Frage nach Methoden zur kontinuierlichen Aktualisierung des Kartenmaterials, welche in der Literatur oft mit dem Stichwort „Lifelong Mapping“ assoziiert werden.

In [56] werden zwei alternative Ansätze zur Berücksichtigung von veränderlichen Umgebungen für die Anwendung in automatisierten Fahrzeugen diskutiert. Der erste Ansatz definiert für das Lokalisierungsszenario eine statische Karte sowie mehrere semi-statische temporäre Karten. Die statische Karte wird als gegeben vorausgesetzt und enthält nur solche Elemente, die als vollkommen unveränderlich angesehen werden können. Temporäre Karten werden automatisch erzeugt, wenn die Beobachtungen des Umgebungssensors nicht mit der statischen Karte konsistent sind. Bei der Lokalisierung wird zunächst die statische Karte genutzt – ergeben sich hier zu große Abweichungen, so wird die am nächsten liegende temporäre Karte genutzt oder es wird eine neue temporäre Karte erstellt, welche die veränderten Umgebungsmerkmale aufnimmt. Der zweite Ansatz in [56] basiert auf einer erweiterten Belegungskarte („Occupancy Grid Map“). Jede Zelle der als Gitter strukturierten Karte enthält hierbei zusätzlich zu der Information über die aktuelle Belegungswahrscheinlichkeit auch eine Wahrscheinlichkeit der Belegungsänderung. Damit wird gegenüber dem konventionellen Algorithmus zur Aktualisierung von Belegungskarten (siehe 2.2) eine verbesserte Dynamik in sich häufig verändernden Kartenbereichen erzielt.

Mehrere Speicherebenen für die verschiedenen Karteninformationen werden auch in [22] eingesetzt. Hier sind die Kartenebenen nach der zeitlichen Beständigkeit der Merkmale strukturiert: Ein Sensorspeicher dient als Vorfilter, in dem zunächst alle Sensordaten

abgelegt werden. Die für die Lokalisierung als geeignet erscheinenden Merkmale werden in den Kurzzeitspeicher überführt. Der Langzeitspeicher enthält nur solche Elemente, die aufgrund mehrerer übereinstimmender Beobachtungen im Kurzzeitspeicher über einen längeren Zeitraum bestätigt wurden. Die Methode bezieht sich explizit auf visuelle Merkmale, die automatisiert aus Bilddaten gewonnen werden.

Weitere Ansätze zur Berücksichtigung veränderlicher Umgebungsbedingungen bei der Kartierung auf Basis von Bilddaten werden in [17] und [57] vorgestellt. Der Fokus liegt hierbei allerdings weniger auf realen Veränderungen in der Umwelt, etwa durch umgeparkte Fahrzeuge, sondern mehr auf der Realisierung zuverlässiger visueller Lokalisierungsmethoden bei veränderlichen Licht- und Wetterbedingungen. Da die von Kameras aufgezeichneten Sensordaten sehr stark von diesen Einflüssen abhängig sind, werden je nach Umgebungsbedingung sehr unterschiedliche Landmarken erkannt. In [17] werden für verschiedene Messbedingungen („Observation Conditions“) spezielle Karten erstellt, von denen für die Lokalisierung die jeweils am besten passende Karte genutzt wird. [57] ermittelt aus den Daten mehrerer Messfahrten unter verschiedensten Wetterverhältnissen eine sogenannte „Summary Map“, welche die Landmarken enthält, die möglichst universell erkennbar waren. Das Konzept einer globalen Karte, welche auf Basis neuer Informationen bei Bedarf aktualisiert wird, verfolgt auch [77]. Hier wird bei jeder Durchfahrt durch den kartierten Bereich eine lokale Karte visueller Merkmale erzeugt. Diese wird im Anschluss mit der bereits existierenden globalen Karte durch einen graphenbasierten Ansatz abgeglichen, wobei die rechnerische Komplexität durch die Definition von kleinen Unterregionen, sogenannten Clustern, verringert wird.

Auf Basis von GNSS-Positionsdaten einer großen Fahrzeugflotte werden in [53] digitale Karten erzeugt und bei einem zentralen Kartendienst kontinuierlich aktualisiert. Der Fokus liegt hier auf den erforderlichen Datenbankstrukturen für die effiziente Verarbeitung der anfallenden Datenmengen. Die genutzten GNSS-Daten erlauben lediglich die Kartierung von Fahrspuren und deren Begrenzungen, weitere Umgebungsmerkmale werden nicht integriert.

Bei den meisten hier vorgestellten Verfahren wird die digitale Karte nur auf dem Lokalisierungscomputer eines Fahrzeuges verwaltet. Aktualisierungen der Karte können also nicht zwischen mehreren Verkehrsteilnehmern synchronisiert werden. Neben [53] bildet die Arbeit [17] eine Ausnahme. Hier wird eine Cloud-basierte Lösung beschrieben, bei der ein Zentralrechner die Aktualisierung des Kartenmaterials realisiert und die jeweils aktuellen Daten über eine geeignete Kommunikationsschnittstelle den Fahrzeugen zur Verfügung stellt. Ähnlich ist auch die „HD Live Map“ der Firma HERE aufgebaut [14]; eine Cloud sammelt die Sensordaten zahlreicher Fahrzeuge und berechnet Korrekturen des Kartenmaterials. Welche Kartenebenen und -informationen auf diese Weise aktualisiert werden können und wie die entsprechenden Algorithmen aufgebaut sind, ist hier nicht dokumentiert.

Insgesamt ist das Thema der kontinuierlichen Aktualisierung digitaler Karten in der Literatur noch vergleichsweise wenig untersucht worden. Anwendungen finden sich hauptsächlich für visuelle kartenbasierte Lokalisierungssysteme sowie für kleinere robotische Fahrzeuge.

4.3 Konzeption

Bei der Konzeption des infrastrukturbasierten Kartendienstes werden einerseits die in Kapitel 4.1 definierten Anforderungen berücksichtigt, andererseits fließen die aus der Literatur gewonnenen Erkenntnisse (siehe Kapitel 4.2) in die Konzeptentscheidungen ein. Ziel ist die Realisierung aller Funktionalitäten, die für einen Betrieb des Kartendienstes an einer realen, innerstädtischen Kreuzung mit V2X-Kommunikationshardware erforderlich sind.

4.3.1 Hardware und Implementierung

Im Bezug auf die Zielhardware des Systems werden die im Anforderungskapitel 4.1 definierten Randbedingungen umgesetzt. Der Kartendienst wird demnach in einer Linux-Umgebung implementiert, die in Form einer virtuellen Maschine auf der MEC-Plattform betrieben werden kann. Durch die Festlegung auf die Programmiersprache C++ ist die einfache Integration von OpenSource-Bibliotheken gewährleistet, die Basisfunktionen etwa für das Management von Punktwolken oder die nichtlineare Optimierung zur Verfügung stellen.

Der Prozess des zu implementierenden Kartenupdates, also der Aktualisierung der digitalen Karte auf Basis von Sensordaten durchfahrender Fahrzeuge, soll asynchron erfolgen – der Prozess ist also nicht an eine feste Taktrate gebunden, sondern wird im Rahmen der zur Verfügung stehenden Rechenleistung schnellstmöglich abgearbeitet. Die vom durchfahrenden Fahrzeug aufgezeichneten Daten werden also für eine Durchfahrt gesammelt und vor dem Verlassen des kartierten Bereiches in Gänze übertragen. Diese batchweise Verarbeitung macht einen dauerhaften, echtzeitfähigen Datenlink verzichtbar und kann aufgrund der begrenzten Größe des kartierten Bereiches zugleich die Anforderung aus Kapitel 4.1 bezüglich der Reaktionszeit des Kartenupdates erfüllen.

4.3.2 Digitale Karte

Das Format der digitalen Karte zur Fahrzeuglokalisierung ist darauf ausgelegt, einen möglichst effizienten und vollständigen Abgleich mit den vom Laserscanner des Fahrzeugs erfassten Punktwolken zu ermöglichen. Entsprechend den in Kapitel 3.3 spezifizierten

Zuständen zur Beschreibung der Fahrzeugpose wird auch der digitalen Karte eine zweidimensionale Ebene zugrunde gelegt, welche zur Fahrbahnoberfläche parallel ist. Kartiert werden alle geometrischen Objekte, die in dieser horizontalen Ebene liegen. Hierunter fallen insbesondere Häuserwände, geparkte PKW und Elemente der Stadtmöblierung. Durch den Schnitt der Konturen dieser Objekte mit der zweidimensionalen Kartenebene ergeben sich Linienzüge, die auch vom Laserscanner mit einer horizontalen Winkelauflösung von 0.4° abgetastet werden. In der digitalen Karte werden diese Linienzüge als Sequenz gerader Liniensegmente abgespeichert. Jedes Liniensegment ist dabei durch seinen Anfangs- und seinen Endpunkt eindeutig definiert. Insbesondere großflächige Objekte wie Häuserfassaden können auf diese Weise sehr speicherplatzeffizient abgebildet werden, da im Gegensatz zur dichten, punktwisen Abtastung mit dem Laserscanner für das entsprechende Liniensegment der digitalen Karte nur zwei zweidimensionale Koordinaten abgespeichert werden müssen. Bei gekrümmten Objekten kann durch die entsprechende Parametrierung ein Kompromiss zwischen einer genauen Abbildung der Geometrie mit vielen kurzen Geradensegmenten und einer speicherplatzeffizienten Approximation mit wenigen längeren Segmenten gewählt werden.

Die digitale Karte wird aus zwei Ebenen zusammengesetzt: Statische Elemente wie Häuserfassaden werden bei der Einrichtung des Kartendienstes manuell vermessen oder aus bereits verfügbaren Datenquellen übernommen. Sie werden sodann als unveränderliche Geometrien in der Karte abgelegt. Hier wird davon ausgegangen, dass das automatische Aktualisieren der entsprechenden Kartensegmente aufgrund von Messunsicherheiten eher zu einer Verminderung der Genauigkeit führt. Die Positionen von Gebäuden sind im Allgemeinen über einen sehr langen Zeitraum unveränderlich, sodass die erforderliche manuelle Korrektur bei Änderungen der Bebauung toleriert werden kann. Bei der Rekonstruktion der Trajektorie können die statischen Kartenelemente zudem als verlässliche Referenzen genutzt werden, um Laserscans präzise an bekannten Geometrien ausrichten zu können.

Alle anderen kartierten Elemente werden als halbstatistisch angesehen. Sie sind der aktualisierbaren Kartenebene zugeordnet, deren Kartensegmente auf Basis der Fahrzeugdaten verändert, entfernt oder erweitert werden können. Dynamische Objekte wie Fußgänger, fahrende PKW und andere Verkehrsteilnehmer dürfen nicht in die Karte übernommen werden. Bei der Kartenerstellung werden daher entsprechende Filterungsalgorithmen vorgesehen, die in Kapitel 4.3.4 genauer eingeführt werden.

4.3.3 Rekonstruktion der Trajektorie

Um die vom Fahrzeug aufgezeichneten Daten des Laserscanners für die Kartierung nutzen zu können, muss zunächst die Fahrzeugtrajektorie rekonstruiert werden, damit die Lage des Sensorkoordinatensystems zum Zeitpunkt jeder Messung in globalen Koordinaten bekannt ist. Da sich Ungenauigkeiten bei diesem Rechenschritt direkt auf die aus den Daten

ermittelten Änderungen der digitalen Karte auswirken, ist das Ziel des hier eingesetzten Algorithmus, die auf Basis aller verfügbaren Informationen bestmögliche Trajektorien-schätzung zu liefern. Zugleich sind die Anforderungen bezüglich der Rechenzeit weniger ausschlaggebend, da für die Aktualisierung der Karte eine relativ lange Reaktionszeit von 1 min toleriert wird (siehe Kapitel 4.1).

Aus diesen Gründen wird als Algorithmus für die Trajektorienrekonstruktion das GraphSLAM-Verfahren eingesetzt. Es ermöglicht nicht nur, alle *zu einem Zeitpunkt* zur Verfügung stehenden Sensordaten aus GNSS, IMU und Odometrie zu berücksichtigen, sondern formuliert *über den gesamten Zeitraum* der Messung ein Optimierungsproblem, das die Trajektorie mit der besten Übereinstimmung mit allen Messdaten ermittelt. Durch den Abgleich des bekannten statischen Layers der digitalen Karte mit den aufgezeichneten Laserscans einer Messfahrt können dem GraphSLAM-Optimierungsproblem zudem weitere Randbedingungen übergeben werden, die auch die LiDAR-Daten für die Trajektorienrekonstruktion nutzbar machen. Das gewählte Verfahren verspricht somit die für den Anwendungsfall bestmögliche Genauigkeit.

4.3.4 Aktualisierung der Karte

Die LiDAR-Daten, die bei der Durchfahrt eines Fahrzeuges durch den kartierten Bereich gewonnen werden, können anhand der rekonstruierten Trajektorie zu einer Kartendarstellung verarbeitet werden, die eine Abbildung des aktuellen Zustandes der Kreuzung ermöglicht. Eine einfache Variante der kontinuierlichen Aktualisierung der digitalen Karte könnte somit darin bestehen, stets die jeweils aktuellsten empfangenen Fahrzeugdaten für die Generierung einer Linienkarte zu nutzen und diese den nachfolgenden Fahrzeugen für die Lokalisierung bereitzustellen. Problematisch wäre in diesem Fall allerdings, dass die Daten eines Fahrzeuges im Allgemeinen nur Informationen über einen Teil des kartierten Bereiches enthalten, da ein einzelnes Fahrzeug in der Regel nicht alle Areale innerhalb des kartierten Bereiches befährt. Zudem würden sich Fehler, die bei der Aufzeichnung oder Verarbeitung der Daten eines Fahrzeuges auftreten, direkt auf die Qualität der bereitgestellten Karte und damit die Lokalisierung nachfolgender Fahrzeuge auswirken.

Aus diesen Gründen wird für den infrastrukturbasierten Kartendienst eine übergeordnete Karte definiert, die als sogenannte Masterkarte die Grundlage für die Lokalisierung nachfolgender Fahrzeuge bildet. Die Masterkarte erfüllt damit die Aufgabe, die Informationen von mehreren Fahrzeugdatensätzen zu filtern und zusammenzuführen, sodass eine konsistente und zugleich aktuelle Repräsentation der gesammelten Informationen entsteht.

Als Format für die Masterkarte wird eine gitterbasierte Belegungskarte (siehe Kapitel 2.2.2) eingesetzt. Der kartierte Bereich ist hierbei in diskrete Gitterzellen unterteilt, wobei für jede Zelle eine Belegungswahrscheinlichkeit gespeichert wird. Auf Basis der in

eingehenden Fahrzeugdaten enthaltenen Beobachtungen werden diese Belegungswahrscheinlichkeiten kontinuierlich aktualisiert. Wurde von einem durchfahrenden Fahrzeug in einer betrachteten Gitterzelle ein LiDAR-Echo erfasst, so wird in der Masterkarte die entsprechende Belegungswahrscheinlichkeit erhöht; wurden Bereiche vom Fahrzeug als frei erkannt, erfolgt eine Reduktion der Belegungswahrscheinlichkeit. Durch diese probabilistische Betrachtung der kartierten Geometrien wird eine automatische Filterung der Informationen und damit eine erhöhte Robustheit gegenüber falschen oder ungenauen Eingangsdaten erzielt.

Bei der Aktualisierung der Masterkarte werden zeitabhängige Gewichte genutzt. Auf diese Weise kann erreicht werden, dass sowohl in Zeiten mit geringem Verkehrsaufkommen, als auch bei vielen durch das kartierte Gebiet fahrenden Fahrzeugen die entsprechenden Beobachtungen in angepasster Weise in der Masterkarte repräsentiert werden. Liegt zwischen mehreren erhaltenen Fahrzeugdatensätzen eine sehr kurze Zeitspanne, so ist davon auszugehen, dass sich das kartierte Gebiet in der Zwischenzeit nicht wesentlich verändert hat. Durch eine niedrigere Gewichtung dieser Informationen kann dann verhindert werden, dass die Belegungswahrscheinlichkeiten in der Masterkarte übermäßig verstärkt werden. Ist dagegen seit den letzten verarbeiteten Fahrzeugdaten viel Zeit vergangen, können die entsprechenden Informationen im Verhältnis stärker gewichtet werden, um die schnelle Anpassung der Masterkarte an mögliche Änderungen im kartierten Gebiet zu gewährleisten.

Um die Linienkarte für die Lokalisierung zu erzeugen, müssen auf Basis der belegten Zellen der Masterkarte passende Liniensegmente generiert werden, welche die zugrunde liegenden geometrischen Objekte möglichst gut abbilden. Der hierfür eingesetzte Algorithmus beruht auf dem Random Sample Consensus (RANSAC)-Verfahren, bei dem durch das Testen zufälliger Kombinationen von Anfangs- und Endpunkten auf effiziente Weise Geraden detektiert werden können.

4.4 Umsetzung

Für die Umsetzung des in Kapitel 4.3 eingeführten Konzeptes des infrastrukturbasierenden Kartendienstes ergibt sich die in Abb. 4.1 dargestellte Systemstruktur. Aus den zur Aktualisierung der Karte verfügbaren Daten wird zunächst unter Verwendung des GraphSLAM-Algorithmus in einem zweistufigen Optimierungsverfahren die vom Fahrzeug gefahrene Trajektorie rekonstruiert. Diese wird im anschließenden Verarbeitungsschritt für die Erzeugung der gitterbasierten Belegungskarte verwendet, aus der schließlich die benötigte Linienkarte extrahiert wird. Die folgenden Kapitel 4.4.1, 4.4.2 und 4.4.3 beschreiben diese Prozessschritte im Detail.

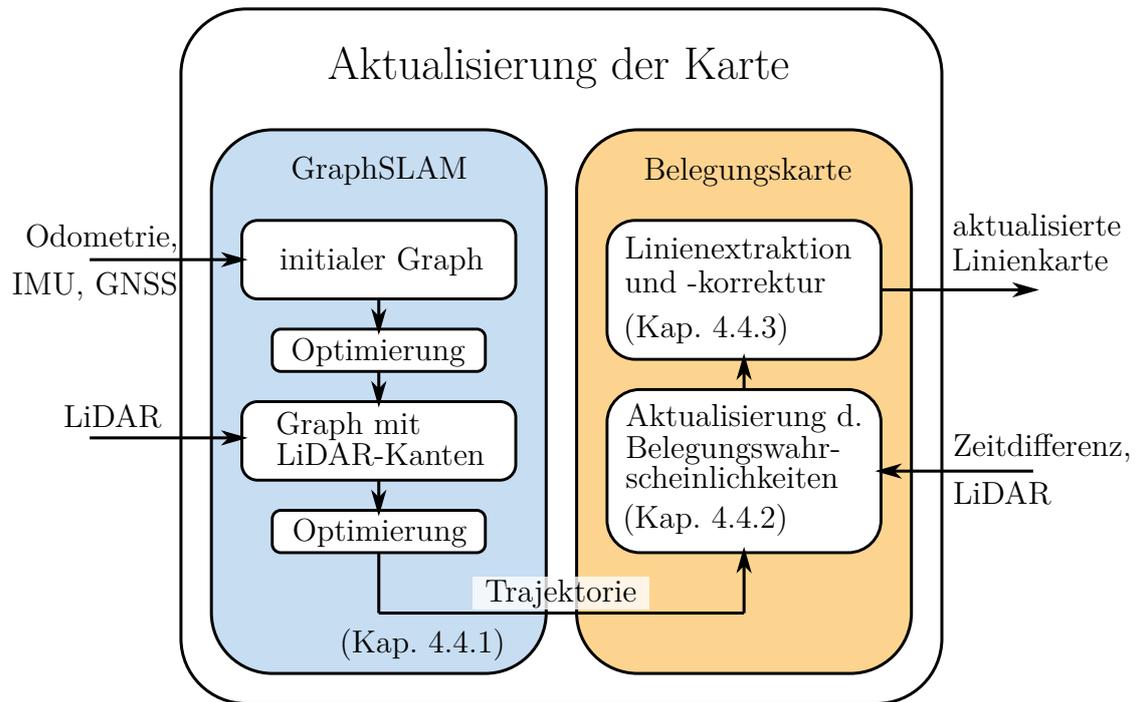


Abb. 4.1: Überblick der Komponenten des infrastrukturbasierten Kartendienstes

Vorarbeiten für den infrastrukturbasierten Kartendienst wurden in den betreuten Masterarbeiten [5] und [6] erbracht. Die eigenen Veröffentlichungen [4], [1] und [2] betreffen ebenfalls diesen Themenschwerpunkt.

4.4.1 GraphSLAM

Für die Rekonstruktion der Fahrzeugtrajektorie aus den aufgezeichneten Messdaten wird mittels des GraphSLAM-Verfahrens eine Optimierung durchgeführt. Ziel ist es, alle verfügbaren Informationen möglichst vollständig auszunutzen, um die größtmögliche Genauigkeit bei der Bestimmung der Fahrzeugposen zu erzielen. In Kapitel 2.1.3 wurden bereits die Grundlagen von GraphSLAM eingeführt, sodass hier auf die für die Anwendung spezifischen Eigenschaften der Implementierung eingegangen werden soll.

Das Optimierungsverfahren wird unter Verwendung von g2o (siehe [47]) als mehrschrittiger Prozess umgesetzt. Im ersten Schritt werden die Knoten des Graphen erzeugt, die die zu optimierenden Zustände repräsentieren. Jeder Knoten beinhaltet somit den Zustand $\vec{x} = (x, y, \varphi)$ bestehend aus Position und Orientierung des Fahrzeuges im lokalen ENU-Koordinatensystem zu einem definierten Zeitpunkt. Die erfassten Zeitpunkte ergeben sich

aus den Messungen des Sensorsystems mit der höchsten Messfrequenz. Im vorliegenden Fall werden sowohl die Raddrehzahlgeber, als auch die IMU mit 100 Hz abgetastet, sodass der Graph 100 Knoten pro Sekunde Messzeit enthält.

Die Knoten des Graphen müssen für die spätere Anwendbarkeit des iterativen Optimierungsverfahrens mit sinnvollen Startwerten initialisiert werden. Hierfür wird zunächst der erste Knoten betrachtet, für dessen Position (x, y) die erste GNSS-Messung des Datensatzes zugrunde gelegt wird. Die Orientierung wird bestimmt, indem die erste GNSS-Messung ermittelt wird, deren Abstand zur initialen Position einen Grenzwert überschreitet; zwischen beiden Punkten wird dann der Winkel berechnet. Für die Initialisierung aller weiteren Knoten wird ausgehend vom ersten Knoten eine Koppelnavigation mittels der Odometrie- und IMU-Daten durchgeführt. Hierbei wird aus der Pose jedes Knotens die Pose des nachfolgenden Knotens durch das inkrementelle Addieren der gemessenen Positions- und Orientierungsänderungen berechnet.

Dem so initialisierten Graphen werden anschließend Kanten hinzugefügt, welche die aus den verschiedenen Messdaten abgeleiteten Randbedingungen repräsentieren. Jede Kante verbindet zwei Knoten miteinander und liefert eine Information über die relative Lagebeziehung beider Knoten. Daneben wird jeder Kante eine Kovarianzmatrix zugeordnet, die die Unsicherheit der zugehörigen Messungen quantifiziert. Da von der statistischen Unabhängigkeit der einzelnen Messgrößen ausgegangen wird, sind die genutzten Kovarianzmatrizen Diagonalmatrizen.

Odometriekanten werden zwischen allen zeitlich direkt aufeinander folgenden Knoten erzeugt. Sie nutzen die bereits bei der Initialisierung verwendeten Daten aus Raddrehzahl- und Inertialsensorik und beschreiben die zwischen zwei Zeitpunkten gemessene Translation $(\Delta x, \Delta y)$ sowie die entsprechende Rotation $\Delta\varphi$ des Fahrzeuges um die Hochachse. Für die Berechnung der Transformation wird das bereits in Kapitel 3.4.1 im Zusammenhang mit der MCL eingeführte kinematische Fahrzeugmodell genutzt (siehe Abbildung 3.2). Die für die Kante benötigten Größen lassen sich damit aus den Sensordaten ω_{hl} , ω_{hr} der Raddrehzahlsensorik sowie $\dot{\varphi}$ der IMU unter Berücksichtigung der Zeitdifferenz zwischen den betrachteten Knoten Δt_K wie folgt berechnen.

$$\Delta x = 0.5 \cdot (\omega_{hl} \cdot r_{hl} + \omega_{hr} \cdot r_{hr}) \cdot \Delta t_K \cdot \cos(0.5 \cdot \dot{\varphi} \cdot \Delta t_K) \quad (4.1)$$

$$\Delta y = 0.5 \cdot (\omega_{hl} \cdot r_{hl} + \omega_{hr} \cdot r_{hr}) \cdot \Delta t_K \cdot \sin(0.5 \cdot \dot{\varphi} \cdot \Delta t_K) \quad (4.2)$$

$$\Delta\varphi = \dot{\varphi} \cdot \Delta t_K \quad (4.3)$$

r_{hl} und r_{hr} bezeichnen hierbei die dynamischen Abrollradien der beiden Hinterräder des Fahrzeuges. Sie werden als bekannt vorausgesetzt und als konstante Werte modelliert.

GNSS-Kanten werden auf Basis der Daten des Satellitenempfängers generiert. Da auch diese Kanten als Verbindung zweier Knoten definiert werden, wird für jede GNSS-Messung

ein zusätzlicher Knoten erzeugt, der die gemessene Position x_{gnss}, y_{gnss} beinhaltet. Die so neu erzeugten Knoten werden bei der späteren Optimierung als unveränderliche Zustände betrachtet. Jeder GNSS-Knoten wird mit dem zum selben Zeitpunkt gehörenden Knoten der Fahrzeugpose über eine Kante mit der Translation $(\Delta x, \Delta y) = (0, 0)$ verbunden. Der durch diese Kante erzeugte Fehlerterm nimmt also den Wert 0 an, wenn die Fahrzeugposition exakt der GNSS-Messung entspricht. Andernfalls ergibt sich der Fehler über das euklidische Abstandsmaß und die zur Kante gehörige Kovarianzmatrix.

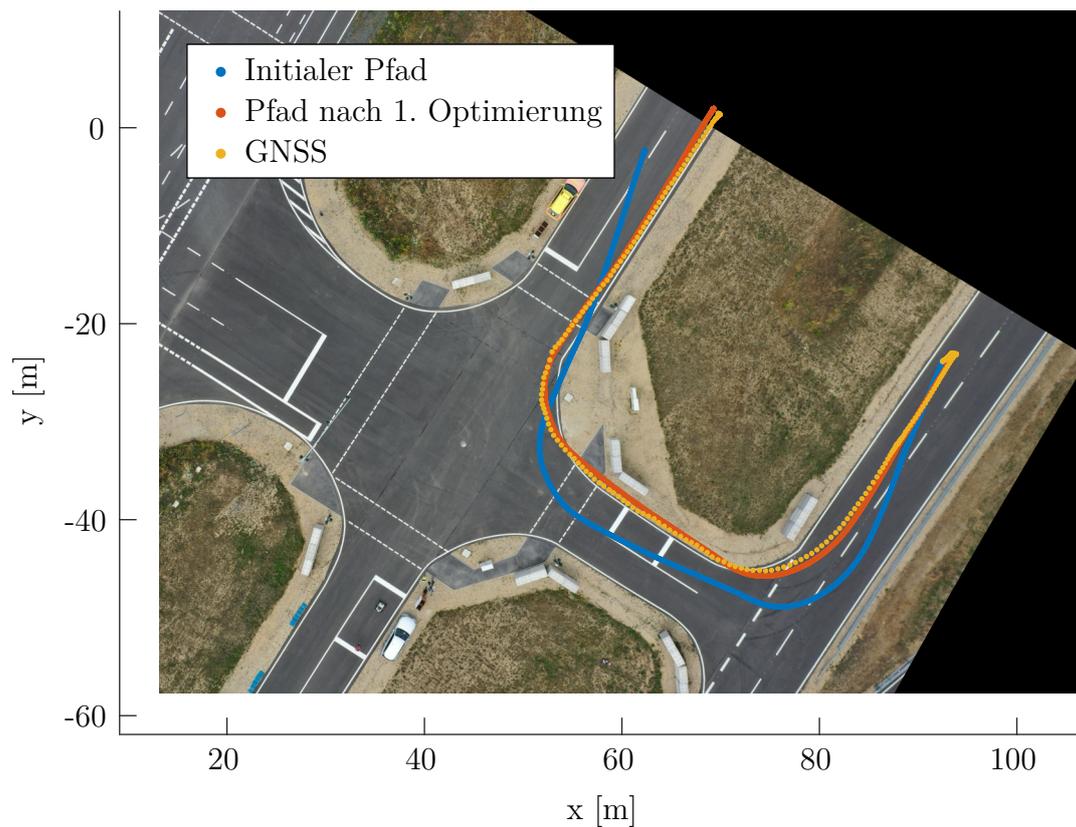


Abb. 4.2: Geschätzte Fahrzeugtrajektorie nach erstem Optimierungsschritt ohne LiDAR-Kanten

Für die Generierung der **LiDAR-Kanten** soll ein paarweiser Vergleich der im Fahrzeugdatensatz enthaltenen Punktwolken des Laserscanners mit den kartierten Geometrien erfolgen. Die im Graphen zu erfassende Information über die Fahrzeugpose ergibt sich dann aus der Transformation, die zu einer optimalen Ausrichtung der Geometrien führt. In Kapitel 2.1.4 wurden die Grundlagen üblicher Methoden zur Ausrichtung von Punktwolken eingeführt. Eine wesentliche Voraussetzung für die Anwendung dieser Methoden liegt in der Verfügbarkeit einer guten initialen Ausrichtung. Um diese zu generieren, wird

vor der Erzeugung der LiDAR-Kanten eine Optimierung des Graphen mit Odometrie- und GNSS-Kanten durchgeführt. Driftfehler sowie Fehler der initialen Ausrichtung, die bei der Initialisierung der Knoten mittels Koppelnavigation entstehen können, werden dadurch ausgeglichen und es steht die auf Basis von GNSS, Odometrie- und Inertialsensoren optimale Trajektorie für den Abgleich der LiDAR-Daten zur Verfügung. Abbildung 4.2 zeigt am Beispiel eines Fahrzeugdatensatzes die Auswirkung dieses ersten Optimierungsschrittes. Die Trajektorien sind einem georeferenzierten Luftbild der Testumgebung überlagert. Es ist erkennbar, dass die Initialisierung der Knoten anhand der Koppelnavigation die grundsätzliche Form der Trajektorie zwar gut wiedergibt, die Schätzung der initialen Ausrichtung allerdings aufgrund der großen Unsicherheit des GNSS-Empfängers stark fehlerbehaftet ist. Nach der Optimierung ist die geschätzte Trajektorie gut an die Stützstellen des GNSS-Empfängers angepasst, wobei für die Bestimmung der Form bzw. der Krümmungen weiterhin die hochfrequenten Daten von Odometrie und IMU eine große Rolle spielen. Die Trajektorie verläuft daher nicht exakt durch alle GNSS-Punkte, sondern gleicht Ungenauigkeiten des Satellitenempfängers auf Basis der Inertial- und Raddrehzahlmessungen aus.

Der wesentliche Mangel der nach der ersten Optimierung bestimmten Trajektorie liegt darin, dass systematische Fehler des GNSS-Empfängers nicht ausgeglichen werden können. In der Praxis können die Fehler des Satellitenempfängers nicht durch eine mittelwertfreie Verteilung beschrieben werden, sondern es treten deutliche, systematische Offset-Fehler auf, die zu einer parallelen Verschiebung der optimierten Trajektorie führen. Die absolute Lage wird daher durch einen Abgleich der LiDAR-Daten mit der digitalen Karte ermittelt. Für den Abgleich werden zwei alternative Verfahren genutzt, deren Vor- und Nachteile Gegenstand der Untersuchung sind. Beide Verfahren nutzen jeden im Messdatensatz enthaltenen Laserscan, um die zur optimalen Ausrichtung an der Referenzkarte erforderliche Transformation $(\Delta x, \Delta y, \Delta \varphi)$ zu ermitteln.

Beim ICP-Verfahren (siehe Kapitel 2.1.4.1) wird aufgrund der Vielzahl an Messdaten, für die der paarweise Vergleich durchgeführt werden muss, die rechenzeiteffiziente Punkt-zu-Punkt-Fehlermetrik verwendet. Um hierfür Punktkorrespondenzen zwischen Laserscan und digitaler Karte aufstellen zu können, wird die aktuelle Linienkarte zunächst einmalig durch diskrete Abtastung in eine Punktwolke transformiert. Die benötigten Punktkorrespondenzen können sodann mit einer „nächster Nachbar“-Suche aufgestellt werden. Die Implementierung des ICP-Verfahrens basiert auf der frei verfügbaren Softwarebibliothek „Point Cloud Library“ (PCL) [69].

Das direkte Scan-Matching-Verfahren (siehe Kapitel 2.1.4.2) erfordert im ersten Schritt die Berechnung eines diskreten Wahrscheinlichkeitsfeldes, in dem diejenigen Zellen hohe Werte aufweisen, in denen der Lasersensor mit hoher Wahrscheinlichkeit ein Echo detektiert. Um eine ausreichend hohe Effizienz des Algorithmus zu erzielen, wird das Verfahren zunächst mit einer groben Diskretisierung durchgeführt. Die Berechnung mit der feinen Diskretisierungsstufe wird sodann nur in den Bereichen durchgeführt, die auf Basis

des ersten Durchlaufes als vielversprechend erkannt wurden. Abbildung 4.3 zeigt das Wahrscheinlichkeitsfeld in der feinen Diskretisierungsstufe für eine Testumgebung mit mehreren kastenförmigen Hindernissen.

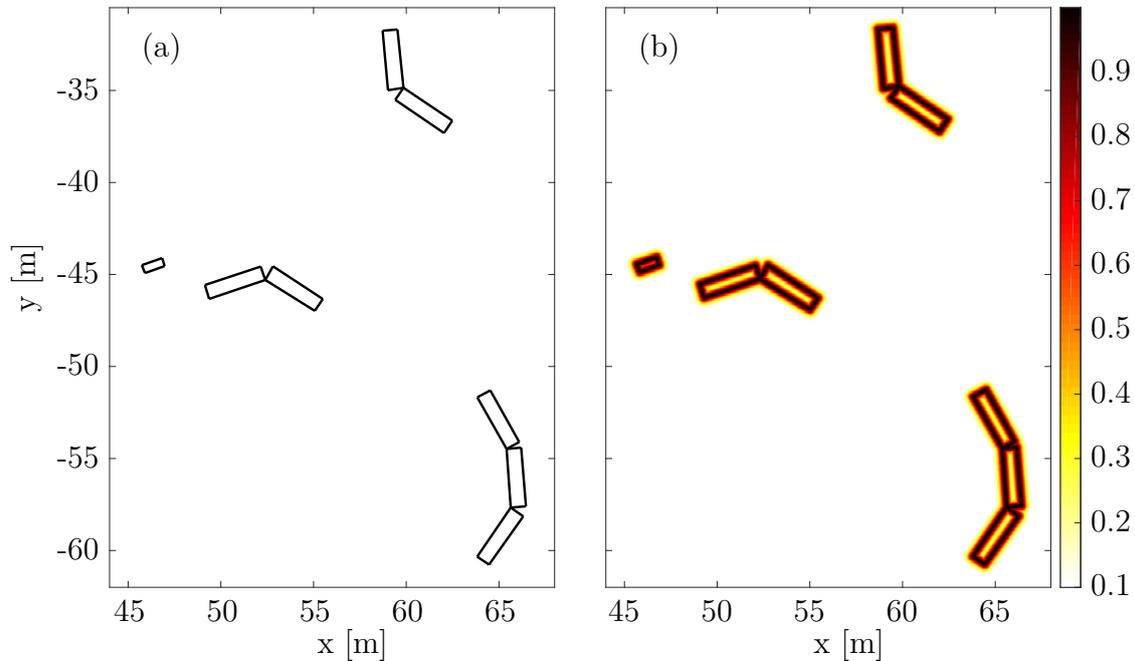


Abb. 4.3: Kartenaufbereitung für Scan-Matching. (a) kartierte Objekte, (b) diskretisiertes Wahrscheinlichkeitsfeld

Das für die Bestimmung der Wahrscheinlichkeiten zugrunde gelegte Sensormodell modelliert die Sensorfehler als normalverteiltes, weißes Rauschen mit bekannter Standardabweichung. Die Implementierung des Scan-Matching-Verfahrens basiert auf der betreuten Masterarbeit [6].

Nach der Ermittlung der Ausrichtungen aller Laserscans kann die Erzeugung der entsprechenden LiDAR-Kanten nach dem gleichen Prinzip wie bei den GNSS-Kanten erfolgen. Zunächst wird ermittelt, welcher Knoten die Pose repräsentiert, die zeitlich dem betrachteten Laserscan zugehörig ist. Anschließend wird dieser Knoten mit einem neu erzeugten LiDAR-Knoten über eine ebenfalls neu erzeugte Kante verbunden, wobei die Kante die Transformation $(\Delta x, \Delta y, \Delta \varphi) = (0, 0, 0)$ enthält und der Knoten mit der auf Basis des Laserscans ermittelten Pose initialisiert wird. Wie bei der Verarbeitung der GNSS-Signale wird auch hier der Knoten nicht als Variable dem Optimierungsproblem hinzugefügt, sondern als unveränderlich fixiert.

Mit der Verarbeitung aller Laserscans ist der Graph vollständig aufgebaut, sodass die finale

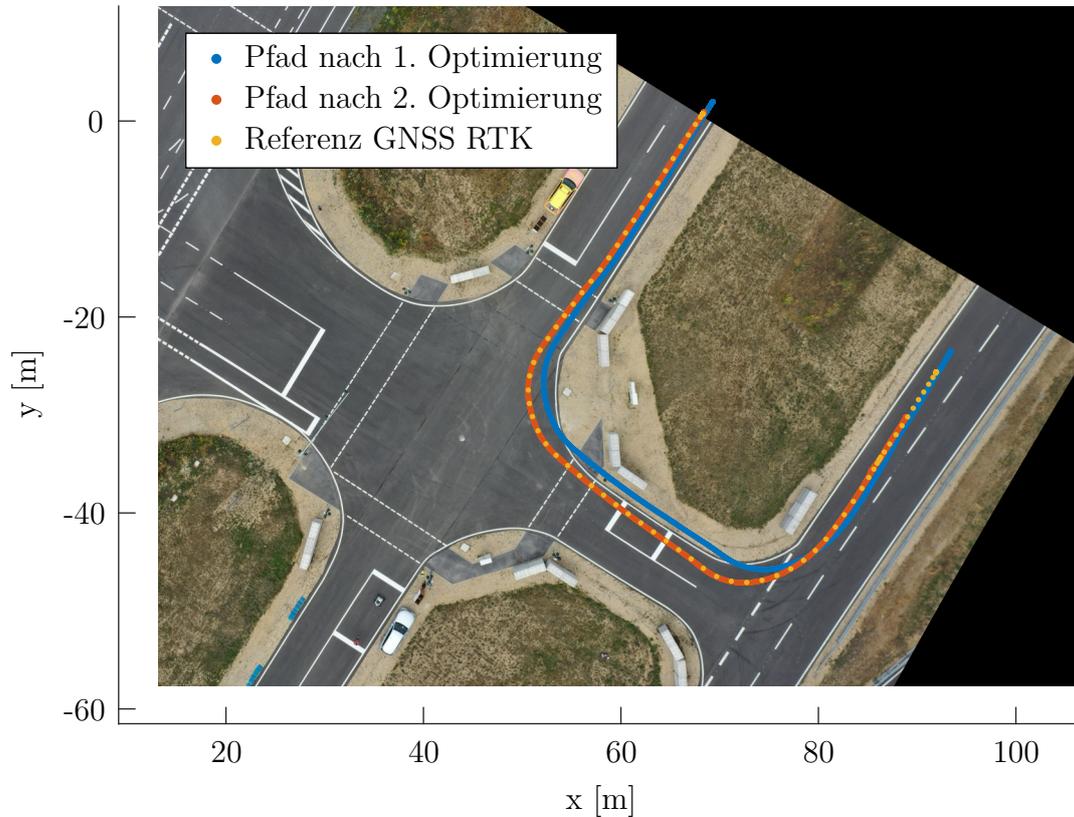


Abb. 4.4: Geschätzte Fahrzeugtrajektorie nach zweitem Optimierungsschritt

Optimierung erfolgen kann. Die als Ergebnis dieses Optimierungsschrittes berechnete Trajektorie ist am Beispiel des zu Abbildung 4.2 identischen Fahrzeugdatensatzes in Abbildung 4.4 dargestellt. Genutzt wurde hier der Scan-Matching-Algorithmus. Es ist erkennbar, dass die systematischen Fehler des GNSS-Empfängers durch den Abgleich mit der Referenzkarte vollständig ausgeglichen werden können. Die Referenzdaten des RTK-fähigen, hochgenauen GNSS-Empfängers zeigen eine sehr gute Übereinstimmung mit der durch das GraphSLAM-Verfahren optimierten Trajektorie. Zu berücksichtigen ist zudem, dass die optimierte Trajektorie im Gegensatz zu den Referenzdaten des RTK-Systems die Ausrichtung des Fahrzeuges φ beinhaltet und mit einer Auflösung von 100 Hz zur Verfügung steht. Im Rahmen der anschließenden Aktualisierung der Belegungskarte steht somit eine zuverlässige Datenbasis für die Transformation der Umgebungsdaten in das ortsfeste Kartenkoordinatensystem zur Verfügung.

Bei der Nutzung des ICP-Algorithmus können Fälle auftreten, in denen suboptimale Transformationen als LiDAR-Kanten in den Graphen aufgenommen werden. Der Grund hierfür liegt in der Anfälligkeit des Algorithmus für die Konvergenz gegen lokale Mini-

ma. Insbesondere in Fällen, in denen die Trajektorie nach dem ersten Optimierungsschritt aufgrund schwacher GNSS-Daten noch relativ weit vom Optimum entfernt ist, kann dieser Effekt zu einer signifikanten Verschlechterung der Trajektorienrekonstruktion führen. In Kombination mit dem ICP-Verfahren wird daher ein zusätzlicher Iterationsschritt vorgesehen, in dem zunächst neue LiDAR-Kanten auf Basis von ICP erzeugt werden und im Anschluss eine erneute Optimierung erfolgt. Gegenüber dem ersten Ausführen des ICP-Algorithmus steht beim zweiten Durchgang durch die zwischenzeitliche Optimierung im Allgemeinen bereits eine wesentlich bessere Trajektorie zur Verfügung. Die Gefahr der Konvergenz gegen lokale Minima und damit der Erzeugung ungenauer LiDAR-Kanten wird somit wesentlich reduziert.

Tabelle 4.3: Parameter des Algorithmus zur Rekonstruktion der Trajektorie

Symbol	Bezeichnung	Wert
σ_{Odo}	Standardabweichung Odometrie-Kanten für Translation im SLAM-Graph	0.002 m
σ_{IMU}	Standardabweichung IMU-Kanten für Rotation im SLAM-Graph	0.11°
σ_G	Standardabweichung GNSS-Kanten im SLAM-Graph	2 m
$\sigma_{L,xy}$	Standardabweichung LiDAR-Kanten für Translation im SLAM-Graph	0.2 m
$\sigma_{L,\varphi}$	Standardabweichung LiDAR-Kanten für Rotation im SLAM-Graph	2.8°
β_{Punkt}	Auflösung der aus der Linienkarte erzeugten Punktwolke (ICP und Scan-Matching)	0.05 m
$d_{Korr,max}$	Max. Distanz zwischen benachbarten Punkten für Korrespondenz (ICP)	5 m
$\beta_{xy,grob}$	Auflösung des groben Rasters (Scan-Matching)	0.5 m
$\beta_{xy,fein}$	Auflösung des feinen Rasters (Scan-Matching)	0.05 m
δ_φ	Auflösung der rotatorischen Transformation (Scan-Matching)	0.5°
r_φ	Suchradius bezüglich der Rotation (Scan-Matching)	5°
r_{xy}	Suchradius bezüglich der Translation (Scan-Matching)	8 m

Die wesentlichen Parameter der zur Trajektorienrekonstruktion eingesetzten Verfahren sind in Tabelle 4.3 zusammengefasst. Es ist erkennbar, dass das GraphSLAM-Verfahren selber mit sehr wenigen Parametern auskommt. Festzulegen sind lediglich die Gewichtungen der verschiedenen Kanten in Form von Standardabweichungen der zugrunde liegenden Messdaten. Aus diesen Informationen können dann die entsprechenden Kovarianzmatrizen generiert werden. Bei den Standardabweichungen von Odometrie-, IMU-, GNSS- und LiDAR-Kanten sind die absoluten Werte für das Optimierungsproblem nicht relevant. Die relativen Verhältnisse der verschiedenen Unsicherheiten sind die wesentliche Einflussmög-

lichkeit, um die Gewichtung der verschiedenen Informationsquellen in der Kostenfunktion zu adaptieren. Zu berücksichtigen ist zudem, dass die Zahlenwerte σ_{Odo} und σ_{IMU} sehr kleine Werte annehmen, da die entsprechenden Kanten zwischen aufeinanderfolgenden Knoten mit einer Rate von 100 Hz sehr kleine Distanzen überbrücken.

Abgesehen von den Kantengewichten beziehen sich alle weiteren Parameter auf die Verfahren zur Ausrichtung der Lasermessungen. Sowohl das ICP-, als auch das Scan-Matching-Verfahren benötigen die Referenzkarte in Form einer Punktwolke, die mit der Auflösung β_{Punkt} durch diskrete Abtastung aus der Linienkarte erzeugt wird. Für das ICP-Verfahren wird darüber hinaus die maximale Distanz definiert, oberhalb derer mögliche Punktkorrespondenzen verworfen werden. Da standardmäßige Werte aus der Softwarebibliothek [69] für die Konvergenzkriterien des ICP-Algorithmus genutzt werden, sind diese hier nicht explizit aufgeführt. Für das Scan-Matching-Verfahren müssen die Schrittweiten der Diskretisierungen des Suchraums in translatorischer und rotatorischer Richtung angegeben werden. Das feine Raster unterscheidet sich vom groben Raster um den Faktor 10, die erreichbare Auflösung des Verfahrens ergibt sich zu 0.05 m in translatorischer und 0.5° in rotatorischer Richtung. Durch die Vorgabe der rotatorischen und translatorischen Suchradien sind die Anforderungen an die Genauigkeit der initialen Ausrichtung vor dem Scan-Matching definiert.

4.4.2 Aktualisierung der Belegungskarte

Wie in Kapitel 4.3.4 dargestellt, besteht das Konzept zur kontinuierlichen Aktualisierung der digitalen Karte darin, eine gitterbasierte Belegungskarte zu definieren, deren Belegungswahrscheinlichkeiten bei Verfügbarkeit neuer Messdaten jeweils angepasst werden. In Kapitel 2.2.2 wurde der grundlegende Aktualisierungsmechanismus unter Verwendung logarithmischer Wahrscheinlichkeiten eingeführt. Für die vorliegende Anwendung wird dieser Algorithmus um ein für den Sensor spezifisches Messmodell erweitert, bei dem die Information jedes einzelnen ausgesendeten Laserstrahls bestmöglich ausgenutzt werden kann. Es ergibt sich die als Pseudocode in Algorithmus 4 dargestellte Struktur. Der gezeigte Algorithmus arbeitet alle über die diskretisierten Zeitpunkte $t = 1 : T$ aufgezeichneten Messdaten chronologisch ab, wobei zu jedem Zeitpunkt t alle vom Sensor ausgesendeten Laserstrahlen bezüglich der entsprechenden Pose \vec{x}_t des Sensors auf das Gitter der Karte projiziert werden. Zur Ermittlung der vom Laserstrahl überstrichenen Kartenzellen wird vereinfachend davon ausgegangen, dass die Aufweitung jedes Strahls der Winkelauflösung des Sensors um die Hochachse entspricht, sodass eine vollständige Abdeckung des Sichtfeldes erzielt wird.

Bei der Aktualisierung der Zellengewichte gemäß Gleichung 2.12 (siehe Kapitel 2.2.2) unterscheidet der Algorithmus drei Fälle: Falls der LiDAR-Sensor bei einem betrachteten Laserstrahl kein Echo detektiert hat, kann abgeleitet werden, dass innerhalb der Reichweite des Sensors keine Hindernisse liegen. Somit wird für die innerhalb der Reichweite

Algorithmus 4 Occupancy Grid Mapping (Karte M , Pose $\vec{x}_{1:T}$, Laserscan $\vec{z}_{1:T}$)

```

1: for Zeitpunkt  $t = 1 : T$  do
2:   for Laserstrahl  $n = 1 : N$  ausgehend von Pose  $\vec{x}_t$  do
3:     for vom Laserstrahl  $n$  überstrichene Kartenzelle  $m = 1 : M$  do
4:       if kein Echo vom Laserstrahl  $n$  then
5:         Aktualisiere Zelle  $M_m$  mit  $p_{frei,Reichw}$ 
6:       else
7:         if Distanz zur aktuellen Zelle kleiner als Distanz aus Laserecho then
8:           Aktualisiere Zelle  $M_m$  mit  $p_{frei}$ 
9:         else if Distanz zur aktuellen Zelle gleich Distanz aus Laserecho then
10:          Aktualisiere Zelle  $M_m$  mit  $p_{belegt}$ 
11:        else if Distanz zur aktuellen Zelle größer als Distanz aus Laserecho then
12:          Gehe über zu nächstem Laserstrahl
13:        end if
14:      end if
15:    end for
16:  end for
17: end for

```

liegenden, vom Laserstrahl überstrichenen Gitterzellen die Belegungswahrscheinlichkeit verringert. Zu berücksichtigen ist in diesem Fall, dass die Reichweite des Sensors je nach Umgebungsbedingungen sowie je nach Reflektivität der Hindernisse variiert, sodass Zellen nur innerhalb einer sicheren, minimalen Reichweite $d_{L,min}$ des Sensors verändert werden. Die Gewichtung der Information aus fehlenden Laserechos wird zudem umgekehrt linear abhängig von der Distanz zur Sensorposition definiert. Diese nur sehr begrenzte Auswertung der Information über freie Kartenbereiche ist sinnvoll, da fälschlicherweise nicht detektierte Echos nicht zu einem vorschnellen „Überschreiben“ von zuvor als „belegt“ erkannten Zellen der Karte führen sollen.

Im Vergleich deutlich höher gewichtet wird die Information eines Laserstrahls, bei dem ein Echo detektiert wurde und somit eine Entfernungsmessung vorliegt. Für alle überstrichenen Zellen, die zwischen dem Sensor und dem detektierten Echo liegen, wird die Belegungswahrscheinlichkeit verringert. Die dem Echo zugeordneten Gitterzellen erfahren eine Erhöhung der Belegungswahrscheinlichkeit.

Abbildung 4.5 visualisiert die verschiedenen Update-Mechanismen anhand einer Prinzipskizze. Dem äquidistanten Gitter sind die Position des LiDAR-Sensors sowie die vom Sensor empfangenen Echos überlagert. Die entsprechend der drei unterschiedenen Fälle vorgenommenen Aktualisierungen sind farblich hervorgehoben. Über die Parameter p_{frei} , $p_{frei,Reichw}$ und p_{belegt} kann die Gewichtung der entsprechenden Informationen in Form von Wahrscheinlichkeitswerten adaptiert werden.

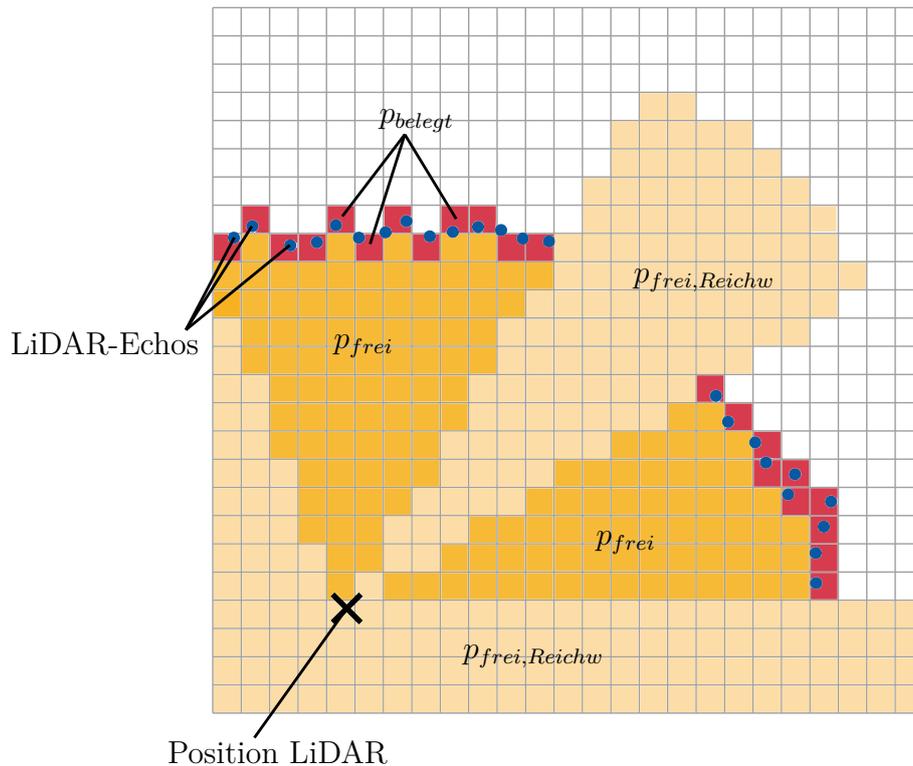


Abb. 4.5: Anpassungen verschiedener Bereiche beim Occupancy Grid Mapping

Um zu vermeiden, dass bei stehendem Fahrzeug die über einen längeren Zeitraum aus der selben Perspektive aufgezeichneten Umgebungsinformationen des Laserscanners zu einer übermäßig starken Änderung der Belegungskarte führen, wird eine zusätzliche Regel implementiert: Ein neuer Laserscan wird immer nur dann verarbeitet, wenn das Fahrzeug seit dem letzten verarbeiteten Laserscan eine minimale Distanz zurückgelegt hat. Diese Distanz wird so parametrisiert, dass ab einer Fahrt mit Schrittgeschwindigkeit (etwa 7 km h^{-1}) jeder mit einer Rate von 20 Hz empfangene Scan verarbeitet wird. Bei langsamerer Geschwindigkeit oder bei Stillstand des Fahrzeuges werden dementsprechend Scans ausgelassen.

Ein weiterer unerwünschter Effekt kann auftreten, wenn für eine Vielzahl von aufeinanderfolgenden Fahrzeugen über einen längeren Zeitraum übereinstimmende Informationen zum Belegungszustand einer Zelle verarbeitet werden, sodass sich die Belegungswahrscheinlichkeit der Zelle sehr nahe an 0 bzw. 1 annähert. Kommt es anschließend zu einer Änderung im realen Verkehrsumfeld – etwa durch das Ausfahren eines geparkten Fahrzeuges – so werden ebenfalls sehr viele Beobachtungen des neuen, geänderten Belegungszustandes benötigt, bis sich die Belegungswahrscheinlichkeit entsprechend anpassen

kann. Da diese potenziell sehr lange Reaktionszeit bei veränderlichen Verkehrsumgebungen unerwünscht ist, werden die Belegungswahrscheinlichkeiten p_m der gitterbasierten Belegungskarte auf ein Intervall $[p_{min} < p_m < p_{max}]$ mit $p_{min} > 0$ und $p_{max} < 1$ begrenzt. Diese Beschränkungen können als Sättigung interpretiert werden: Ab der definierten Grenze führen weitere, übereinstimmende Beobachtungen zu keiner weiteren Anpassung der Belegungswahrscheinlichkeit.

Für jeden einzelnen, von einem durchfahrenden Fahrzeug bereitgestellten Datensatz wird ebenfalls ein maximaler Einfluss Δl_{max} auf die Änderung des Wahrscheinlichkeitswertes jeder Kartenzelle festgelegt. Auf diese Weise wird verhindert, dass von einem einzelnen Fahrzeug mit möglicherweise defekter oder schlecht kalibrierter Sensorik eine übermäßige Degradierung der Karte ausgehen kann. Die definierte Begrenzung der Änderungen der Belegungswahrscheinlichkeiten stellt sicher, dass durch die inhärente statistische Mittelung des Aktualisierungsverfahrens einzelne Ungenauigkeiten ausgeglichen werden können.

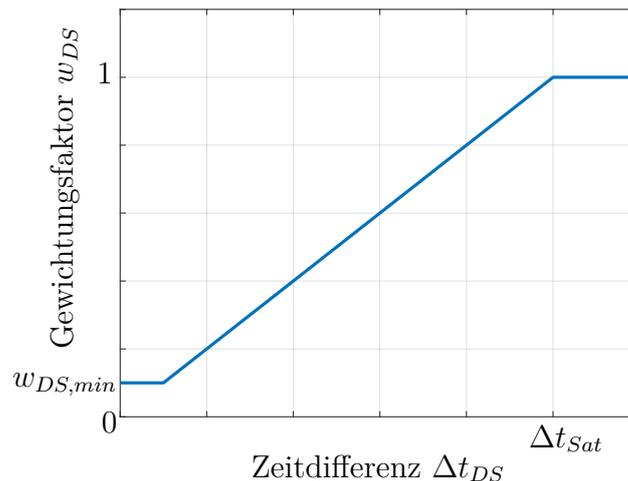


Abb. 4.6: Gewichtungsfaktor eines Datensatzes in Abhängigkeit der Zeitdifferenz zum letzten Datensatz

Eine weitere wichtige Umgebungsbedingung, welche das Verhalten des Aktualisierungsalgorithmus beeinflusst, ist die Verkehrsdichte im kartierten Gebiet. Durchfahren viele vernetzte Fahrzeuge den betrachteten Bereich, so fallen in kurzer Zeit deutlich mehr Sensordaten an, als bei einem geringen Verkehrsaufkommen. Damit ergibt sich in diesen Fällen auch potenziell eine deutlich schnellere Dynamik bzw. eine kürzere Reaktionszeit der digitalen Karte auf Änderungen im Verkehrsumfeld. Im Gegenzug kann es vorkommen, dass in Zeiten mit geringem Verkehrsaufkommen die Aktualisierung der Karte nach einer Änderung des Verkehrsumfeldes sehr lange dauert, da nur wenige Datensätze von durchfahrenden Fahrzeugen bereitgestellt werden. Bei der Parametrierung des Algorithmus

ergibt sich somit ein Zielkonflikt zwischen der Verbesserung von Genauigkeit und Robustheit der Kartenupdates durch die statistische Mittelung der empfangenen Datensätze und einer kurzen Reaktionszeit der Karte auf geänderte Umgebungsgeometrien. Ersteres erfordert eine kleine Gewichtung einzelner Fahrzeugdatensätze, letzteres lässt sich durch höhere Gewichte erzielen. Zur Auflösung des beschriebenen Zielkonfliktes wird daher ein von der Zeit abhängiger Gewichtungsfaktor eingeführt. Wie in Abbildung 4.6 dargestellt, wird zur Berechnung des Faktors w_{DS} die Zeitdifferenz Δt_{DS} zur vorherigen Änderung der Karte herangezogen und ein im Wesentlichen linearer Zusammenhang definiert. Ein hohes Verkehrsaufkommen bringt kurze Zeitdifferenzen und somit geringere individuelle Gewichte mit sich, während bei niedrigem Verkehrsaufkommen jeder Datensatz höher gewichtet wird. Mit Δt_{sat} kann die Zeitdifferenz parametrisiert werden, oberhalb derer das individuelle Gewicht eines Datensatzes nicht weiter erhöht wird; $w_{DS,min}$ bezeichnet den minimalen Einfluss, den jeder Datensatz selbst bei sehr großem Verkehrsaufkommen haben soll. Die Sensordaten eines Fahrzeugdatensatzes decken im Allgemeinen nicht alle Bereiche des kartierten Areals ab, da das Fahrzeug in der Regel nur einen Teil des Areals befährt. Die für die Bestimmung des Gewichtungsfaktors benötigten Zeitdifferenzen werden daher für jede Gitterzelle der Karte individuell berechnet.

Die vorgenannten Mechanismen bringen einige Parameter mit sich, die für die optimale Funktionalität des Gesamtsystems konsistent parametrisiert werden müssen. Um diesen Parametersatz entsprechend den Besonderheiten verschiedener Verkehrsumfelder möglichst intuitiv und systematisch anpassen zu können, werden abstrakte, übergeordnete Parameter definiert, aus denen sich die unterlagerten Grenzwerte und Wahrscheinlichkeiten automatisch berechnen lassen. Eine wesentliche Abwägung betrifft dabei die zeitliche Dynamik der Kartenänderungen, also die Geschwindigkeit, mit der die Karte auf durch die Fahrzeuge detektierte Änderungen reagiert. Als abstrahierter Parameter kann dem Algorithmus hierfür die minimale Anzahl an durchfahrenden Fahrzeugen vorgegeben werden, die benötigt werden, um ausgehend von einer Gitterzelle mit der minimalen Belegungswahrscheinlichkeit p_{min} den Grenzwert für die Ausführung der Linienextraktion p_{Lin} , also der Betrachtung der Zelle als „belegt“, zu überschreiten. Die gleiche minimale Fahrzeuganzahl wird sodann auch für die Änderung der Belegungswahrscheinlichkeit von p_{max} auf einen Wert unter p_{Lin} gefordert. In ähnlicher Weise können die benötigten Fahrzeuganzahlen für die Detektion eines zu kartierenden Objektes ausgehend von der initialen Belegungswahrscheinlichkeit p_0 , also von einer Zelle ohne bekannte Vorinformation, festgelegt werden. Bei bekannten Grenzen für p_{min} und p_{max} können somit automatisch der maximale Einfluss eines Fahrzeuges Δl_{max} als logarithmische Wahrscheinlichkeitsdifferenz, die initiale Belegungswahrscheinlichkeit p_0 sowie der Grenzwert für die Linienextraktion p_{Lin} ermittelt werden.

Gleichungen 4.4 bis 4.6 zeigen die für die Parametrierung verwendeten Zusammenhänge. Aufgrund der logarithmischen Formulierung werden die Wahrscheinlichkeitsänderungen durch einfache Additionen bzw. Subtraktionen realisiert. Die Umrechnung in Wahrscheinlichkeitswerte erfolgt entsprechend der in Kapitel 2.2.2 eingeführten Grundgleichungen.

$$l_{Lin} = 0.5 \cdot l_{min} + 0.5 \cdot l_{max} \quad (4.4)$$

$$\Delta l_{max} = \frac{l_{max} - l_{min}}{2 \cdot n_{Fzg,Zust}} \quad (4.5)$$

$$l_0 = \frac{l_{max} \cdot (n_{Fzg,Zust} - n_{Fzg,Init}) + l_{min} \cdot (n_{Fzg,Zust} + n_{Fzg,Init})}{2 \cdot n_{Fzg,Zust}} \quad (4.6)$$

Für die Parametrierung der Wahrscheinlichkeitswerte p_{frei} , $p_{frei,Reichw}$ und p_{belegt} wird eine standardisierte Testfahrt zugrunde gelegt, bei der sich für definierte, geradlinige Hindernisse am Straßenrand eine Wahrscheinlichkeitsänderung von $+\Delta l_{max}$ und für die davor liegenden freien Bereiche eine Wahrscheinlichkeitsänderung von $-\Delta l_{max}$ ergeben soll. In diesem definierten Standardfall soll ein durchfahrendes Fahrzeug also den ihm zur Änderung der Karte zur Verfügung gestellten Rahmen ausschöpfen. Der Einfluss von $p_{frei,Reichw}$ kann schlussendlich im festen Verhältnis zu p_{frei} definiert werden.

Tabelle 4.4: Parameter des Kartieralgorithmus

Symbol	Bezeichnung	Wert
d_{Zelle}	Kantenlänge der quadratischen Gitterzellen	0.05 m
$n_{Fzg,Zust}$	Min. Anzahl Fahrzeuge für Zustandsänderung	2.2
$n_{Fzg,Init}$	Min. Anzahl Fahrzeuge für Initialisierung	1.1
p_0	Initiale Belegungswahrscheinlichkeit	0.09133
l_0	Initiale logarithmische Belegungswahrscheinlichkeit	-2.298
p_{min}	Minimale Belegungswahrscheinlichkeit	0.01
l_{min}	Minimale logarithmische Belegungswahrscheinlichkeit	-4.595
p_{max}	Maximale Belegungswahrscheinlichkeit	0.99
l_{max}	Maximale logarithmische Belegungswahrscheinlichkeit	4.595
p_{Lin}	Grenzwert Belegungswahrscheinlichkeit für Linienextraktion	0.5
l_{Lin}	Grenzwert logarithmische Belegungswahrscheinlichkeit für Linienextraktion	0
Δl_{max}	Maximale logarithmische Wahrscheinlichkeitsänderung	2.09
p_{belegt}	Einzelwahrscheinlichkeit bei als „belegt“ beobachteter Zelle	0.1013
p_{frei}	Einzelwahrscheinlichkeit bei als „frei“ beobachteter Zelle	0.08882
$p_{frei,Reichw}$	Einzelwahrscheinlichkeit bei Zelle in Sensorreichweite ohne Beobachtung	0.09063
$d_{L,min}$	Minimale Reichweite LiDAR für Aktualisierung mit $p_{frei,Reichw}$	15 m
Δt_{Sat}	Zeitdifferenz für maximale Gewichtung eines Datensatzes	30 s
$w_{DS,min}$	Minimaler Gewichtungsfaktor eines Datensatzes	0.1

Tabelle 4.4 zeigt eine Übersicht der für die Testumgebung vorgenommenen Parametrierung. Dargestellt sind sowohl die abstrahierten Parameter, als auch die sich ergebenden

Wahrscheinlichkeiten bzw. logarithmischen Differenzen. Die gewählte Parametrierung ist ausgelegt auf ein kartiertes Areal mit geringem Verkehrsaufkommen. $n_{Fzg,Zust}$ wurde so gewählt, dass – bei maximaler Gewichtung auf Basis der Zeitdifferenz Δt_{DS} – bereits nach dem dritten Fahrzeug eine Zustandsänderung von „frei“ auf „belegt“ und umgekehrt erfolgen kann. Mit $n_{Fzg,Init} = 1.1$ erfordert die Initialisierung eines neuen geometrischen Elementes ausgehend von der initialen Belegungswahrscheinlichkeit p_0 nur zwei übereinstimmende Fahrzeugbeobachtungen. Bei der Festlegung von p_{frei} , $p_{frei,Reichw}$ und p_{belegt} auf Basis der standardisierten Testfahrt wird deutlich, dass aufgrund der Vielzahl an Einzelbeobachtungen, die jede Testfahrt enthält, nur relativ geringe Abweichungen von p_0 parametrisiert werden, um die gewünschte Gewichtung zu erzielen.

Ein beispielhaftes Simulationsergebnis zur Wirkung des probabilistischen Ansatzes ist in Abbildung 4.7 dargestellt. Die Belegungswahrscheinlichkeiten p der einzelnen Gitterzellen sind über eine farbliche Abstufung sichtbar gemacht, deren Maximum bei $p = 0.45$ liegt. Auf diese Weise kann der gesamte Wahrscheinlichkeitsbereich, der aufgrund der definierten Grenzen nach einer Durchfahrt erreicht werden kann, abgedeckt werden.

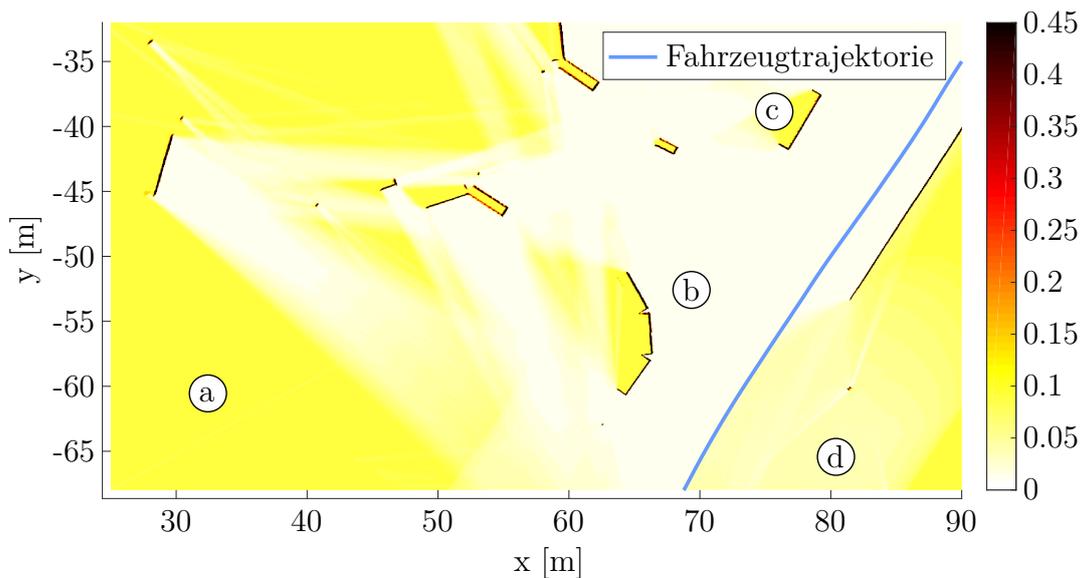


Abb. 4.7: Simulationsergebnis Belegungskarte nach Durchfahrt eines Fahrzeuges

In der Graphik werden deutlich die verschiedenen Aktualisierungsmechanismen mit den entsprechenden Auswirkungen auf die Gitterkarte sichtbar. Zunächst ist festzustellen, dass statische Objekte in Form von scharf abgegrenzten Linienzügen mit hohen Belegungswahrscheinlichkeiten erfasst werden. Sie dienen als Grundlage für die anschließende Linienextraktion. Der Bereich (a) erfährt gegenüber der initialen Belegungswahrscheinlichkeit p_0 keine Änderung. Die entsprechenden Zellen liegen außerhalb der für die Anwendung von $p_{frei,Reichw}$ definierten Messdistanz und es sind keine Objekte vorhanden,

die Laserechos erzeugen. Der Bereich (b) liegt zwischen der vom Fahrzeug befahrenen Trajektorie und einer großflächigen Wand. Daher werden hier Aktualisierungen mit p_{frei} vorgenommen, die dazu führen, dass die entsprechenden Belegungswahrscheinlichkeiten deutlich reduziert werden. Im Bereich (c), der sich aus Sicht des LiDAR-Sensors hinter einer Wand befindet, werden aufgrund der Verdeckung der Laserstrahlen keine Aktualisierungen der Zellenwahrscheinlichkeiten vorgenommen. Die Wahrscheinlichkeit bleibt unverändert bei p_0 . Im Bereich (d) ist schließlich der Einfluss von $p_{frei,Reichw}$ erkennbar. Zwar werden hier keine Laserechos empfangen, der Bereich liegt aber nahe an der befahrenen Trajektorie, sodass mit gegenüber Bereich (b) geringerer Gewichtung von einem nicht belegten Areal ausgegangen werden kann.

4.4.3 Linienextraktion

Die zur kontinuierlichen Aktualisierung der Umgebungsinformationen genutzte gitterbasierte Belegungskarte entspricht nicht dem in Kapitel 4.3.2 definierten Format für die Lokalisierung automatisierter Fahrzeuge. Die benötigte Linienkarte, in der die geometrischen Elemente durch diskrete Linienzüge abstrahiert werden, muss somit in einem zusätzlichen Verarbeitungsschritt erzeugt werden. Basierend auf Ergebnissen der betreuten Masterarbeit [6] werden im folgenden die genutzten Algorithmen für die Linienextraktion eingeführt. Das Verfahren umfasst 4 Teilschritte, die sequenziell durchlaufen werden.

Im Zuge der **Clusteranalyse** werden zunächst solche Zellen zusammengefasst, die ein abgegrenztes Objekt innerhalb der Belegungskarte beschreiben. Hierfür wird die Belegungskarte binarisiert, wobei der im vorangegangenen Kapitel 4.4.2 eingeführte Parameter p_{Lin} den Schwellwert für die Belegungswahrscheinlichkeit definiert, ab der eine Zelle als belegt übernommen wird. Ausgehend von einer Startposition innerhalb der binären Belegungsmatrix wird anschließend eine Tiefensuche durchgeführt, bei der alle unmittelbar aneinander angrenzenden Zellen einem Cluster hinzugefügt werden. Als angrenzend werden hierbei sowohl in Zeilen- und Spaltenrichtung, als auch diagonal benachbarte Zellen betrachtet. Im Zuge der Clusteranalyse werden alle belegten Zellen aussortiert, die sich innerhalb eines definierten Abstandes $d_{Lin,min}$ zu bereits bekannten Linien der statischen Referenzkarte befinden, um die Erzeugung redundanter Linienobjekte zu verhindern.

Um beim späteren Einpassen der Liniensegmente in die ermittelten Cluster die ursprüngliche, vom Laserscanner gemessene Objektkante möglichst genau abzubilden, wird eine **Gewichtung der Zellen** innerhalb der Cluster vorgenommen. Abbildung 4.8 visualisiert den Effekt, der diesen Prozessschritt notwendig macht. Dargestellt ist, welche logarithmischen Wahrscheinlichkeiten sich für verschiedene Gitterzellen entlang eines Laserstrahls ergeben, wenn der Sensor ein festes Objekt im Abstand d_0 beobachtet. Es wird dabei angenommen, dass die Distanzmessungen des Laserscanners mit einem normal verteilten Rauschen mit Standardabweichung $\sigma_{Las} = 0.1$ m überlagert sind. Die einzelnen Säulen des Diagramms entsprechen den diskreten Gitterzellen entlang des Laserstrahls,

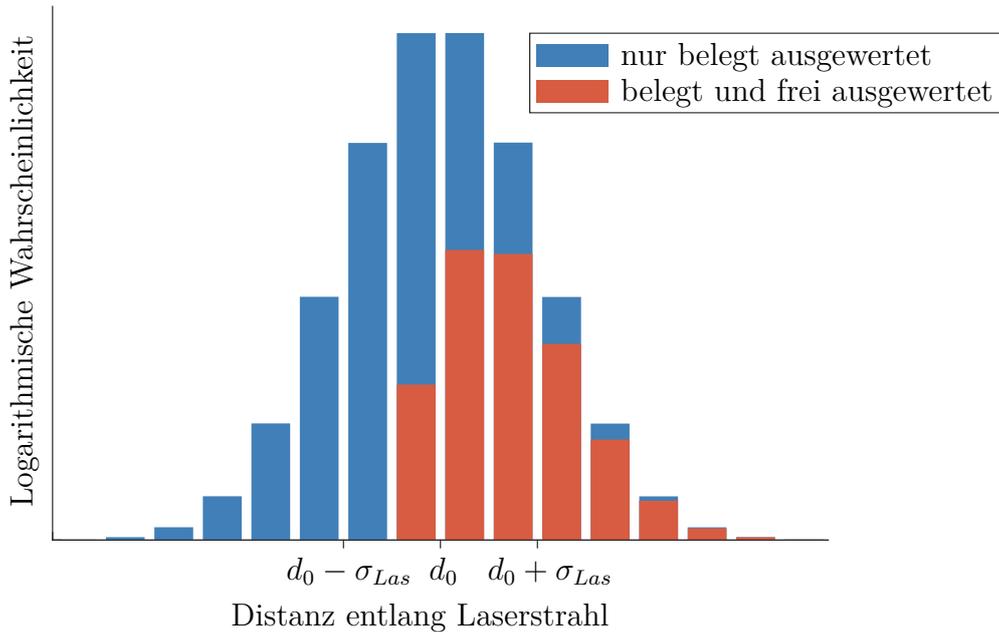


Abb. 4.8: Logarithmische Wahrscheinlichkeiten der Gitterzellen entlang eines Laserstrahls bei Überlagerung der Entfernungsmessung mit normalverteiltem Rauschen

wobei die Zellenbreite $d_{Zelle} = 0.5$ m der in Kapitel 4.4.2 definierten Gitterkarte zugrunde gelegt wird. In Blau ist der Fall dargestellt, dass die Gewichte der Zellen lediglich erhöht werden, wenn die Sensormessung ein Objekt auf der entsprechenden Zelle detektiert. Die Information, dass zugleich alle Zellen zwischen dem Sensor und dem erfassten Objekt als frei anzusehen sind, wird hier ignoriert. Es ist erkennbar, dass sich in diesem Fall erwartungsgemäß eine diskrete Normalverteilung mit Mittelwert d_0 und Standardabweichung σ_{Las} ergibt. Die optimale Lage der Objektkante könnte hier also einfach aus dem Mittelwert der Normalverteilung rekonstruiert werden. Eine abweichende Darstellung ergibt sich, wenn – wie bei der regulären Aktualisierung der Karte – auch die Verringerung der Belegungswahrscheinlichkeit der vom Laserstrahl überstrichenen Zellen einbezogen wird. Da dieser Mechanismus lediglich die in Blickrichtung des Sensors vor dem Objekt liegenden Zellen beeinflusst, entsteht die in Rot dargestellte verschobene Verteilung, deren Mittelwert nicht mehr der wahren Distanz entspricht. Das Ausmaß der Verschiebung hängt dabei sowohl von σ_{Las} , als auch von p_{frei} und p_{belegt} (siehe Tabelle 4.4) ab.

Da der beschriebene Effekt eine Verschiebung der Verteilung als „belegt“ markierter Gitterzellen in Blickrichtung des LiDAR-Sensors hervorruft, wird für die Kompensation die mittlere Blickrichtung des Sensors auf jede belegte Gitterzelle ermittelt. Dieser Prozess erfolgt bereits im Rahmen der Aktualisierung der Belegungskarte durch eine Mittelwertbildung über die Strahlrichtungen aller für eine Zelle relevanten Lasermessungen.

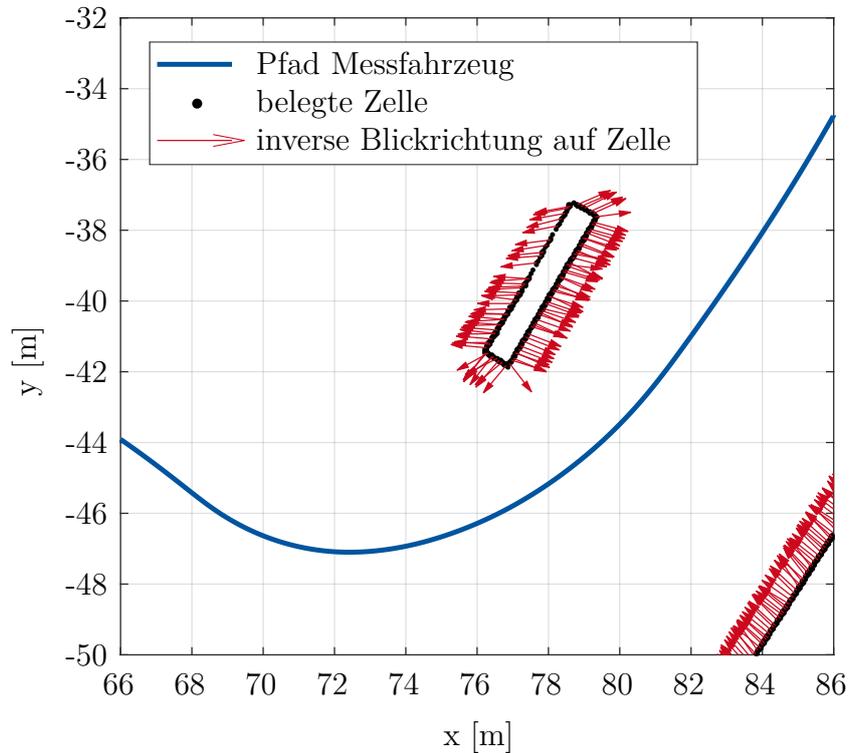


Abb. 4.9: Mittlere Strahlrichtungen auf belegte Gitterzellen bei Vorbeifahrt mit Messfahrzeug

Abbildung 4.9 zeigt beispielhaft die sich ergebenden Blickrichtungen auf Basis einer Vorbeifahrt des Messfahrzeuges an stationären Hindernissen. Aus Gründen der besseren Erkennbarkeit zeigen die dargestellten Pfeile die inversen Blickrichtungen.

Anhand der bekannten Blickrichtungen kann nun die Gewichtung der Zellen vorgenommen werden, wobei ein hohes Gewicht zu einer verstärkten Berücksichtigung einer Zelle bei der Bestimmung der Lage des Geradensegmentes führt. Aus Abbildung 4.8 kann entnommen werden, dass bei der vorliegenden Parametrierung der Gradient der Belegungswahrscheinlichkeit in Blickrichtung des Laserscanners ein wirksames Maß für die Lage der Objektkante darstellt. Daher wird das Gewicht w_Z jeder Zelle auf Basis der Differenz der Belegungswahrscheinlichkeit Δp_{Blick} zur relevanten Nachbarzelle vorgenommen. Als relevante Nachbarzelle wird dabei stets die entgegen der Strahlrichtung angrenzende Zelle gewählt, sodass ein großes Δp_{Blick} mit einer hohen Gewichtung korreliert. Der lineare Zusammenhang zwischen dem Gewicht w_Z und Δp_{Blick} wird so aufgestellt, dass sich bei maximalem Δp_{Blick} das Gewicht $w_Z = 1$ und bei minimalem Δp_{Blick} das Gewicht $w_Z = 0$ ergibt. Unter Berücksichtigung der Tatsache, dass nur Zellen mit einer Belegungswahrscheinlichkeit $p > p_{Lin}$ bei der Geradenextraktion berücksichtigt

werden, ergibt sich der Zusammenhang gemäß Formel 4.7 [6].

$$w_Z = \frac{\Delta p_{Blick} + 1 - p_{Lin}}{2 - p_{Lin}} \quad (4.7)$$

Im Anschluss an die Gewichtung der Gitterzellen erfolgt die **Liniensuche** mittels eines RANSAC-Algorithmus. RANSAC wurde ursprünglich in [28] eingeführt und beschreibt ein robustes Verfahren zum Schätzen von Modellparametern für die optimale Anpassung des Modells an einen Messdatensatz. Das Grundprinzip besteht darin, zufällig Punkte auszuwählen, auf deren Basis testweise die Modellparameter berechnet werden. Anschließend wird geprüft, wie viele weitere Punkte des Datensatzes mit dem so generierten Modell im Rahmen einer definierten Toleranz übereinstimmen. Ein Grenzwert für die Anzahl an Übereinstimmungen – den sogenannten „Inliern“ – legt fest, ob das Modell akzeptiert oder verworfen wird. Der Prozess wird iterativ wiederholt, wobei Punkte, die einem akzeptierten Modell als zugehörig definiert wurden, üblicherweise in den folgenden Iterationsschleifen ignoriert werden, sodass sich der durchsuchte Datensatz kontinuierlich verkleinert.

Im vorliegenden Fall ist das Modell durch eine Geradengleichung gegeben, die auf Basis von zwei Punkten berechnet werden kann. Bei der Ermittlung der Anzahl an Inliern wird die zuvor bestimmte Gewichtung berücksichtigt, sodass solche Punkte, die wahrscheinlich im Bereich der realen Objektkante liegen, einen höheren Einfluss auf die Akzeptanz des aktuell getesteten Modells haben. Wird eine Gerade akzeptiert, erfolgt zudem eine weitere iterative Optimierung, bei der die Parameter der Geraden so angepasst werden, dass diese bestmöglich zu allen zugeordneten Inliern passt. Anschließend werden auf Basis der angepassten Geraden die neuen Inlier bestimmt und die Optimierung wiederholt, bis ein Konvergenzkriterium erfüllt ist. Als Fehlermaß wird die gewichtete quadratische Summe der orthogonalen Abstände zwischen Inlier-Punkten und der Geraden genutzt. Die Liniensuche erfolgt stets nur innerhalb eines Clusters.

Der beschriebene Prozess ist in Abbildung 4.10 anhand eines Beispielobjektes visualisiert. Die schwarzen Pixel entsprechen den Zellen der Belegungskarte, die den Schwellwert für die Liniensuche überschreiten. In (a) sind die ersten beiden zufällig ausgewählten Punkte innerhalb des Clusters gelb markiert. Die daraus berechnete Gerade ist in grüner Farbe eingezeichnet. Die Inlier, die innerhalb des Toleranzbandes liegen und als der Gerade zugehörig erkannt werden, sind in (b) in blauer Farbe markiert, wobei ein dunklerer Farbton eine höhere Gewichtung impliziert. Als Ergebnis der iterativen Optimierung der Geraden ergibt sich die in (c) dargestellte, finale Gerade, die vorläufig übernommen wird. In (d) wird auf den verbleibenden Pixeln des Clusters eine neue Gerade aus zwei zufälligen Punkten generiert, die aufgrund der zu geringen Zahl an Inliern – erkennbar in (e) – verworfen wird. Die beiden nächsten Zufallspunkte in (f) führen schließlich zu einer passenden Geradengleichung in (g), die ebenfalls nach dem Durchlaufen der iterativen Optimierung in ihrer endgültigen Lage in (h) vorläufig übernommen wird. Es

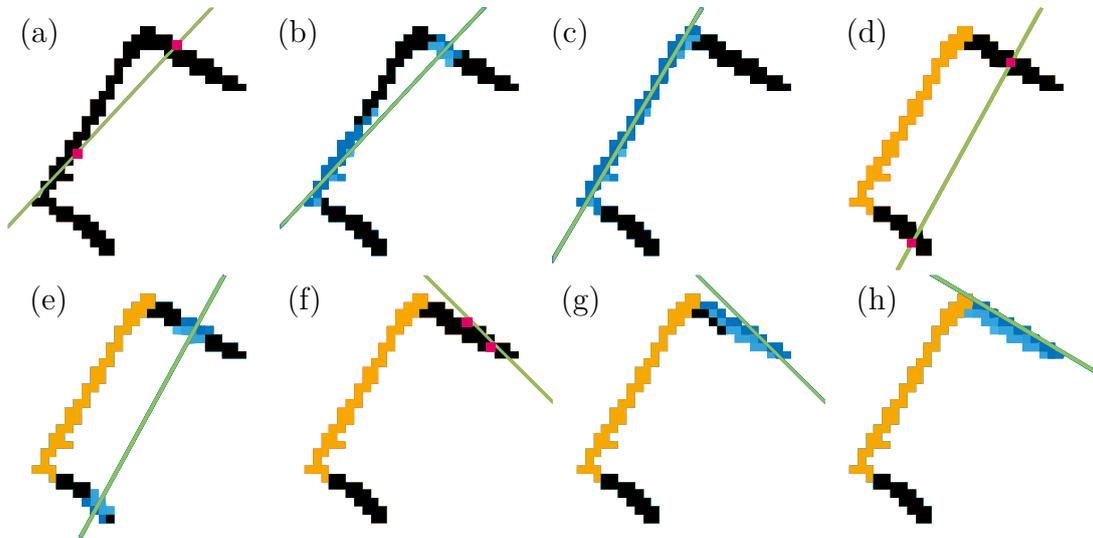


Abb. 4.10: Ablauf der Linienextraktion am Beispiel eines kastenförmigen Clusters

ist erkennbar, dass die Gerade aufgrund der blickrichtungsabhängigen Gewichtung der Zellen auf der Außenseite des rechteckigen Objektes liegt.

Tabelle 4.5: Parameter des Algorithmus zur Linienextraktion

Symbol	Bezeichnung	Wert
$d_{Lin,min}$	Min. Abstand neuer Cluster zu Linien der statischen Karte	0.6 m
$v_{Inl,min}$	Min. gewichtete Summe der Inlier für Akzeptanz der Geraden	12
$d_{Inl,max}$	Max. Distanz zur Linie für Zählung als Inlier	0.25 m
$v_{Clu,min}$	Min. gewichtete Summe der Zellen eines Clusters für Fortführung der Liniensuche	12
$n_{It,max}$	Max. Anzahl an Iterationen der Liniensuche auf einem Cluster	200
$d_{Lücke,max}$	Max. Länge einer Lücke innerhalb einer Linie	0.16 m

Vor der Übernahme der gefundenen Geraden erfolgt als finaler Verarbeitungsschritt eine **Plausibilisierung**, bei der ein möglicher Fehlerfall bei der Linienextraktion mittels RANSAC ausgeschlossen wird. Überbrückt eine Gerade eine größere Lücke, in der keine Inlier liegen, so wird sie potenziell trotzdem vom RANSAC-Algorithmus akzeptiert, da dieser lediglich die gewichtete Summe der Anzahl an Inliern als Akzeptanzkriterium nutzt, nicht jedoch deren Verteilung über die Gerade. Auf diese Weise können Linien extrahiert werden, die sich über eigentlich nicht belegte Bereiche der Belegungskarte erstrecken. Der Plausibilisierungsmechanismus sucht daher für jede akzeptierte Gerade den maximalen Abstand zwischen benachbarten Punkten auf der Geraden. Überschreitet dieser einen

Schwellwert, so wird die Gerade verworfen. Alle übrigen Geraden gelten als bestätigt und werden mit ihren Anfangs- und Endpunkten in die Datenbasis übernommen.

Die wesentlichen Parameter, die der Linienextraktion zugrunde liegen, sind in Tabelle 4.5 zusammengefasst. Die Zahlenwerte wurden iterativ auf Basis von experimentellen Daten aus zwei verschiedenen Testumgebungen ermittelt und stellen einen Kompromiss zwischen einer möglichst hohen Genauigkeit bei der Abbildung der Umwelt, der Vermeidung von Artefakten bei Ungenauigkeiten in der Belegungskarte und einer begrenzten Datenmenge der entstehenden Linienkarte dar.

4.4.4 Gesamtsystem Infrastruktur

Das infrastrukturseitige Gesamtsystem beinhaltet neben den Algorithmen zur Kartenaktualisierung auch die Bereitstellung einer Schnittstelle für die Fahrzeugdaten. Abbildung 4.11 zeigt einen Überblick über die entsprechende Datenverarbeitungsstruktur. Für die Kommunikation mit den vernetzten Fahrzeugen existieren zwei Mechanismen: Einerseits wird ein Eingangspuffer bereitgestellt, der die Sensordaten der Fahrzeuge nach der Durchfahrt durch den kartierten Bereich entgegennimmt. Die Fahrzeuge legen hier die entsprechenden Daten aus LiDAR, GNSS, Odometrie und IMU als zusammengefassten Datensatz in einem binären Protobuf-Format [32] ab. Andererseits existiert eine Ausgangsschnittstelle, über die die Fahrzeuge die jeweils aktuellste Linienkarte abrufen können. Als Datenformat kommt hier die Extensible Markup Language (XML) zum Einsatz – die entsprechenden Tags sind an das GPS Exchange Format (GPX) angelehnt.

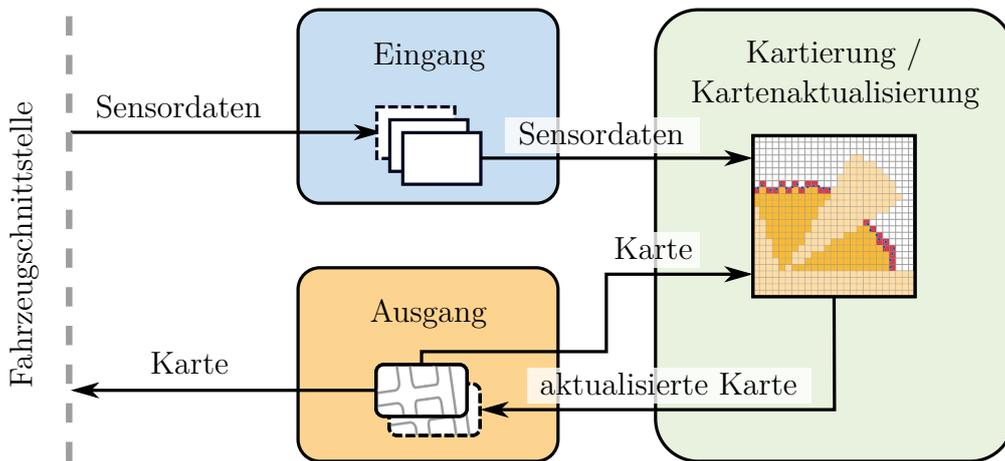


Abb. 4.11: Datenverarbeitung der infrastrukturseitigen Kartierungssoftware

Bei der Verarbeitung der Eingangsdaten nutzt die Kartierungssoftware ein First In – First Out (FIFO)-Prinzip. Die Datensätze werden also entsprechend ihrer Eingangsreihenfolge abgearbeitet. Für die Ausrichtung der in den Eingangsdaten enthaltenen Laserscans wird die jeweils aktuellste Linienkarte genutzt. Sobald ein Fahrzeugdatensatz vollständig verarbeitet wurde und somit eine neue Linienkarte verfügbar ist, wird diese auf der Ausgangsschnittstelle bereitgestellt.

5 Validierung

Die Validierung der entwickelten Algorithmen erfolgt anhand von experimentellen Untersuchungen in einem vom öffentlichen Verkehr abgetrennten Automobiltestzentrum. Hier kann ein passendes innerstädtisches Testszenario mit der Möglichkeit der Vernetzung zwischen einem prototypischen Testfahrzeug und den erforderlichen infrastrukturseitigen Komponenten realisiert werden. In den folgenden Kapiteln werden zunächst das Testumfeld und das Testfahrzeug inklusive der genutzten Sensorik vorgestellt. Anschließend werden die durchgeführten Experimente beschrieben und die Ergebnisse ausgewertet und diskutiert, wobei sich die beiden wesentlichen Schwerpunkte der Arbeit – die Kartierung und die Fahrzeuglokalisierung – in entsprechenden Unterkapiteln widerspiegeln.

5.1 Testumfeld

Das für die experimentelle Validierung genutzte Testfeld ist das „Aldenhoven Testing Center“, ein vom öffentlichen Verkehr abgetrenntes Testzentrum für Fahrzeugtechnik und Mobilität. Das Testzentrum beinhaltet auf einer Gesamtfläche von circa 40 ha verschiedenste Testelemente, auf denen Testszenarien für alle wesentlichen Automobilfunktionen realisiert werden können. [10]

Für die im Rahmen der vorliegenden Arbeit benötigten Umgebungsbedingungen wird das innerstädtische Testfeld genutzt, das aus mehreren verbundenen Kreuzungen mit Ampelanlagen, verschiedenen Fahrspurkonfigurationen und verstellbaren Kulissenwänden besteht. Abbildung 5.1 zeigt ein Luftbild des Testbereiches. Hervorgehoben ist das Areal, für das die digitale Karte erstellt und aktualisiert wird und in dem die kartengestützte Lokalisierung erfolgt. Es handelt sich dabei um einen Teilbereich des innerstädtischen Testfeldes, der auf einer Fläche von 4942 m^2 zwei Kreuzungen mit insgesamt fünf angeschlossenen Zufahrten umfasst. Die Verbindung zur Ovalbahn des Testzentrums am rechten Bildrand wird bei den Tests nicht befahren. Die gewählte Definition des Kartenbereiches erlaubt es, die auftretenden Effekte beim Übergang von einem nicht kartierten in einen kartierten Bereich systematisch zu testen und mit begrenztem Aufwand alle möglichen Routen durch das Areal im Rahmen von Testfahrten abzudecken. Neben dem kartierten Areal sind in der Abbildung zwei der Kommunikationsstationen des



Abb. 5.1: Luftbild des für die Validierung genutzten Testfeldes

Testzentrums sowie einige Kulissenwände markiert. Letztere dienen im Rahmen der Experimente als kartierte statische Elemente. Die Umrisse dieser Objekte werden mittels eines geodätischen GNSS-Empfängers mit einer Unsicherheit von unter 3 cm eingemessen.

Die Kommunikation zwischen Fahrzeugen und Infrastruktur kann am Aldenhoven Testing Center auf verschiedene Weisen realisiert werden. Mit dem „Vodafone 5G Mobility Lab“ steht eine flächendeckende Versorgung mit den Standards Global System for Mobile Communications (GSM), Universal Mobile Telecommunications System (UMTS) und Long Term Evolution (LTE) auf Basis von kleinzelligen Mobilfunknetzen sowohl als geschlossenes Test-Netz, als auch als offenes Live-Netz zur Verfügung. Als Erweiterungen zu LTE ermöglichen die Systeme LTE Advanced, LTE Broadcast (eMBMS) und LTE V2X eine latenzarme Anbindung von Fahrzeugen mit hoher Bandbreite. Durch die Integration eines MEC-Systems direkt am Testzentrum steht zudem infrastrukturseitig Rechenkapazität zur Verfügung, die es ermöglicht, rechenintensive Dienste für vernetzte Fahrzeuge in Form einer dezentralen Architektur mit schneller Kommunikationsanbindung zur Verfügung zu stellen. [85] Die beschriebene, auf Mobilfunk basierende Kommunikationsarchitektur wurde im Rahmen der in [1] veröffentlichten Vorarbeiten genutzt und im Hinblick auf die erzielbaren Datenraten und Latenzen evaluiert.

Neben der Mobilfunktechnologie verfügt das Testzentrum auch über eine vollständige Abdeckung mit Wireless Local Area Network (WLAN) gemäß IEEE-802.11b,g,n,ac sowohl mit 2.4 GHz, als auch mit 5 GHz. Eine fest installierte Netzwerkverkabelung auf Basis von Glasfaserkabeln und Kupferkabeln mit Gigabit-Ethernet stellt zudem zahlreiche Zugangspunkte für den Anschluss von infrastrukturseitigen Recheneinheiten bereit. Da für die im Rahmen dieser Arbeit durchgeführten Experimente das Mobilfunksystem nicht zur Verfügung stand, wird hier die Kommunikation mittels WLAN realisiert. Das Testfahrzeug verfügt hierfür über eine konventionelle WLAN-Schnittstelle, die sowohl die Kommunikation über 2.4 GHz, als auch über 5 GHz unterstützt. Als Recheneinheit wird ein eingebetteter Prototypencomputer mit Linux-Betriebssystem genutzt, der zu dem auf dem Fahrzeug installierten System (siehe Kapitel 5.2.2) identisch ist. Die Softwarearchitektur entspricht damit auch derjenigen des MEC-Systems [55], sodass eine Übertragbarkeit der Software gewährleistet ist.

5.2 Testfahrzeug

Als Testfahrzeug für die Validierung der fahrzeugseitigen Komponenten des Kartierungs- und Lokalisierungssystems wird ein IRT-Buggy eingesetzt. Es handelt sich hierbei um eine Eigenentwicklung des Institutes für Regelungstechnik der RWTH Aachen, die als flexible Plattform für Test- und Verifikationsaufgaben konzipiert ist [65].

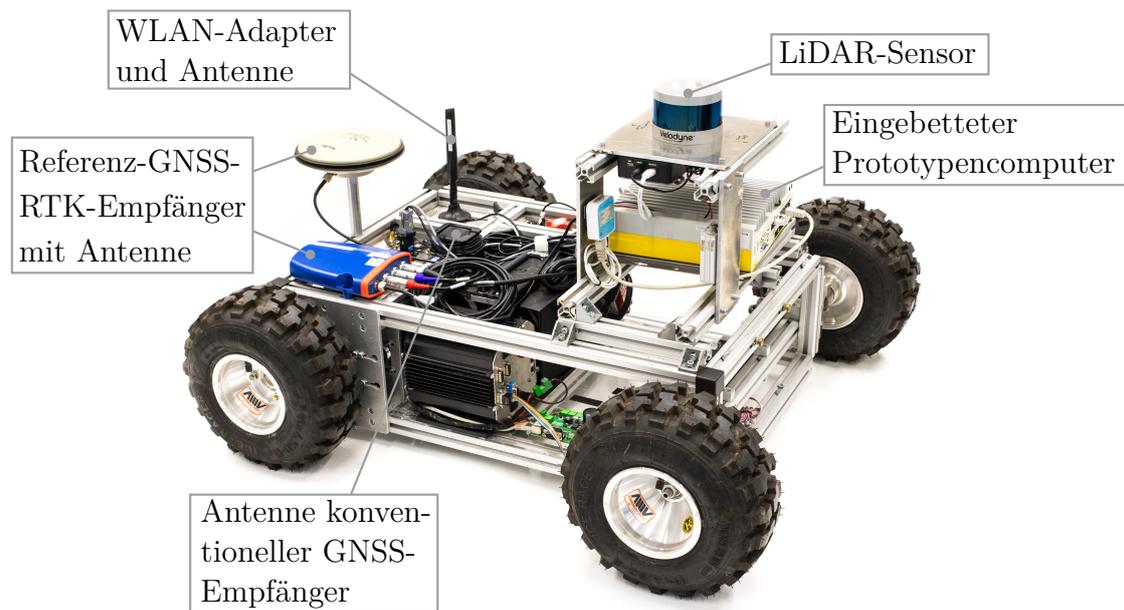


Abb. 5.2: IRT-Buggy mit Erweiterung für kartengestützte Lokalisierung

Abbildung 5.2 zeigt das Fahrzeug mit der im Rahmen der vorliegenden Arbeit entwickelten Erweiterung, welche die gegenüber der Basiskonfiguration zusätzlichen Sensor- und Recheneinheiten enthält. Der IRT-Buggy unterstützt in der genutzten Ausbaustufe automatisiertes Fahren auf Basis von vorprogrammierten digitalen Karten. Für die hier untersuchten Fragestellungen wird jedoch der manuelle Fahrbetrieb mittels Funkfernbedienung genutzt, da es wünschenswert ist, möglichst variable Fahrprofile mit entsprechenden Kurvengeschwindigkeiten und Beschleunigungs- und Bremsverhalten abzudecken. Das Testfahrzeug ist mit einer Höchstgeschwindigkeit von 40 km h^{-1} in der Lage, die für den kartierten Bereich realistischen Geschwindigkeitsbereiche abzudecken.

5.2.1 Sensorik

Um den grundlegenden Bewegungszustand des Fahrzeuges direkt und zuverlässig detektieren zu können, verfügt der IRT-Buggy über Odometriesensoren an den beiden Hinterrädern. Die berührungslos arbeitenden Hall-Effekt-Sensoren erlauben sowohl eine absolute Winkelbestimmung mit einer Auflösung von 10 bit, als auch die Ausgabe von Winkelinkrementen, sodass die Raddrehzahlen mit hoher Präzision ermittelt werden können. Das Antriebssystem des Fahrzeuges ist so konzipiert, dass die beiden Hinterräder über radindividuelle Elektromotoren angetrieben werden. Die beiden Raddrehzahlen werden unabhängig voneinander entsprechend der gewünschten Fahrgeschwindigkeit

eingeregelt, wobei bei Kurvenfahrt eine geometrische Korrektur gemäß des Ackermann-Prinzips vorgenommen wird. Die Regelung wirkt somit nicht den in Kurven entstehenden Drehzahlunterschieden entgegen. Die dynamischen Radhalbmesser der beiden Hinterräder werden experimentell ermittelt und als konstant angenommen, sodass von den Raddrehzahlen auf die Fahrzeuggeschwindigkeit und die zurückgelegte Strecke geschlossen werden kann. Antriebs- und Bremsschlupf werden hierbei nicht explizit korrigiert, sondern als statistische Sensorunsicherheit modelliert.

Im Mittelpunkt der Hinterachse des Fahrzeuges ist die IMU platziert. Sie definiert mit dieser Lage zugleich den verwendeten Ursprung des Fahrzeugkoordinatensystems. Die Genauigkeit des dreiachsigen Gyroskopsensors beträgt 0.05°s^{-1} bei einem Messbereich von $\pm 300^\circ \text{s}^{-1}$. Die translatorische Beschleunigungsmessung des Sensors wird in der vorliegenden Anwendung nicht genutzt.

Das LiDAR-System ist in einer Höhe von 55 cm über der Bodenebene im vorderen Bereich des Testfahrzeuges angebracht. Der Sensor wird mit einer Drehzahl von 1200 min^{-1} betrieben und erreicht eine Winkelauflösung in Rotationsrichtung von 0.4° .

Als relativ kostengünstiges, konventionelles GNSS-System kommt ein Einfrequenzempfänger mit Unterstützung für GPS und GLONASS zum Einsatz. Positionsupdates stehen mit einer Rate von 5 Hz zur Verfügung.

Das Referenzsystem zur Ermittlung der Genauigkeiten der Lokalisierungs- und Kartierungsalgorithmen ist ein RTK-fähiger Mehrfrequenz-Satellitenempfänger. Durch die Bereitstellung von Echtzeitkorrekturdaten über den Satellitenpositionierungsdienst der deutschen Landesvermessung (SAPOS) unter Verwendung einer mobile Internetverbindung wird eine Positionsgenauigkeit mit einer Standardabweichung unter 1 cm erreicht.

Im Rahmen der Sensordatenverarbeitung werden die geometrischen Informationen aller Sensorsysteme auf den Ursprung des Fahrzeugkoordinatensystems am Mittelpunkt der Hinterachse transformiert. Zudem werden die im Rahmen einer manuellen Kalibrierung ermittelten Zeitdifferenzen aufgrund von Totzeiten bei der Datenvorverarbeitung und -übertragung bei der Datenauswertung berücksichtigt und korrigiert. Detaillierte Spezifikationen der genutzten Sensorsysteme und Kommunikationsschnittstellen sind im Anhang A.1 dargestellt.

5.2.2 Prototypencomputer

Die fahrzeugseitigen Algorithmen werden auf einem eingebetteten Prototypencomputer mit x86-Architektur ausgeführt. Der lüfterlose, passiv gekühlte Rechner kommt ohne mechanisch bewegliche Komponenten aus und wird über das Bordnetz des Testfahrzeuges mit Strom versorgt. Als Betriebssystem wird XUbuntu Linux 16.04LTS mit der Kernelversion 4.4.0-71 verwendet. Der Rechner nutzt eine CPU vom Typ Intel i7-6700TE

mit 4 Kernen, 8 Threads und einer maximalen Turbo-Taktfrequenz von 3.4 GHz. Der Arbeitsspeicher beträgt 16 GB, eine dedizierte Graphikkarte ist nicht vorhanden.

Der Prototypencomputer ermöglicht den direkten Anschluss des Laserscanners über Ethernet und der beiden GNSS-Empfänger sowie des WLAN-Adapters über serielle Schnittstellen. Zudem greift der Rechner über eine CAN-Bus-Schnittstelle auf die Basissensorik des IRT-Buggys zu, sodass auch die Daten von IMU und die Raddrehzahlinformationen direkt in der Software zur Verfügung stehen.

5.3 Methodik

Zur Verifikation der entwickelten Algorithmen werden Messkampagnen mit dem IRT Buggy auf dem beschriebenen Testgelände durchgeführt. Das Testfahrzeug wird dabei manuell fernbedient gesteuert, wobei die geltenden Verkehrsregeln – etwa bei der Wahl der Fahrspur oder der Fahrgeschwindigkeit beim Abbiegen – eingehalten werden. Zugleich wird angestrebt, eine möglichst variantenreiche Fahrweise abzubilden und alle möglichen Pfade durch den kartierten Bereich zu berücksichtigen. Abbildung 5.3 zeigt einen Überblick über die auf diese Weise gewonnenen Trajektorien im lokalen ENU-Koordinatensystem. Dargestellt sind die Positionsdaten des RTK-fähigen GNSS-Empfängers in gelber Farbe, die definierte Kartengrenze in Blau sowie die Elemente der statischen Referenzkarte in Rot. Wie die Graphik zeigt, entspricht jede Messung einer vollständigen Durchfahrt durch den kartierten Bereich. Start- und Endpunkt liegen also außerhalb bzw. am Rand der Kartengrenzen, sodass die Vorgänge beim Einfahren in das Kartensegment und beim Verlassen des Segmentes vollständig abgebildet werden können. Im Zuge der Auswertungen der Messdaten wird stets der Ausschnitt einer Messung betrachtet, bei dem sich das Versuchsfahrzeug innerhalb der Kartengrenzen befindet.

Während der Messfahrten werden sowohl die Algorithmen zur Kartierung und Lokalisierung im Echtzeitbetrieb ausgeführt, als auch die Rohdaten aller Sensoren vollständig mit Zeitstempeln aufgezeichnet. Letzteres ermöglicht die nachträgliche Reproduktion der Ergebnisse mittels des Simulationsmodus der Fahrzeug- und Infrastruktursoftware. Im Simulationsmodus erfolgt ein echtzeitsynchrones Abspielen der Rohdaten. Dem Fahrzeug- bzw. Infrastrukturcomputer können damit die zum Realbetrieb identischen Signale in identischer zeitlicher Abfolge bereitgestellt werden, sodass das Systemverhalten im Detail untersucht und dokumentiert werden kann. Insbesondere für die Validierung der Robustheit der Kartierungssoftware ergibt sich hiermit die Möglichkeit, besonders kritische Szenarien aus den aufgezeichneten Daten zusammenzustellen. Die Messfahrten können in verschiedenen Reihenfolgen verarbeitet werden und es kann geprüft werden, ob die Kartierungssoftware stets adäquat auf die in den verschiedenen Messungen enthaltenen Änderungen des kartierten Umfeldes reagiert.

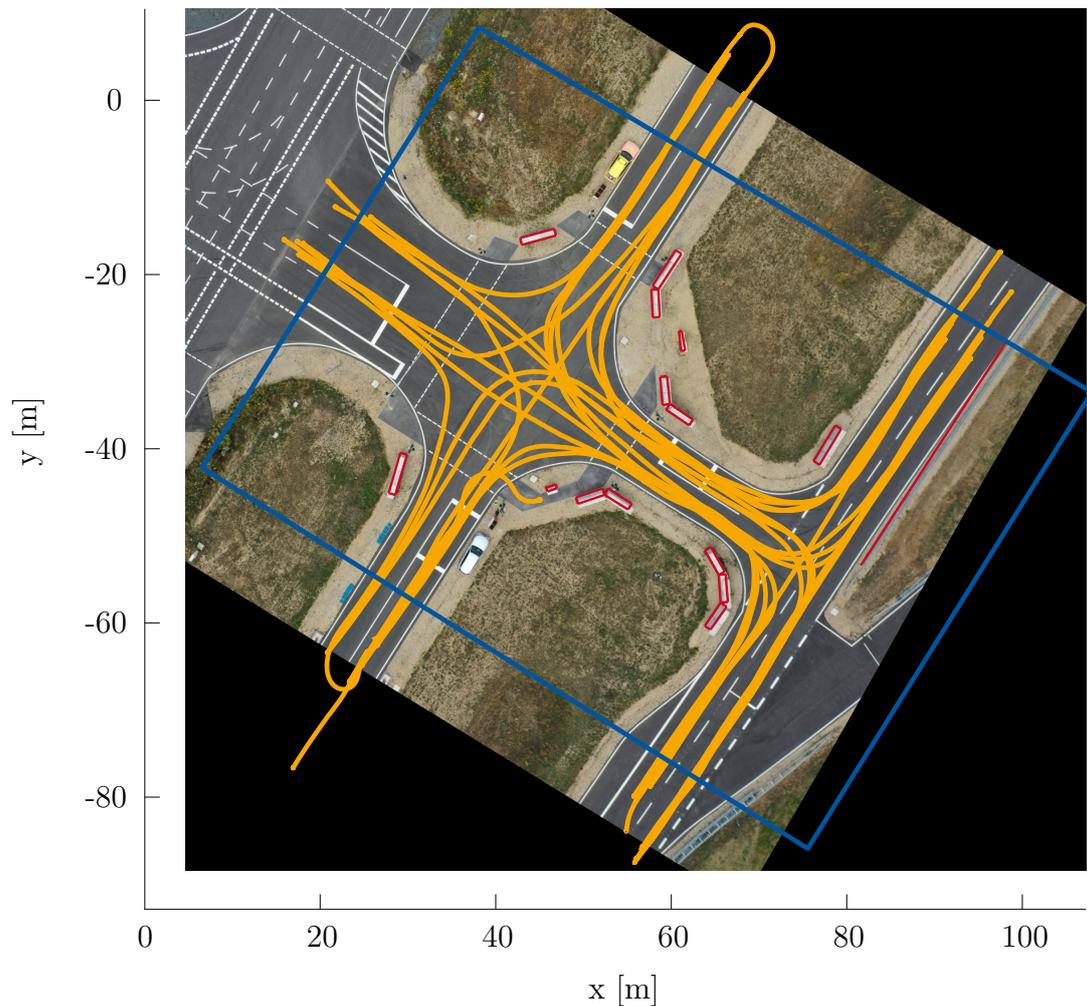


Abb. 5.3: Datensatz der Trajektorien

Im Rahmen der beschriebenen Messkampagnen werden verschiedene, spezifische Szenarien untersucht, bei denen innerhalb des kartierten Bereiches bewegliche und unbewegliche Objekte platziert und verändert werden. In verschiedenen Konstellationen werden dabei geparkte PKW genutzt, die als a priori unbekannte, aber unbewegliche Objekte für die Lokalisierung relevant sind und in der Karte verzeichnet werden sollen. Weitere unbewegliche Objekte werden mittels großer, rechteckiger Pappkartons generiert. Hiermit können veränderliche, kastenförmige Hindernisse erzeugt werden oder vollständige bzw. teilweise Verdeckungen bekannter Hindernisse realisiert werden. Neben unbewegten Objekten werden auch bewegte Verkehrsteilnehmer berücksichtigt. Mehrere Fußgänger durchqueren den kartierten Bereich während der Messfahrten, sodass der LiDAR-Sensor langsam bewegte oder zum Teil still stehende Objekte detektiert, die nicht für die

Lokalisierung geeignet sind und nicht kartiert werden sollen. Gleiches gilt für fahrende und zwischenzeitlich anhaltende PKW, die zudem – je nach geometrischer Konstellation – zu einer Verdeckung mehr oder weniger großer Teilbereiche des LiDAR-Sichtfeldes führen.

5.4 Ergebnisse

Bei der Darstellung der experimentellen Ergebnisse wird im Folgenden zunächst auf das infrastrukturseitige System zur Kartierung eingegangen. Anschließend erfolgt die quantitative Untersuchung der mittels der digitalen Karte erzielbaren Eigenlokalisierung des Testfahrzeuges. Die Vorgänge beim Einfahren in den kartierten Bereich bzw. beim Verlassen desselben sowie die Kommunikations- und Datenübertragungsaspekte werden im letzten Teil des Kapitels ausgewertet.

5.4.1 Kartierung

Als erster Schritt der Kartierung wird zunächst eine quantitative Auswertung der Trajektorienrekonstruktion vorgenommen. Anschließend wird die Qualität der auf Basis dieser Trajektorie abgeleiteten digitalen Karte evaluiert.

5.4.1.1 Trajektorienrekonstruktion

Nachdem die grundsätzliche Wirksamkeit der Algorithmen zur Trajektorienrekonstruktion bereits in Kapitel 4.4.1 dargestellt wurde, soll im Folgenden eine quantitative Auswertung im Hinblick auf die Genauigkeit, Robustheit und den Zeitaufwand der Verfahren vorgenommen werden.

Genauigkeit Position Um eine Aussage über die Genauigkeit der durch die Optimierung ermittelten Trajektorie treffen zu können, wird der hochgenaue, RTK-fähige GNSS-Empfänger als Referenz genutzt. Eine möglichst große Datenbasis kann erzielt werden, indem die entsprechende Analyse für alle zur Verfügung stehenden Messfahrten durchgeführt wird. Als Fehlermaß wird zunächst der euklidische Abstand zwischen der optimierten Position jedes Knotens des SLAM-Graphen und der entsprechenden Referenzposition des RTK-GNSS-Empfängers verwendet. Da die GNSS-Daten nur mit einer Rate von 10 Hz zur Verfügung stehen, wohingegen die optimierte Trajektorie eine zeitliche Auflösung von 100 Hz aufweist, werden zusätzliche Stützstellen durch lineare Interpolation des GNSS-Signals erzeugt. Bei der Auswertung werden Messfahrten, bei denen aufgrund von spezifischen Fehlern die Trajektorienrekonstruktion fehlschlägt, nicht mit

berücksichtigt. Die entsprechenden Fehlermechanismen werden im Nachgang gesondert diskutiert.

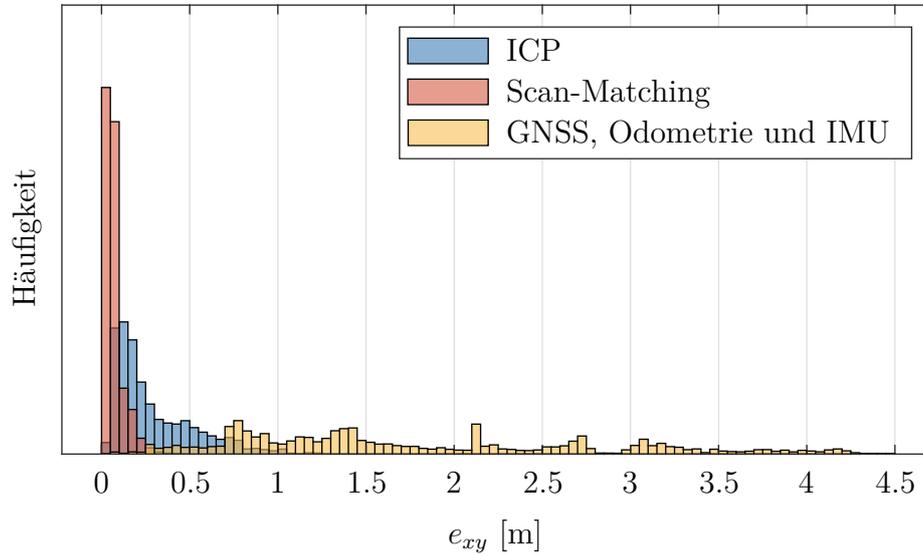


Abb. 5.4: Häufigkeitsverteilung der euklidischen Abstandsfehler bei der Trajektorienrekonstruktion mit und ohne LiDAR-Daten

Abbildung 5.4 zeigt die Häufigkeitsverteilungen der ermittelten Abstandsfehler für drei verschiedene Fälle: In Gelb ist die Verteilung für eine Trajektorienrekonstruktion ohne LiDAR-Informationen, also lediglich auf Basis des konventionellen GNSS-Empfängers sowie der Odometrie- und IMU-Daten dargestellt. Die euklidischen Fehler betragen hier zum Teil mehr als 4 m, da das Optimierungsverfahren systematische Abweichungen des GNSS-Empfängers aufgrund der relativen Lageinformationen aus Odometrie und IMU nicht ausgleichen kann. Die Nutzung der LiDAR-Daten für einen Abgleich mit der digitalen Karte kann diese systematischen Fehler ausgleichen und führt zu einer wesentlich höheren Genauigkeit. In Abbildung 5.5 ist die Gegenüberstellung der verbleibenden Fehler von ICP und Scan-Matching vergrößert dargestellt. Für das Scan-Matching liegt bei den ausgewerteten Testfahrten der Hochpunkt der Häufigkeitsverteilung bei einem Fehler zwischen 3 cm und 4.5 cm, die maximalen Einzelfehler liegen unter 0.33 m. Unter Berücksichtigung der Auflösung des Suchraumes von 5 cm (siehe Tabelle 4.3) kann dieses Ergebnis als sehr gut gewertet werden. Der ICP-Algorithmus schneidet im Vergleich deutlich schlechter ab. Zwar liegen die meisten Abweichungen mit dem Intervall zwischen 7.5 cm und 9 cm ebenfalls in einem sehr guten Bereich, es ergeben sich jedoch auch maximale Einzelfehler von bis zu 1.08 m.

Abbildung 5.6 visualisiert einen Fehlermechanismus, der bei der Ausrichtung mit ICP im genutzten Testumfeld auftreten kann und die Genauigkeit der Lokalisierung be-

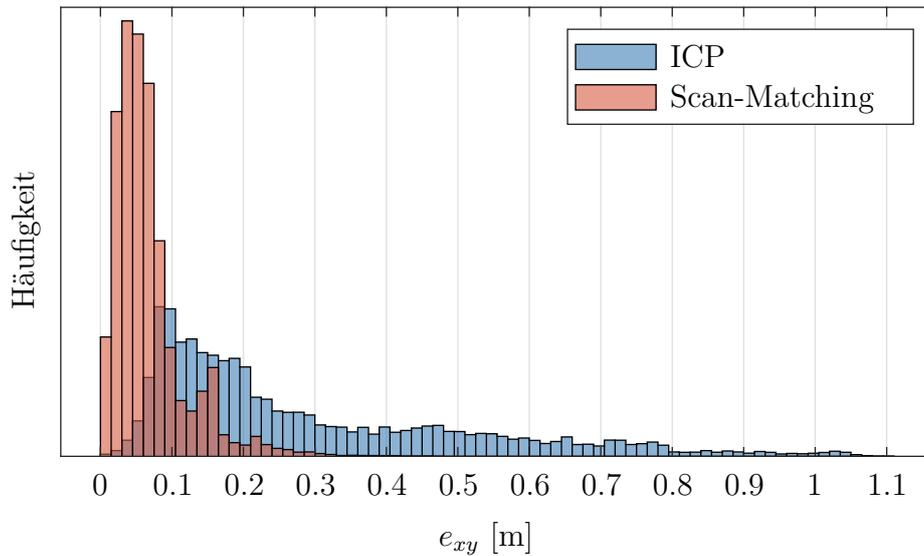


Abb. 5.5: Häufigkeitsverteilung der euklidischen Abstandsfehler bei Scan-Matching und ICP

einträchtig. Dargestellt sind verschiedene Transformationen eines vom LiDAR-Sensor aufgenommenen Laserscans. Die schwarzen Linien zeigen die Referenzkarte, an der der Laserscan ausgerichtet werden soll. In Rot ist die sich ergebende Punktwolke bei der Transformation der LiDAR-Daten mit der initialen Position auf Basis des konventionellen GNSS-Empfängers sowie der Odometrie- und IMU-Daten dargestellt. Die gelbe Punktwolke stellt das Ergebnis der Ausrichtung mittels ICP dar.

Im Vergleich zur grünen Punktwolke, die die optimale Ausrichtung des Laserscans an der Referenzkarte repräsentiert, weist das Ergebnis des ICP-Algorithmus wesentliche Abweichungen auf. Zwar wurde die Ausrichtung gegenüber der initialen Punktwolke verbessert, das globale Optimum konnte allerdings nicht erreicht werden. Der Grund hierfür liegt in der iterativen Struktur des ICP-Algorithmus, die eine sukzessive Verringerung des Fehlermaßes realisiert. Mit diesem Prinzip ist ICP anfällig für die Konvergenz gegen lokale Minima. Im vorliegenden Beispiel besteht ein solches lokales Minimum in der Ausrichtung der Punktwolke an der aus Sicht des Laserscanners verdeckten Rückseite einiger kartierter Objekte. Die auf der Teststrecke errichteten Wände haben die besondere Eigenschaft, auf der Vor- und Rückseite eine praktisch identische Geometrie aufzuweisen, sodass für die Ausrichtung eines vom Laserscanner erkannten Linienzuges häufig zwei plausible Lösungen existieren. Erst aus der Kombination gleichzeitig erkannter Objekte ist in der Regel die Ermittlung eines eindeutigen Optimums möglich. Im dargestellten Fall ist erkennbar, dass der mittels ICP ausgerichtete Laserscan in einigen Bereichen eine sehr gute Überlappung mit der „falschen“ Seite eines kartierten Objektes aufweist, die zu einem geringen Fehlermaß bei der Optimierung führt. Zwar ergeben sich in an-

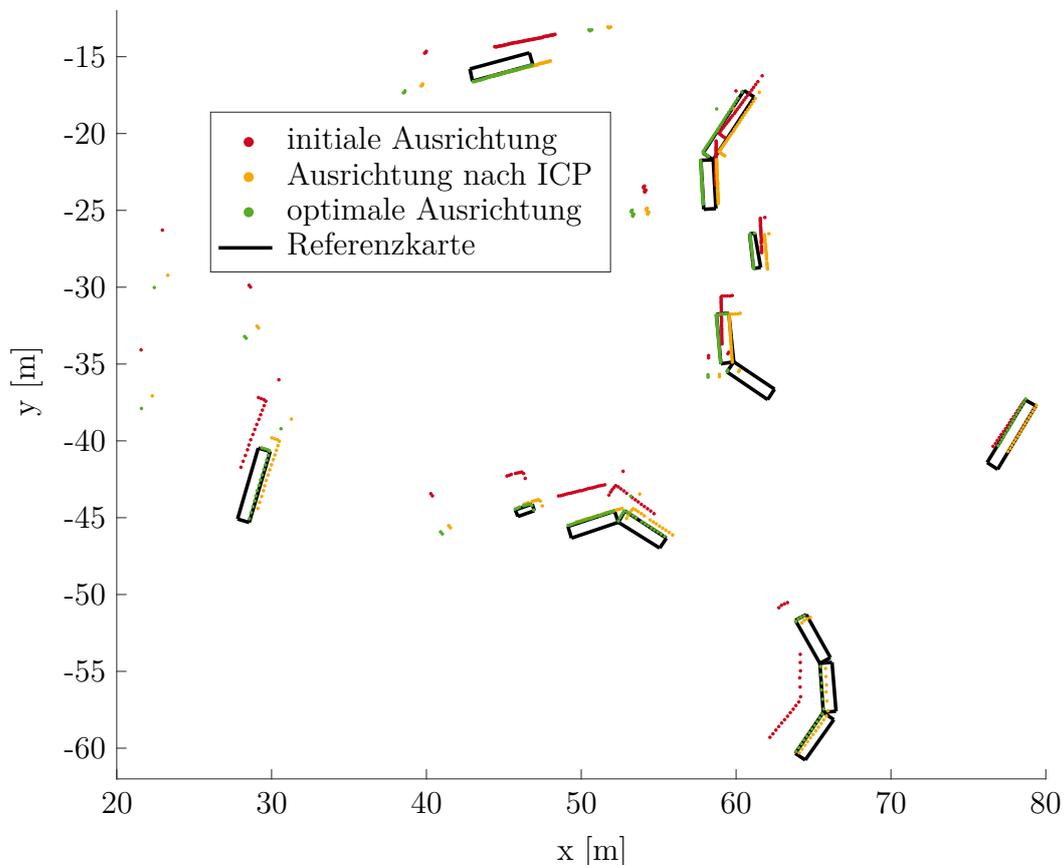


Abb. 5.6: Konvergenz gegen lokales Minimum bei der Anwendung von ICP

deren Bereichen zugleich suboptimale Übereinstimmungen; trotzdem stellt die gezeigte Ausrichtung ein klares Minimum der Kostenfunktion dar, gegen das der Algorithmus konvergiert. Welches Minimum mittels ICP erreicht wird, hängt in einem solchen Fall wesentlich von der Startbedingung, also der initialen Ausrichtung ab. Dass die Dicke der auf der Teststrecke errichteten Wände und damit die potenzielle Uneindeutigkeit bei der Ausrichtung 80 cm beträgt, ist eine Erklärung für die in Abbildung 5.5 erkennbare Stufe bei einem Abstandsfehler von 0.8 m für das ICP-Verfahren, der nur von sehr wenigen Messpunkten überschritten wird.

Da das Scan-Matching im Gegensatz zum ICP-Verfahren eine globale Optimierung im betrachteten Suchbereich vornimmt, ist der beschriebene Fehlermechanismus hier nicht vorzufinden und es wird im betrachteten Testumfeld eine deutlich genauere Lokalisierung erreicht. Für den Einsatz in realen, städtischen Umgebungen kann davon ausgegangen werden, dass die systematischen Fehler des ICP-Verfahrens deutlich geringer ausfallen, da die im Testfeld beobachteten starken Doppeldeutigkeiten der besonderen Geometrie der

hier kartierten Objekte geschuldet sind. Bei realer städtischer Bebauung sind dagegen wesentlich komplexere Geometrien mit geringeren Symmetrieeffekten zu erwarten.

Genauigkeit Orientierung Neben der Genauigkeit der mittels GraphSLAM rekonstruierten Position (x, y) spielt für die Qualität der automatisch erzeugten digitalen Karte auch die Orientierung φ eine wesentliche Rolle. Da der Referenz-GNSS-Empfänger die Fahrzeugorientierung nicht direkt messen kann, steht für die quantitative Auswertung der Orientierung φ kein verlässliches Referenzsignal zur Verfügung. Allerdings kann aus den GNSS-Positionsmessungen eine indirekte Berechnung der Orientierung erfolgen. Abbildung 5.7 zeigt eine Gegenüberstellung der auf Basis der GNSS-Daten mittels zentraler Differenzen ermittelten Orientierung mit der im Rahmen der Trajektorienrekonstruktion bestimmten Orientierung. Da sich kleine Ungenauigkeiten bei der GNSS-Positionsbestimmung relativ stark auf die daraus berechnete Orientierung auswirken, zeigt das Referenzsignal ein deutliches Rauschen. Die unter Verwendung von Scan-Matching optimierte Trajektorie verläuft dagegen völlig glatt und bildet den gleitenden Mittelwert des Referenzsignals gut ab. Die Auswertung deutet somit darauf hin, dass auch die Orientierungsschätzung mittels GraphSLAM den Genauigkeitsanforderungen der vorliegenden Anwendung genügt.

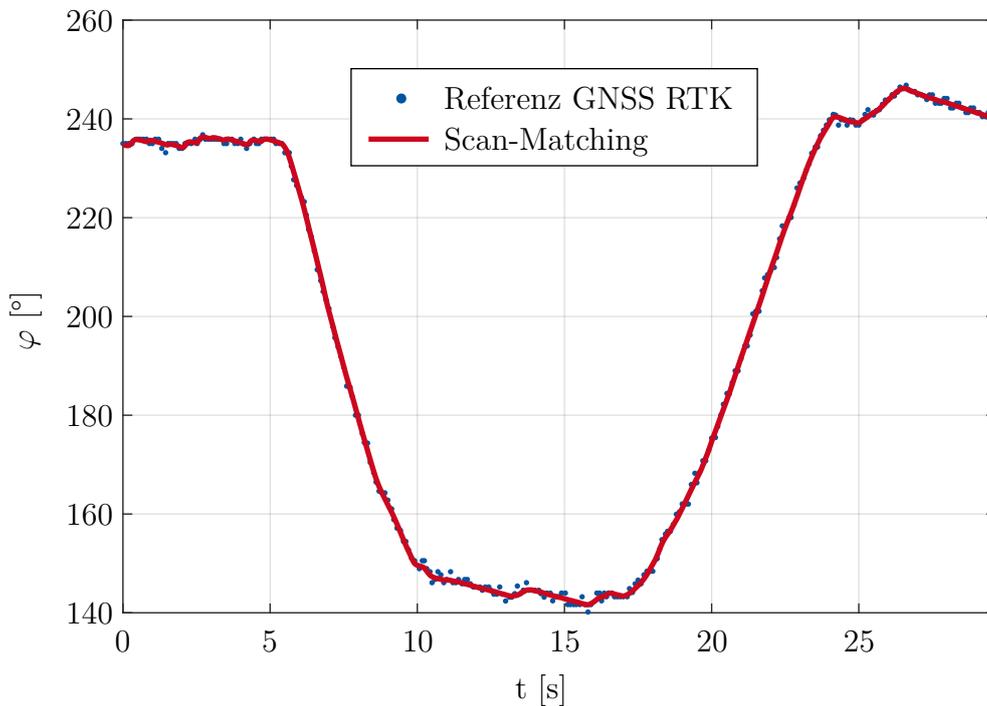


Abb. 5.7: Geschätzte Fahrzeugorientierung aus GraphSLAM und GNSS-Referenz

Abbildung 5.8 zeigt die Differenz der geschätzten Orientierungen bei Verwendung des Scan-Matching- bzw. ICP-Verfahrens für die bereits in Abbildung 5.7 dargestellte Messfahrt. Die maximale Abweichung zwischen beiden Schätzungen liegt im Bereich von $\pm 1.7^\circ$. Der Vergleich mit dem GNSS-Referenzsignal legt nahe, dass diese Abweichung größtenteils aufgrund von Fehlern beim Ausrichten der Punktwolken mittels ICP zustande kommt und bei der Orientierungsschätzung mittels Scan-Matching eine höhere Genauigkeit erzielt wird. Der wesentliche Fehlermechanismus beim ICP-Verfahren besteht auch im Bezug auf die Orientierung in der in Abbildung 5.6 visualisierten Konvergenz gegen lokale Minima.

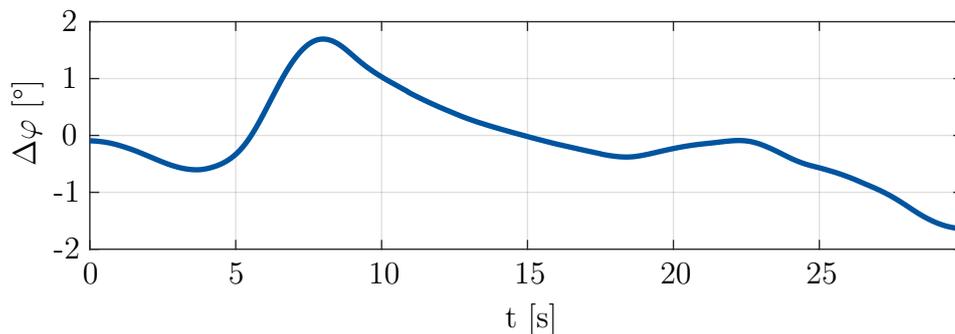


Abb. 5.8: Differenz Fahrzeugorientierung zwischen Scan-Matching und ICP

Sonderfälle Ein möglicher Fehlerfall, der zu einem Scheitern der genauen Trajektorienrekonstruktion führen kann, wird durch Lokalisierungsfehler des GNSS-Empfängers hervorgerufen. In Abbildung 5.9 ist ein solcher Fall dargestellt. Die Signale des GNSS-Empfängers wurden in diesem Fall durch eine Abschirmung der Antenne so stark verschlechtert, dass auch die nach der Fusion mit Odometrie und IMU als Ergebnis der ersten Optimierung bestimmte Trajektorie noch weit von der Realität abweicht. Von den Datenpunkten des Referenz-GNSS-Empfängers wurde im Interesse einer besseren Erkennbarkeit nur jeder fünfte Punkt dargestellt. Es ist erkennbar, dass in diesem Fall das Scan-Matching nicht in der Lage ist, die optimale Ausrichtung des Laserscans zur digitalen Karte zu ermitteln, sodass die Trajektorie auch nach der zweiten Optimierung nicht die Referenz erreicht. Zugleich zeigt die Abbildung, dass die Ausrichtung mittels ICP hier nicht zu Problemen führt. Das iterative Verfahren liefert eine Trajektorie, die nur sehr geringe Abweichungen von der Referenz aufweist.

Der Grund für die dargestellte Problematik liegt im beschränkten Suchraum des Scan-Matching-Algorithmus. Wie in Kapitel 4.4.1 dargestellt wurde, muss für dieses Verfahren ein Bereich der Transformationskoordinaten (x, y, φ) definiert werden, in dem das globale Optimum gesucht wird. Zur sinnvollen Beschränkung der Rechenzeit kann dieser

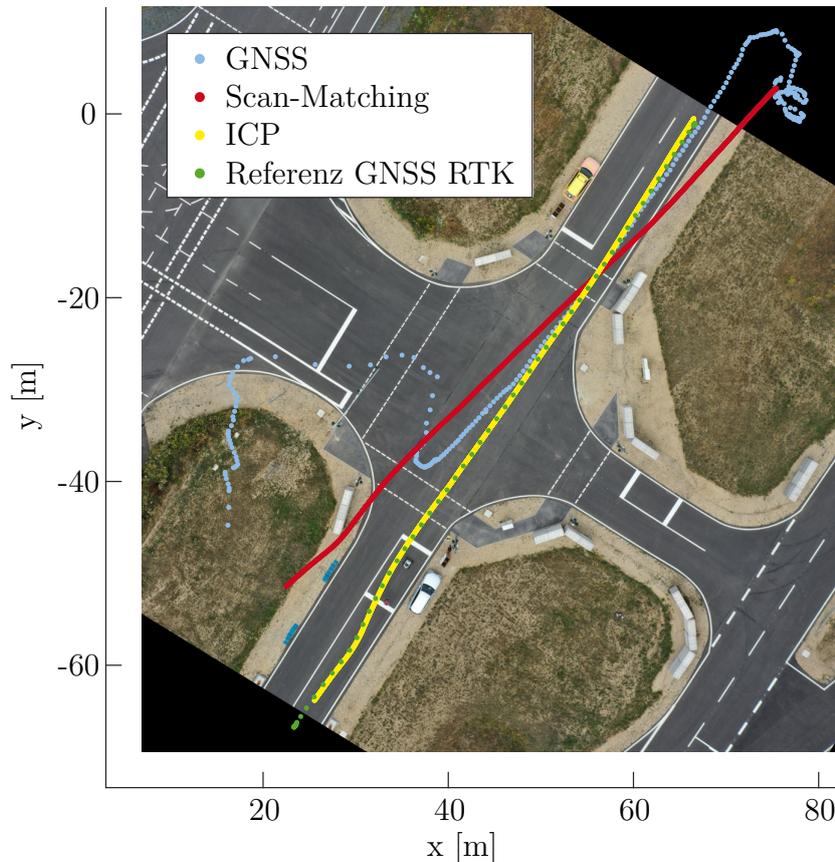


Abb. 5.9: Fehler bei der Trajektorienrekonstruktion mittels Scan-Matching

Suchbereich nicht beliebig groß gewählt werden. Wie Tabelle 4.3 zeigt, wurden der translatorische Suchradius auf $\pm 8\text{ m}$ und der rotatorische Suchradius auf $\pm 8^\circ$ festgelegt, um den Großteil der getesteten Szenarien abdecken zu können. Im hier gezeigten Beispiel liegen die Abweichungen zwischen der geschätzten Fahrzeugtrajektorie nach der 1. Optimierung und der optimalen Ausrichtung allerdings außerhalb des Suchbereiches, sodass die vom Scan-Matching-Algorithmus ermittelte optimale Transformation nicht zu einer korrekten Ausrichtung von Laserscan und Referenzkarte führt. Abbildung 5.10 zeigt das Ergebnis des selben Testfalls bei einer Erweiterung des Suchbereiches auf $\pm 22\text{ m}$ und $\pm 13^\circ$. Mit dieser Parametrierung erreicht auch der Scan-Matching-Algorithmus eine optimale Ausrichtung der Punktwolken und somit eine korrekte Rekonstruktion der Fahrzeugtrajektorie.

Im Gegensatz zum Scan-Matching ist beim ICP-Verfahren der Suchraum nicht prinzipbedingt begrenzt. Durch die iterative Zuordnung neuer Punktkorrespondenzen können auch größere Abweichungen zwischen einer Punktwolke und der Referenz ausgeglichen werden.

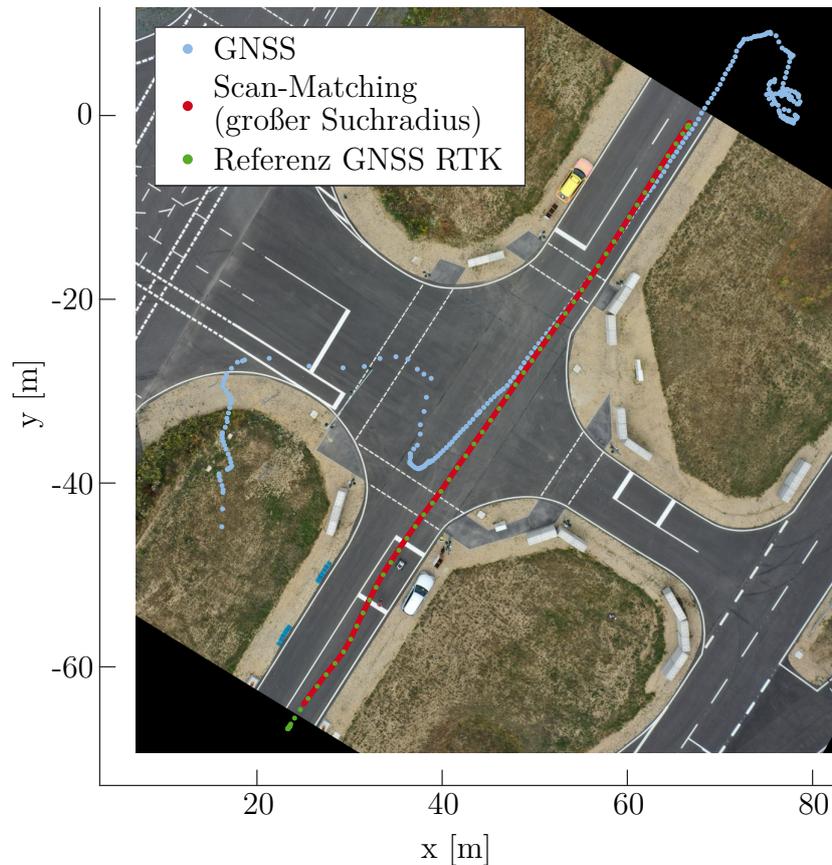


Abb. 5.10: Trajektorienrekonstruktion mittels Scan-Matching mit erhöhtem Suchradius

Die hier eingesetzte zweifache Schleife aus GraphSLAM-Optimierung und ICP-Matching verbessert die Robustheit des Verfahrens in diesen Fällen zusätzlich.

Ein weiterer Fehlermechanismus, der bei einzelnen Messfahrten beobachtet wurde, tritt auf, wenn ein großer Teil der für die Ausrichtung von Laserscans genutzten kartierten Geometrien durch unkartierte Objekte verdeckt wird. In Abbildung 5.11 ist ein solches Szenario erkennbar. Die kartierte Betonwand wird hier in weiten Teilen von einem geparkten Fahrzeug sowie vier kastenförmigen Hindernissen verdeckt. Im Sichtbereich des Versuchsfahrzeuges befinden sich zudem nur wenige andere Geometrien, die mit kartierten Elementen in Übereinstimmung gebracht werden können.

Wie Abbildung 5.12 zeigt, schlägt das Scan-Matching in diesem Fall fehl. Der gelb dargestellte Laserscan, der zum betrachteten Zeitpunkt hauptsächlich die Kontur des geparkten Fahrzeuges sowie die kastenförmigen Hindernisse abbildet, wird fälschlicherweise an der in der Karte enthaltenen Betonwand ausgerichtet. Bei weiterer Vorbeifahrt



Abb. 5.11: Weitgehende Verdeckung eines kartierten statischen Objektes durch bewegliche Hindernisse

des Versuchsfahrzeuges werden zwar wieder mehr bekannte Geometrien sichtbar und die Ausrichtung der Laserscans gelingt wieder korrekt. Nichtsdestotrotz weist die im Rahmen der GraphSLAM-Optimierung berechnete, finale Trajektorie im hier dargestellten kritischen Bereich deutliche Fehler auf. Dieser Fehlermechanismus ist unabhängig von der genutzten Methode zur Ausrichtung der Laserscans. Er tritt bei ICP und Scan-Matching in praktisch identischer Form auf.

Als zweiter Fall wird in Abbildung 5.12 auch das Ergebnis des Scan-Matchings dargestellt, nachdem auf Basis mehrerer anderer Messfahrten, bei denen kein fehlerhaftes Matching auftrat, die zusätzlichen Objekte zur erweiterten Referenzkarte automatisiert hinzugefügt wurden. Der identische Laserscan wird bei Kenntnis der zugehörigen Geometrien vollkommen korrekt ausgerichtet (grüne Punktwolke), sodass auch die entsprechende Fahrzeugtrajektorie präzise rekonstruiert werden kann.

Der dargestellte Fehlerfall ist prinzipbedingt nicht vermeidbar. Sind im Extremfall alle kartierten Geometrien durch unbekannte Objekte verdeckt, so kann die Ausrichtung der LiDAR-Punktwolken nicht korrekt gelingen. In der Regel kann jedoch davon ausgegangen werden, dass Änderungen im Verkehrsumfeld zwischen aufeinanderfolgenden Fahrzeugen nur Teilbereiche des LiDAR-Sichtfeldes betreffen, sodass die neuen Objekte sukzessive in die Referenzkarte übernommen werden können. Des Weiteren führt eine größere Dichte der Bebauung und damit der kartierten Objekte dazu, dass auch bei einzelnen Verdeckungen noch hinreichend Geometrien für die korrekte Ausrichtung der LiDAR-Daten verfügbar sind. Schlussendlich besteht eine wichtige Anforderung an den

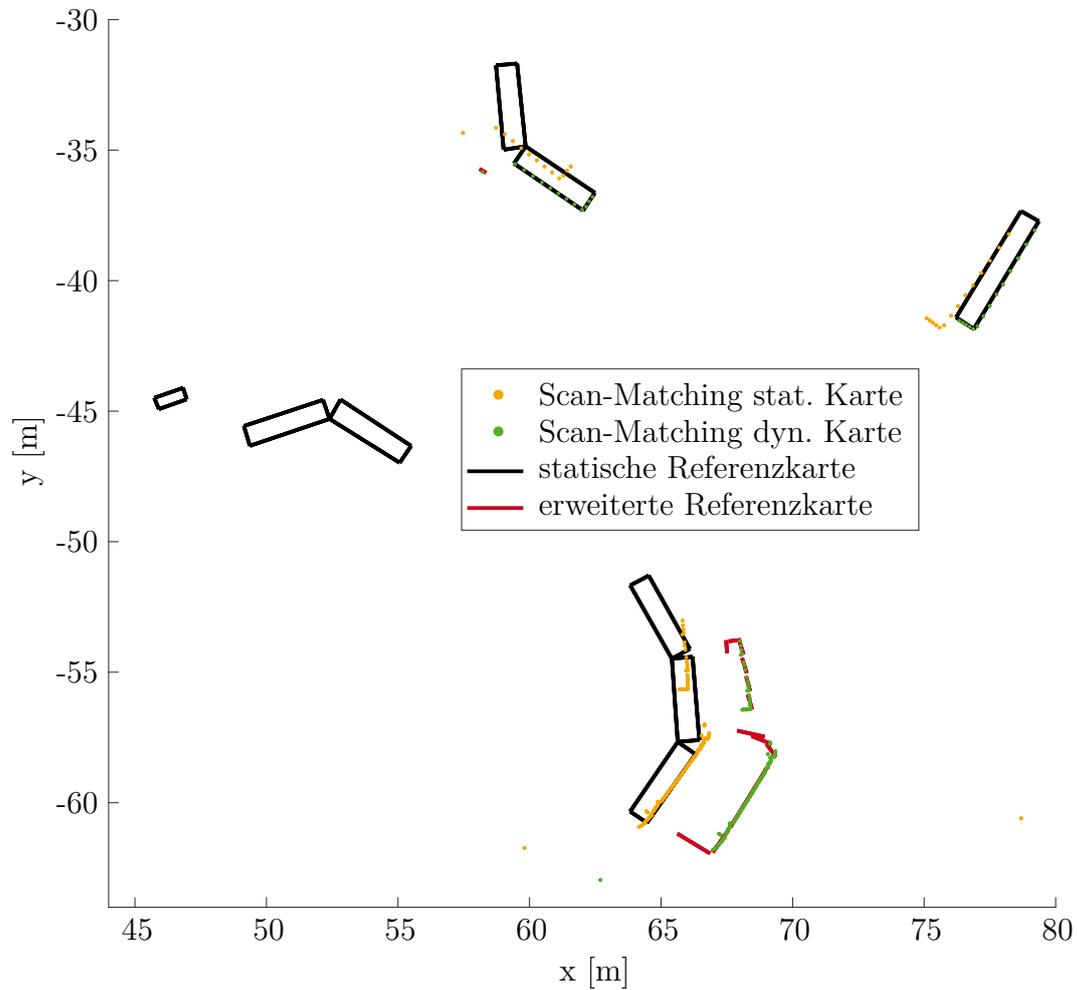


Abb. 5.12: Scan-Matching bei unkartierten Objekten

Algorithmus zur Aktualisierung der Karte darin, gegenüber einzelnen Fehlmessungen robust zu reagieren (siehe Kapitel 5.4.1.2), sodass trotz unvermeidbarer Fehler bei der Trajektorienrekonstruktion die Integrität der gesamten Karte nicht gefährdet wird.

5.4.1.2 Aktualisierung der Karte

Auf Basis der rekonstruierten Trajektorie und der LiDAR-Daten wird als zweiter Schritt des Kartierungsalgorithmus die Aktualisierung der Karte vorgenommen, deren qualitative und quantitative Auswertung in den folgenden Abschnitten dargestellt wird.

Kartierung veränderlicher Objekte Die wichtigste Anforderung an das System besteht darin, bislang unbekannte unbewegliche Objekte der digitalen Karte hinzuzufügen und zugleich solche Objekte zu entfernen, die noch kartiert, aber in der Realität nicht mehr vorhanden sind. Abbildung 5.13 zeigt das Luftbild eines Testszenarios, bei dem die zwei hervorgehobenen Objekte auf dem Testgelände wechselweise platziert und entfernt wurden. Es werden hierbei drei Phasen unterschieden: In Phase 1 befindet sich nur der geparkte PKW auf der dargestellten Position. Hinzu kommt in Phase 2 das Hindernis aus zwei rechteckigen Kartons (diese Phase ist im Bild dargestellt). In Phase 3 wird der geparkte PKW entfernt und es verbleibt das kastenförmige Hindernis. In jeder Phase werden mehrere Durchfahrten mit dem Messfahrzeug realisiert, von denen einige direkt an den kartierten Hindernissen vorbei führen und andere den betrachteten Bereich nur partiell abdecken. Zwischen den einzelnen Durchfahrten liegt jeweils ein Zeitraum $\Delta t > \Delta t_{Sat}$, sodass die Gewichtung der Messfahrten nicht zeitabhängig reduziert wird.

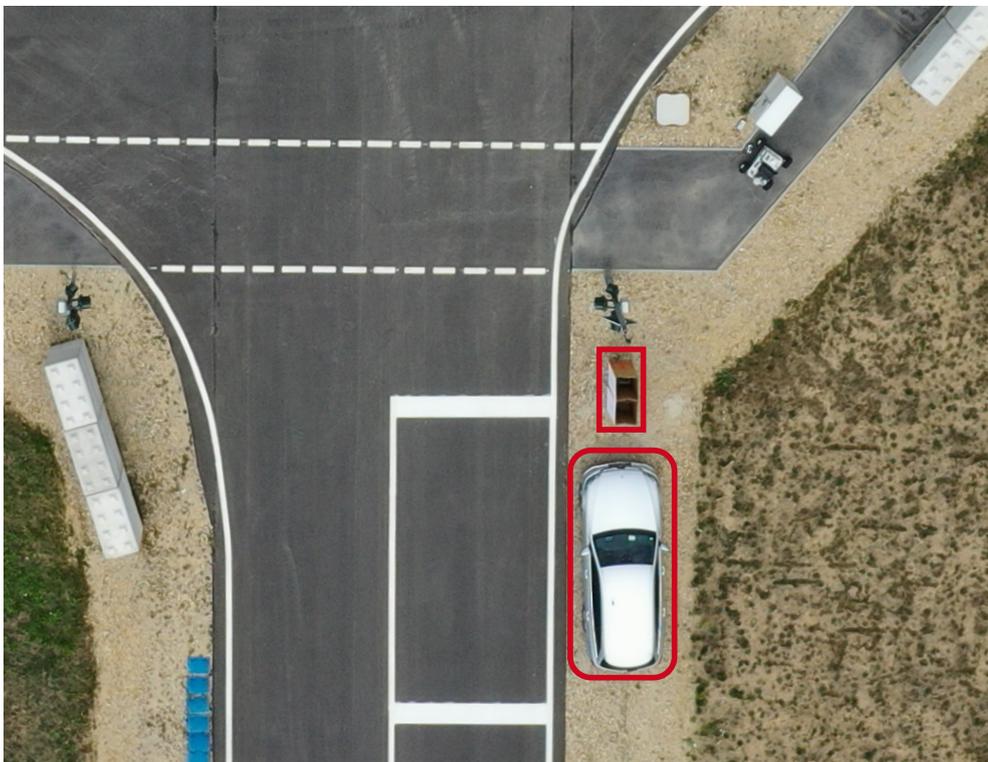


Abb. 5.13: Zu kartierende, veränderliche Objekte

Abbildung 5.14 stellt drei charakteristische Stadien der jeweils resultierenden Belegungskarte dar. Die Farbcodierung gibt den Wahrscheinlichkeitswert jeder Gitterzelle wieder; in blauer Farbe ist zudem die automatisch generierte Linienkarte dargestellt. Bild (a) zeigt das betrachtete Kartenareal, nachdem ausgehend von der initialen Belegungswahrscheinlichkeit vier Messfahrten zu Phase 1 ausgewertet wurden. Es ist erkennbar, dass

die Kontur des geparkten PKW sehr gut von Zellen hoher Belegungswahrscheinlichkeit abgebildet ist. Sowohl die der Straße zugewandte Seite des parkenden Fahrzeuges, als auch die abgewandte Seite sind in der Karte erfasst, da bei einigen Messfahrten aufgrund des weitgehend offenen Testgeländes auch ausgehend von einer Parallelstraße eine direkte optische Sichtlinie auf den PKW bestand. Die automatisch erzeugte Linienkarte besteht aus vier Liniensegmenten, welche die weitgehend rechteckige Grundform des PKW gut abbilden können. In Bild (b) ist der Zustand nach weiteren vier Durchfahrten aus Phase 2 visualisiert. Es ist erkennbar, dass auch das zusätzlich aufgestellte kastenförmige Hindernis nun in der Karte abgebildet wird. Die Rückseite wurde hier nicht erfasst, da mit keiner Messfahrt der entsprechende Sichtbereich abgedeckt wurde. Auch die Seitenwände führen zu im Vergleich zur Vorderfront deutlich geringeren Belegungswahrscheinlichkeiten, sodass die Linienextraktion hier zum gezeigten Zeitpunkt noch keine Liniensegmente erzeugt. Bild (c) stellt die Situation in Phase 3 dar; erneut wurden vier Fahrzeugdurchfahrten ausgewertet. Das kastenförmige Objekt ist nun noch deutlich ausgeprägter in der Belegungskarte abgebildet, zugleich wurden die Belegungswahrscheinlichkeiten der dem parkenden Fahrzeug zugeordneten Zellen wieder weitgehend unter die Schwelle für die Linienextraktion reduziert.

Dass die Umrisse des Fahrzeuges nicht vollkommen gleichmäßig in der Karte ausgelöscht wurden, ist auf die verschiedenen Mechanismen bei der Aktualisierung von Gitterzellen in Richtung „frei“ zurückzuführen. Liegt aus Sicht des LiDAR-Sensors hinter der nicht mehr belegten Gitterzelle ein anderes Objekt, das ein Laserecho zur Folge hat, so wird die entsprechende Zelle mit p_{frei} und $p_{frei,Reichw}$ aktualisiert. Andernfalls kommt lediglich $p_{frei,Reichw}$ innerhalb der minimalen Reichweite des Laserscanners zur Anwendung. Aufgrund der geometrischen Konstellation mit einem kartierten Objekt an der Parallelstraße hinter dem geparkten PKW erfolgt die Aktualisierung mit p_{frei} im vorliegenden Fall verstärkt im vorderen Bereich des geparkten Fahrzeuges.

Die Ergebnisse zeigen, dass der implementierte Kartierungsmechanismus in der Lage ist, die Karte den in der Umwelt auftretenden Veränderungen entsprechend zu aktualisieren. Die Geschwindigkeit der Adaption entspricht der in Kapitel 4.4.2 beschriebenen Parametrierung, bei der eine Mindestzahl an Durchfahrten für die Änderung der Zellenwahrscheinlichkeiten unter bzw. über die Schwelle zur Linienextraktion definiert wird. Durch die gute Übereinstimmung der aus verschiedenen Blickwinkeln aufgezeichneten Laserscans ergibt sich ein scharf abgegrenztes Bild mit wenig Rauschen, aus dem passende linienförmige Kartensegmente extrahiert werden können.

Genauigkeit der Kartierung Zur Quantifizierung der Genauigkeit der automatisch erstellen digitalen Karte wird in einem weiteren Testszenario ein kastenförmiges Hindernis mit bekannten Abmaßen innerhalb der Kartengrenzen platziert und mittels eines hochgenauen, geodätischen GNSS-Empfängers georeferenziert eingemessen. Anschließend wird auf Basis mehrerer Messfahrten, bei denen das unkartierte Objekt erfasst wird,

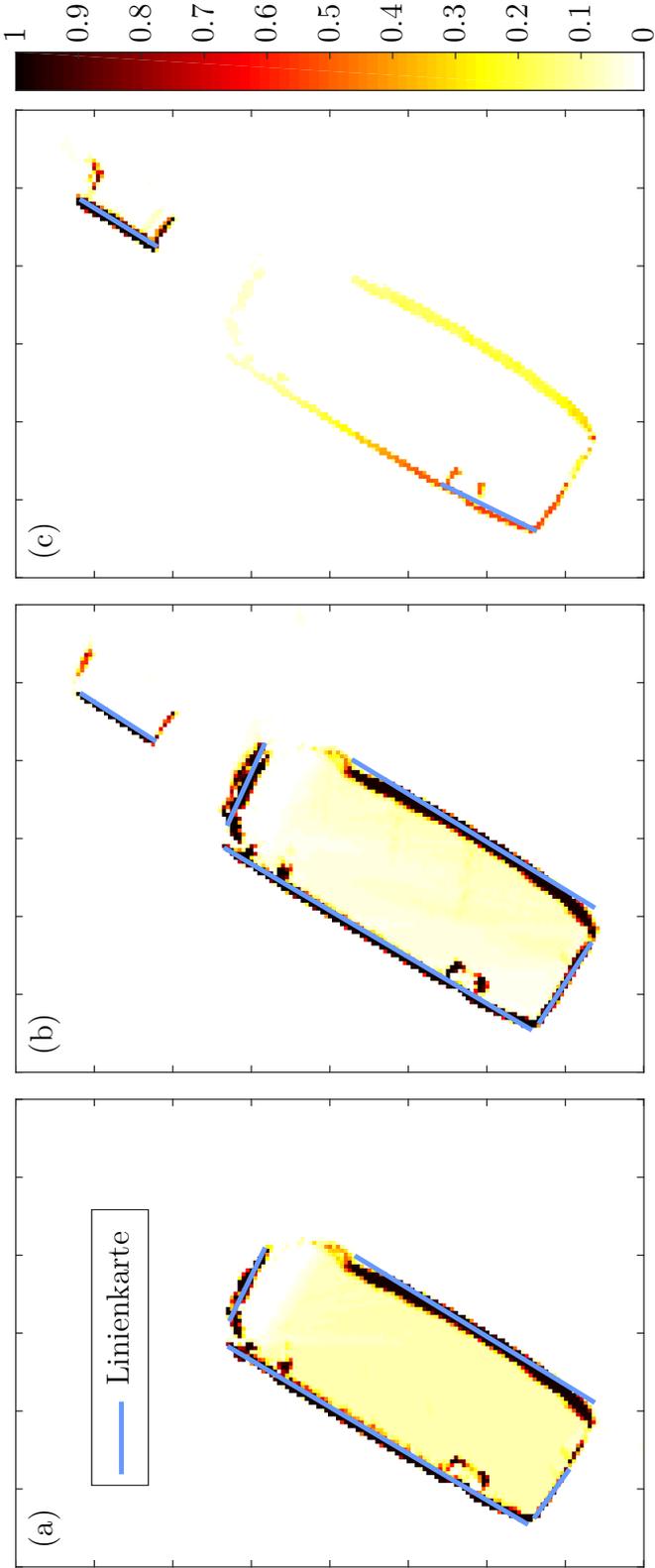


Abb. 5.14: Stadien der Belegungskarte bei veränderlichen Hindernissen

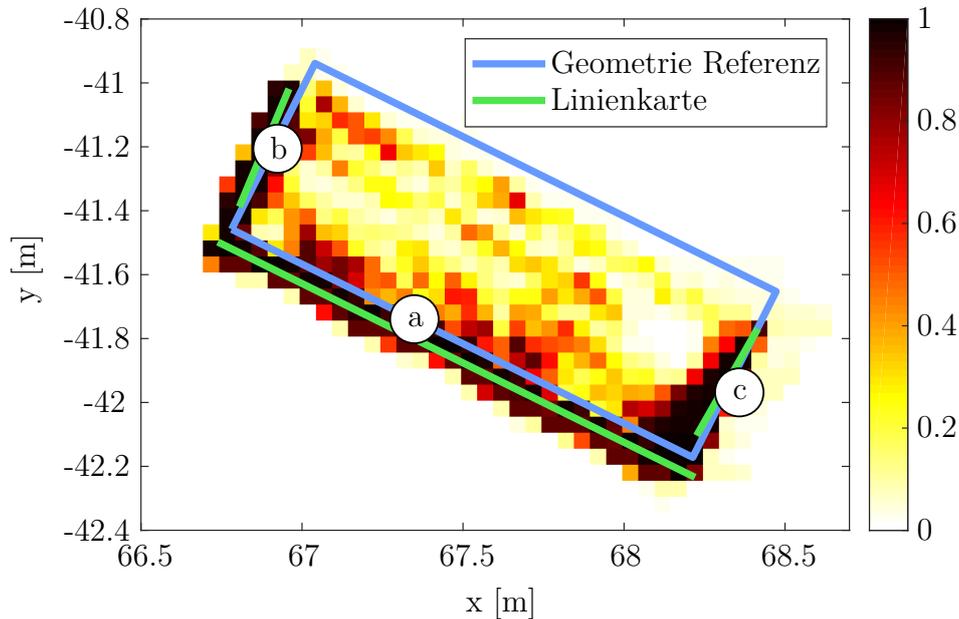


Abb. 5.15: Automatische Kartierung und Liniextraktion eines kastenförmigen Referenzobjektes

die automatische Kartenerstellung ausgeführt. Abbildung 5.15 zeigt das Ergebnis der Kartierung nach der vierten Messfahrt bei Verwendung des Scan-Matching-Algorithmus. Dargestellt sind die Wahrscheinlichkeitswerte der gitterbasierten Belegungskarte, die bekannte rechteckige Referenzgeometrie sowie die automatisch generierten Liniensegmente der digitalen Karte.

Als Maße für die Abweichungen zwischen der Linienkarte und der Referenz werden verschiedene Kenngrößen definiert. Der Winkelfehler beschreibt die Differenz zwischen dem Winkel des Segmentes der Linienkarte und der zugehörigen Kante der Referenzmessung. Mit dem Längenfehler wird der prozentuale Unterschied zwischen der realen Länge der zu kartierenden Geometrie und der Länge des erzeugten Liniensegmentes angegeben, wobei ein positiver Wert bedeutet, dass das erzeugte Liniensegment eine größere Länge als die Referenz aufweist. Der Versatz der beiden Geraden wird durch den Abstandsfehler quantifiziert. Hierfür wird der senkrechte Abstand der beiden Endpunkte des automatisch erzeugten Liniensegmentes zu der zugehörigen Referenzgeraden ermittelt. Die beiden Werte repräsentieren somit den minimalen und maximalen Abstand zwischen beiden Geraden. Die beschriebenen Kennwerte sind in Tabelle 5.1 sowohl für die Kartierung auf Basis einer Trajektorienrekonstruktion mittels Scan-Matching (bezeichnet als SM<n>), als auch für die Nutzung des ICP-Algorithmus (bezeichnet als ICP<n>) dargestellt. <n> gibt hierbei an, die wievielte Durchfahrt nach dem Aufstellen des Referenzobjektes ausge-

wertet wird. Mit (a), (b) und (c) werden die jeweiligen Kanten entsprechend Abbildung 5.15 kenntlich gemacht.

Tabelle 5.1: Kennwerte zur Genauigkeit der Kartierung eines Referenzobjektes

	SM2	SM3	SM4	ICP3	ICP4	ICP5
Winkelfehler (a) [°]	0.836	0.101	0.0657	1.41	1.95	1.96
Längenfehler (a) [%]	2.1	2.04	3.42	-28.7	-9.21	-18.9
min. Abstandsfehler (a) [cm]	8.41	7.04	5.29	13.7	15.2	15.7
max. Abstandsfehler (a) [cm]	10.8	7.33	5.48	10.8	10.28	11.3
Winkelfehler (b) [°]		9.31	3.1			
Längenfehler (b) [%]		-27.8	-33.7			
min. Abstandsfehler (b) [cm]		1.06	0.103			
max. Abstandsfehler (b) [cm]		5.81	1.98			
Winkelfehler (c) [°]		2.39	3.84			
Längenfehler (c) [%]		-26.5	-31			
min. Abstandsfehler (c) [cm]		2.49	0.992			
max. Abstandsfehler (c) [cm]		4.27	3.68			

Im Hinblick auf die Kartierung mit Scan-Matching ist erkennbar, dass die Vorderkante des Referenzobjektes nach der zweiten Durchfahrt in der Linienkarte erfasst wird. Der maximale Abstandsfehler beträgt 10.8 cm, der Winkelfehler liegt bei 0.836° und die Länge der Objektkante wird mit einer Abweichung von 2.1 % leicht überschätzt. Die folgenden Durchfahrten 3 und 4 führen zu einer kontinuierlichen Verbesserung der Lage der Objektkante. Nach der vierten Durchfahrt liegt der maximale Abstandsfehler unter 5.5 cm und es ist nahezu keine Winkelabweichung mehr vorhanden. Zugleich werden auch die beiden Seitenkanten des Objektes erfasst und kartiert, wobei hier größere Längenfehler von bis zu -33.7 % zu verzeichnen sind. Dieses Unterschätzen der Länge der seitlichen Kanten kann auf zwei Mechanismen zurückgeführt werden: Einerseits werden die Pixel, die im Zuge des RANSAC-Algorithmus bei der Linienextraktion der Vorderkante zugeordnet werden, aus dem entsprechenden Cluster entfernt, sodass sie bei der anschließenden Liniensuche der Seitenkanten nicht mehr berücksichtigt werden. Andererseits werden die aus Sicht der Straße hintersten Ecken des Referenzobjektes nur aus begrenzten Sichtbereichen vom LiDAR-Sensor erfasst, sodass die zugehörigen Gitterzellen der Belegungskarte nur langsam in Richtung des belegten Zustandes aktualisiert werden. Insgesamt ist die erzielte Genauigkeit unter Berücksichtigung der begrenzten Auflösung der zugrunde liegenden Belegungskarte von 5 cm als sehr gut zu bewerten.

Erwartungsgemäß liegen die Abweichungen bei der Kartierung mit dem ICP-Algorithmus deutlich höher. Hier wird die Vorderkante des Referenzobjektes erst nach der dritten Durchfahrt in die Linienkarte übernommen und für die seitlichen Kanten werden auch

nach der fünften Durchfahrt keine Linien detektiert. Da die Genauigkeit der rekonstruierten Trajektorie bei diesem Verfahren schlechter ist, entstehen bei der probabilistischen Kartierung mehr Widersprüche bezüglich des Belegungszustandes der relevanten Zellen. Es kommt daher vor, dass eine eigentlich belegte Zelle aufgrund einer ungenauen Ausgangspose des LiDAR-Sensors zwischenzeitlich wieder in Richtung des freien Belegungszustandes aktualisiert wird, sodass die Detektion der zugehörigen Kante verzögert wird. Eine weitere Auswirkung ist das Unterschätzen der Länge des Liniensegmentes, da Gitterzellen an den Rändern der Kante häufiger fälschlicherweise als „nicht belegt“ detektiert werden. In Anbetracht der Fehlerverteilung bei der Trajektorienrekonstruktion mittels ICP, bei der Abweichungen oberhalb von 1 m auftreten (siehe Abbildung 5.5), zeigen die Ergebnisse dennoch deutlich die Stärke des probabilistischen Verfahrens zur Aktualisierung der gitterbasierten Belegungskarte: Die Genauigkeit der detektierten Vorderkante liegt in allen Fällen unter 16 cm im Hinblick auf die Abstandsfehler und unter 2° bezüglich der Winkelabweichung. Zusätzliche Durchfahrten führen im getesteten Szenario allerdings nicht zu einer weiteren Verbesserung der Genauigkeit – die initial geschätzte Lage des Liniensegmentes verändert sich nach der dritten Durchfahrt nur noch unwesentlich.

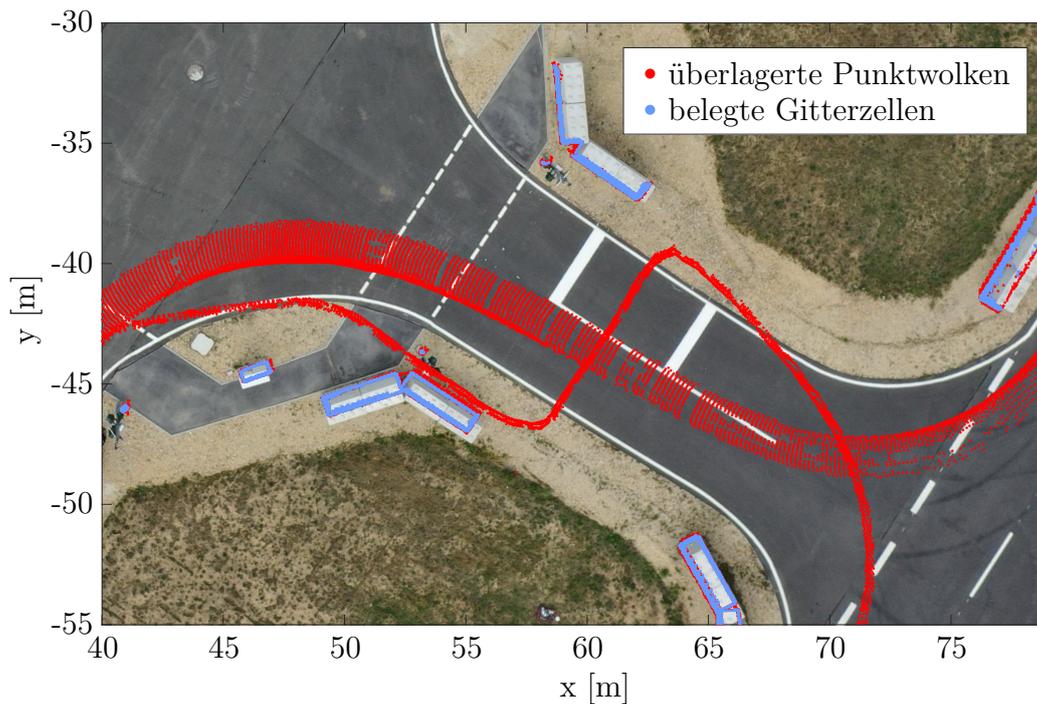


Abb. 5.16: Robustheit gegenüber bewegten Objekten bei der automatischen Kartierung

Filterung bewegter Objekte Neben der Genauigkeit der in der Karte abgebildeten Objekte besteht eine weitere wichtige Anforderung an den Kartierungsalgorithmus darin, bewegte Objekte nicht in der Karte abzubilden. Um diese Funktionalität zu prüfen, wird ein Szenario betrachtet, in dem das Messfahrzeug beim Durchqueren des kartierten Areals einem anderen PKW direkt mit geringem Abstand folgt. Desweiteren bewegt sich ein Fußgänger neben dem Messfahrzeug am Straßenrand und überquert vor dem stehenden Messfahrzeug die Straße. Abbildung 5.16 stellt eine Überlagerung aller aufgezeichneten LiDAR-Punktwolken dar, jeweils transformiert mit der entsprechenden Fahrzeugpose zum Zeitpunkt der Messung. Daneben sind die Zellen der gitterbasierten Belegungskarte hervorgehoben, die nach der Anwendung des probabilistischen Aktualisierungsalgorithmus in Richtung „belegt“ verändert wurden.

Bei den überlagerten Laserscans ist der Einfluss der bewegten Verkehrsteilnehmer sehr deutlich erkennbar. Neben den von den statischen Objekten hervorgerufenen LiDAR-Echos enthält die zusammengefasste Punktwolke auch zahlreiche Echos, die von der Rückseite des vorausfahrenden PKW erzeugt wurden. Auch der Pfad des Fußgängers mit der Überquerung der Straße vor dem Messfahrzeug ist sehr deutlich in der Punktwolke abgebildet. Nach der Aktualisierung der gitterbasierten Belegungskarte sind diese Messpunkte dagegen vollständig von den zu anderen Zeitpunkten erhaltenen „frei“-Informationen zu den selben Gitterzellen überlagert. In Richtung „belegt“ werden lediglich die Zellen aktualisiert, die statischen Objekten zuzuordnen sind und somit während der gesamten Messfahrt konsistente LiDAR-Echos erzeugen konnten. Die Auswertung belegt somit die Filterungswirkung des Kartierungsalgorithmus im Hinblick auf für die Lokalisierung ungeeignete, bewegte Objekte.

5.4.1.3 Rechenzeitbedarf

Eine wesentliche Erwägung bei der Konzeption und Parametrierung der Kartierungsalgorithmen betrifft die für die einzelnen Rechenschritte erforderliche Rechenzeit. Anhand des für die Validierungsexperimente eingesetzten Rechners mit Intel i7-6700TE Prozessor (siehe Kapitel 5.2.2) wurde der Rechenzeitbedarf der verschiedenen Teilschritte der Trajektorienrekonstruktion und der Kartenaktualisierung ermittelt. Tabelle 5.2 zeigt die so ermittelten Kennzahlen. Da die der Kartierung zugrunde liegenden Messwerte mit konstanten Datenraten aufgezeichnet werden, kann die Länge der Messung eines Datensatzes als wesentlicher Einflussfaktor auf die erforderliche Rechenzeit vorausgesetzt werden. Die dargestellten Kennwerte sind daher auf die Messzeit normiert, sodass sich ein Echtzeitfaktor f_t angeben lässt, der aus der Messdauer t_m des Datensatzes und der Rechenzeit t_r für den entsprechenden Verarbeitungsschritt gemäß $f_t = \frac{t_m}{t_r}$ berechnet wird.

Der erste Verarbeitungsschritt bei der Trajektorienrekonstruktion ist die Ausführung des GraphSLAM-Frontends, also die Erzeugung der Knoten und Kanten des zu op-

Tabelle 5.2: Gemittelter Rechenzeitbedarf Kartierung

KenngroÙe	Wert
Echtzeitfaktor GraphSLAM-Frontend ohne LiDAR	8620
Echtzeitfaktor GraphSLAM-Optimierung	399
Echtzeitfaktor Ausrichtung Laserscans mit Scan-Matching	1.81
Faktor Scan-Matching bei Verdoppelung des Winkel-Suchbereiches	1.82
Faktor Scan-Matching bei Verdoppelung des xy-Suchbereiches	3.25
Echtzeitfaktor Ausrichtung Laserscans mit ICP bei 1. Optimierung	18
Echtzeitfaktor Ausrichtung Laserscans mit ICP bei 2. Optimierung	29.9
Echtzeitfaktor Aktualisierung Belegungskarte	4.69
Faktor Aktualisierung Belegungskarte bei Halbierung der Zellenbreite	2.74
Echtzeitfaktor Linienextraktion	658

timierenden Graphen aus den Messdaten. Im ersten Schritt werden hierbei nur die Daten von Odometrie, IMU und GNSS verarbeitet, LiDAR-Daten werden noch nicht berücksichtigt. Es ist erkennbar, dass dieser Schritt einen nur unwesentlichen Anteil am gesamten Rechenaufwand einnimmt: Die Rechenzeit für eine typische Messfahrt im betrachteten Testfeld liegt bei 6.7 ms. Im Vergleich dazu steigt der Rechenaufwand für die Optimierung des Graphen um mehr als den Faktor 20. Mit einem Echtzeitfaktor von 399 ist der Rechenaufwand allerdings immer noch sehr gering – die Optimierung des Graphen einer 40 s langen Messfahrt benötigt damit etwa 0.1 s. Da diese Optimierung im Rahmen der Trajektorienrekonstruktion bis zu 3 mal durchgeführt wird, wobei unterschiedliche Konfigurationen an GNSS und LiDAR-Knoten und -Kanten enthalten sind, müsste die Auswertung zum Zeitaufwand des Optimierungsschrittes eigentlich für die unterschiedlichen Fälle getrennt vorgenommen werden. In der Praxis konnte jedoch kein signifikanter Unterschied zwischen den verschiedenen Optimierungsstufen festgestellt werden, da die Anzahl an Knoten und Kanten der immer enthaltenen Odometrie- und IMU-Messungen den mit Abstand größten Anteil am Gesamtgraphen ausmacht.

Ein großer Anteil des Rechenaufwandes entfällt auf die Verarbeitung der Laserscans mit dem Ziel, die entsprechenden Transformationen als Knoten und Kanten in den Graphen zu integrieren. Wie Tabelle 5.2 zeigt, ist insbesondere die Nutzung des Scan-Matching-Verfahrens sehr rechenintensiv: Werden LiDAR-Messungen mit einer Rate von 20 Hz bereitgestellt, so liegt der Echtzeitfaktor für das Scan-Matching lediglich bei 1.81. Das ICP-Verfahren erweist sich als im Vergleich etwa 10 mal so schnell, wobei die Ausrichtung der Punktwolken beim zweiten ICP-Durchlauf aufgrund der besseren initialen Ausrichtung und damit schnelleren Konvergenz noch weniger rechenaufwendig ist. Für das Scan-Matching kann erwartungsgemäß eine große Abhängigkeit der Rechenzeit von der Definition des Suchbereiches ausgemacht werden. Aufgrund des in den drei Koordinaten (x, y, φ) stets vollständig abgesuchten Suchraumes ist eine Abhängigkeit

der Rechenzeit von der Größe des Suchraumes mit der 3. Potenz zu erwarten. In der Praxis zeigt sich für die untersuchten Szenarien eine etwas geringere Abhängigkeit, da Teile des Zeitaufwandes – etwa für die initiale Verarbeitung der Eingangsdaten – von der Suchraumgröße unabhängig sind. Der Rechenzeitauswertung liegt die in Tabelle 4.4 dargestellte Parametrierung mit einem Suchradius von 8° bzw. 8m zugrunde, die für die überwiegende Mehrzahl der Messfahrten zu sehr guten Ergebnissen führt.

Mit einem Echtzeitfaktor von 4.69 macht auch die Aktualisierung der gitterbasierten Belegungskarte einen wesentlichen Anteil an der für die Kartierung erforderlichen Rechenzeit aus. Da die Anzahl der zu aktualisierenden Gitterzellen aufgrund der zweidimensionalen Kartenstruktur quadratisch mit der Erhöhung der Gitterauflösung steigt, hat dieser Parameter den größten Einfluss auf den Rechenzeitbedarf für die Aktualisierung der Zellenwahrscheinlichkeiten. Aufgrund zusätzlicher von der Auflösung weitgehend unabhängiger Rechenschritte liegt der reale Zeitfaktor bei Verdoppelung der Auflösung lediglich bei 2.74. Trotzdem kann mit der Anpassung der Gitterauflösung somit stark auf die Rechenzeit Einfluss genommen werden. Die auf RANSAC basierende Linienextraktion ist demgegenüber um zwei Größenordnungen schneller; typische Verarbeitungszeiten liegen hier für die untersuchten Szenarien bei 0.03 - 0.04 s.

5.4.1.4 Fazit Kartierung

Zusammenfassend kann festgestellt werden, dass die Trajektorienrekonstruktion als erster Verarbeitungsschritt der Kartierung die gewünschte Funktionalität bereitstellt. Durch den Abgleich der LiDAR-Daten mit der statischen Referenzkarte wird im regulären Betrieb eine Genauigkeit der georeferenzierten Trajektorie im Zentimeterbereich erzielt, obwohl nur ein konventioneller GNSS-Empfänger mit großer Lokalisierungsunsicherheit genutzt wird. Die mehrstufige Optimierung mittels GraphSLAM gewährleistet die bestmögliche Ausnutzung aller zur Verfügung stehenden Sensordaten; es ergibt sich eine glatte und hoch aufgelöste Trajektorie in den drei Koordinaten (x, y, φ) .

Im genutzten Testumfeld kann mittels der Scan-Matching-Methode im Vergleich zum ICP-Algorithmus eine deutlich präzisere Rekonstruktion der Trajektorie realisiert werden. Zum Teil ist dies auf Symmetrieeffekte aufgrund der speziellen Geometrie der kartierten Hindernisse zurückzuführen, sodass in Arealen mit stärkerer oder weniger gleichförmiger Bebauung mit einem weniger großen Unterschied zu rechnen ist. Zugleich ist festzustellen, dass ICP eine gute Robustheit gegenüber größeren initialen Abweichungen aufweist und – insbesondere mit dem zweistufigen Optimierungsschema – eine schnelle und gute Konvergenz realisiert. Grundsätzlich können für die vorgesehene Anwendung also beide Verfahren sinnvoll eingesetzt werden. Steht ausreichend Rechenkapazität zur Verfügung, so kann mittels Scan-Matching im Allgemeinen eine höhere Genauigkeit der Trajektorienrekonstruktion erreicht werden; andernfalls können mittels ICP die bereitgestellten Daten schnell für die Kartierung nutzbar gemacht werden. Großes Optimierungspotenzial

besteht zudem bei der Parametrierung des Scan-Matching-Verfahrens. Stehen Informationen über die Verlässlichkeit der initialen GNSS-Daten zur Verfügung, so kann der Suchraum gegebenenfalls deutlich verkleinert werden, sodass aufgrund der Abhängigkeit der Rechenzeit mit der 3. Potenz eine erhebliche Beschleunigung des Matchings erzielbar ist.

Die Ergebnisse der Aktualisierung der gitterbasierten Belegungskarte zeigen ebenfalls, dass das implementierte System die in Kapitel 4.1 definierten Anforderungen erfüllen kann. Die Karte reagiert innerhalb des durch die Parametrierung vorgegebenen zeitlichen Rahmens auf Änderungen in der Umwelt, es werden zusätzliche Objekte kartiert und nicht mehr vorhandene Objekte entfernt. In Bewegung befindliche Verkehrsteilnehmer werden durch das Verfahren vollständig aus den Daten herausgefiltert. Die im Rahmen der Testmessungen eingesetzten Hindernisse können durch die Liniensegmente der automatisch generierten Linienkarte gut abgebildet werden. Bei Einsatz des Scan-Matching-Verfahrens für die Trajektorienrekonstruktion liegen die Abweichungen zwischen Karte und Realität unter 5.5 cm. Mit dem deutlich schnelleren ICP-Verfahren liegen die Fehler unter 16 cm. Bezüglich ungenauer Messdaten sowie bezüglich Fehlern bei der Trajektorienrekonstruktion zeigt sich das Kartierungsverfahren robust. Widersprüchliche Informationen führen tendenziell dazu, dass die Belegungswahrscheinlichkeiten in der Belegungskarte sinken, sodass zwar die Erfassung bislang unkartierter Objekte verzögert wird, die Wahrscheinlichkeit für falsche Liniensegmente in der finalen Linienkarte jedoch gering ist.

5.4.2 Lokalisierung

Um quantitative Aussagen über die Güte der Lokalisierung treffen zu können, werden erneut Positionsdaten des hochgenauen GNSS-Empfängers als Referenz genutzt. Anders als bei der Auswertung der Trajektorienrekonstruktion (siehe Kapitel 5.4.1.1) ist es hier allerdings nicht sinnvoll, die Lokalisierungsgenauigkeit aus allen Einzelfehlern zwischen der Lokalisierungslösung und der Referenzposition abzuleiten, da in diesem Fall die Konvergenzphase des Lokisierungsalgorithmus in den ermittelten Fehler einfließen würde. Zur Verdeutlichung ist in Abbildung 5.17 das Lokalisierungsergebnis einer typischen Testfahrt im Vergleich zum konventionellen GNSS-Empfänger und zum Referenzsystem dargestellt. Für eine bessere Erkennbarkeit ist aus dem Referenzdatensatz nur jeder fünfte Datenpunkt eingezeichnet. Abbildung 5.18 stellt für die selbe Messung den Verlauf des euklidischen Abstandsfehlers e_{xy} über der Messzeit t dar. Die Abbildungen zeigen, dass bei der zugrunde liegenden Testfahrt die Initialisierung der Startposition auf Basis des konventionellen GNSS-Empfängers einen Fehler von etwa 4 m aufweist. Dieser initiale Fehler reduziert sich innerhalb der ersten Sekunden deutlich; im weiteren Verlauf stellen sich nur noch langsamere Änderungen ein und der Fehler verbleibt auf einem niedrigeren Niveau.

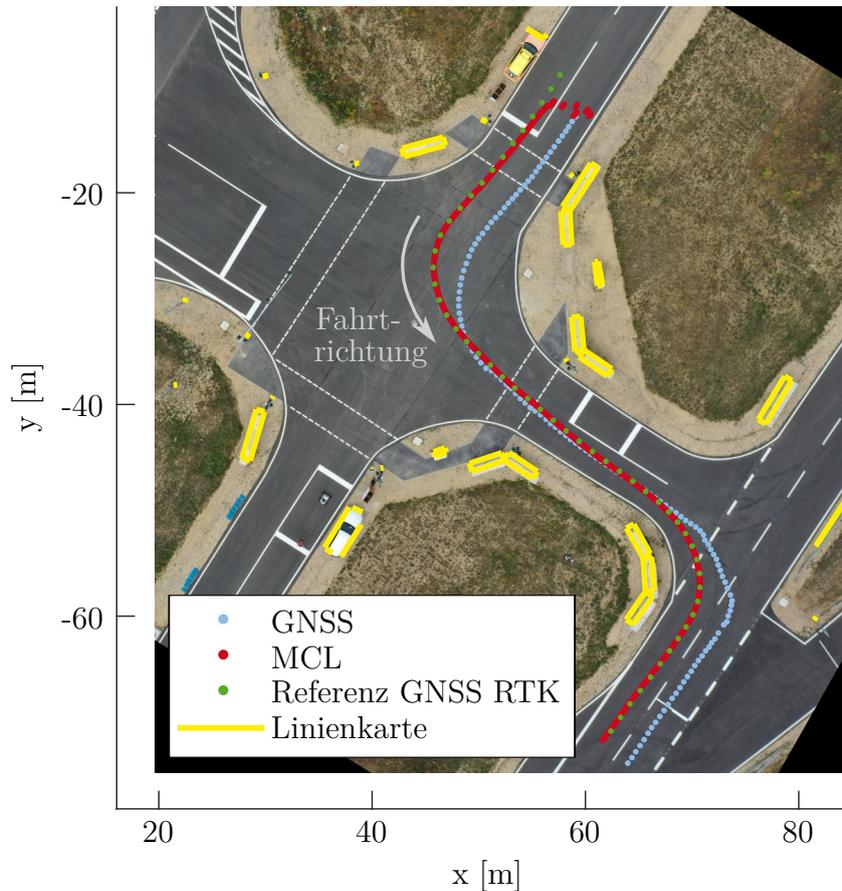


Abb. 5.17: Fahrzeugtrajektorie im Vergleich zu konventionellem GNSS und Referenz bei kartengestützter Lokalisierung

Das gezeigte Verhalten ist typisch für alle ausgewerteten Messfahrten und kann in zwei Phasen unterteilt werden. Die **Konvergenzphase** ist von der Kompression der initialen Partikelwolke auf Basis der Übereinstimmung von LiDAR und digitaler Karte geprägt. In der anschließenden **Tracking-Phase** hat die Partikelwolke im Allgemeinen eine kompakte Form angenommen und folgt der Fahrzeugpose. Für die Güte des Lokalisierungsverfahrens ist einerseits von Interesse, wie hoch die Genauigkeit der Lokalisierung in der Tracking-Phase ist. Die Höhe des initialen Fehlers bzw. die in der Konvergenzphase vorliegenden Fehler sollen hierbei nicht berücksichtigt werden, da das Lokalisierungsverfahren auf die Qualität der Initialisierung keinen Einfluss hat. Andererseits ist zu bestimmen, wie viel Zeit die Konvergenzphase in Anspruch nimmt. Wünschenswert ist eine schnelle Konvergenz und damit eine kurze Zeitspanne bis zum Erreichen einer genauen Positionsbestimmung.

Aus diesen Gründen wird bei der Auswertung der Testfahrten im Hinblick auf die Fahrzeuglokalisierung eine automatische Erkennung der Konvergenz- und Tracking-Phase

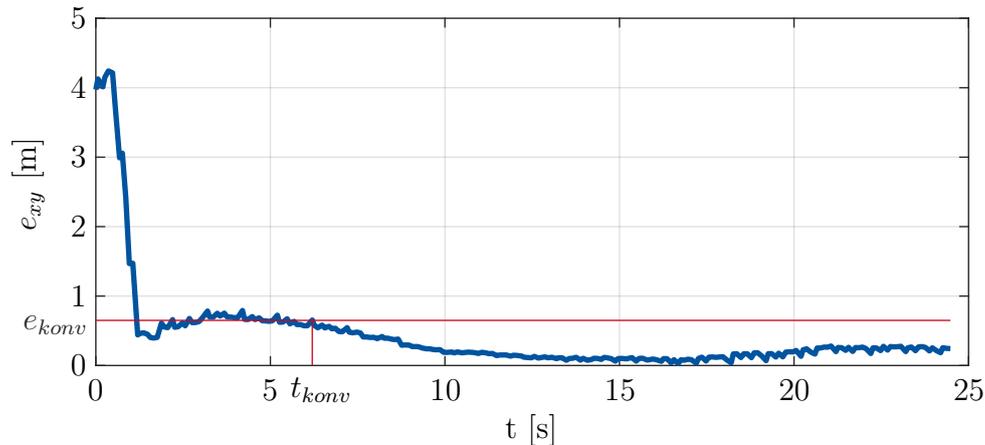


Abb. 5.18: Beispielhafter Verlauf des euklidischen Abstandsfehlers bei der kartengestützten Lokalisierung und Konvergenzkriterium

auf Basis eines empirisch ermittelten Grenzwertes vorgenommen. Der gesuchte Zeitpunkt t_{konv} , zu dem die Lokalisierung als „konvergiert“ betrachtet wird, ist demnach der früheste Zeitpunkt, nach dem der euklidische Abstandsfehler die Fehlerschranke e_{konv} nicht wieder überschreitet. Die Wahl von e_{konv} beeinflusst damit wesentlich die im Rahmen der weiteren Auswertung ermittelten Kennwerte: Ein großes e_{konv} führt tendenziell zur Ermittlung kurzer Konvergenzzeiten; zugleich steigt aber mit großem e_{konv} der durchschnittliche Tracking-Fehler. Umgekehrt kann ein zu kleines e_{konv} dazu führen, dass aufgrund von Ausreißern innerhalb der Tracking-Phase keine Konvergenz detektiert wird. Mit $e_{konv} = 0.65$ m wird in der vorliegenden Arbeit die kleinste mögliche Schranke gewählt, bei der für die überwiegende Mehrzahl der Messungen ein sinnvoller Konvergenzzeitpunkt ermittelt werden kann. Diejenigen Testfälle, bei denen nach diesem Kriterium keine Konvergenz detektiert werden kann, werden gesondert analysiert.

Eine weitere Besonderheit bei der Auswertung der Lokalisierung besteht darin, dass innerhalb des Lokalisierungsalgorithmus Zufallszahlen zum Einsatz kommen, sodass sich ein nicht-deterministisches Ergebnis ergibt. Die ermittelte Fahrzeugpose kann variieren, je nachdem, welche Partikel beim Resampling bzw. welche Fehler aus den angenommenen Wahrscheinlichkeitsverteilungen gezogen werden. Im Rahmen der Auswertung wird daher unter Verwendung des in Kapitel 5.3 beschriebenen Echtzeit-Simulationssystems jede Messfahrt fünf Mal simuliert und eine anschließende Mittelung der Ergebnisse vorgenommen. Abbildung 5.19 stellt beispielhaft die sich ergebenden Fehlerverläufe von fünf Auswertungen einer Testfahrt dar. Es ist erkennbar, dass die qualitativen Verläufe aller Messungen ähnlich sind. Trotzdem ergibt die quantitative Auswertung ein Verhältnis von 1.2 zwischen den mittleren Tracking-Fehlern der besten und schlechtesten Messung; bei den Konvergenzzeiten beträgt das Verhältnis 1.48. Mit der Mittelwertbildung aus

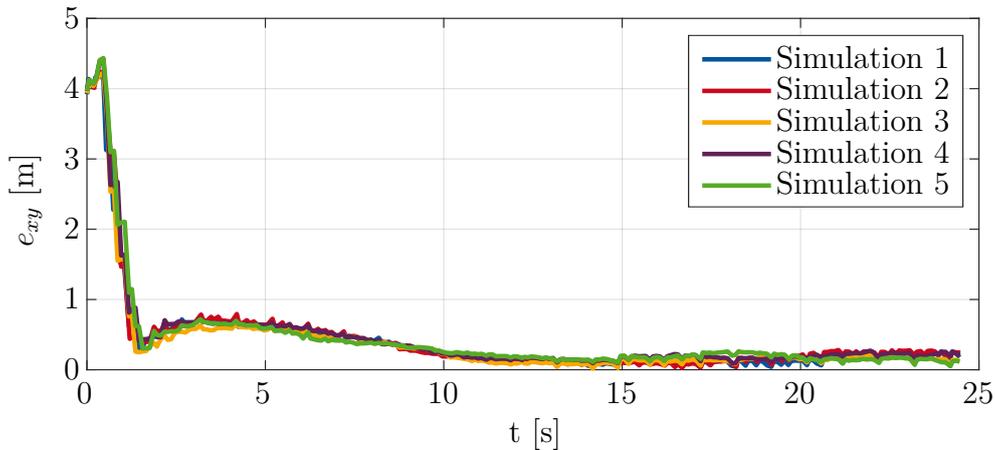


Abb. 5.19: Verläufe der euklidischen Abstandsfehler bei mehreren Simulationen mit identischen Startbedingungen

mehreren Simulationen kann somit die Aussagekraft der Testergebnisse signifikant erhöht werden.

5.4.2.1 Genauigkeit Position

Zur Ermittlung der Positionsgenauigkeit bei der kartengestützten Fahrzeuglokalisierung werden die Ergebnisse von 17 verschiedenen Testfahrten durch den kartierten Bereich genutzt. Es wird die GNSS-basierte Initialisierung des Lokalisierungsalgorithmus verwendet. Datensätze, bei denen aufgrund der künstlichen Abschirmung der GNSS-Antenne keine Initialisierung möglich ist, werden daher nicht betrachtet. Mit fünf Simulationsläufen pro Testfahrt ergeben sich somit 85 Datensätze, die entsprechend der zuvor dargestellten Methodik ausgewertet werden. Jede Testfahrt deckt eine vollständige Durchfahrt durch den kartierten Bereich ab.

Bei der automatischen Auswertung mit der Fehlerschranke $e_{konv} = 0.65$ m wird in 81 von 85 Fällen eine erfolgreiche Konvergenz detektiert. Für diese Fälle können dementsprechend ein mittlerer Tracking-Fehler e_{xy} und eine Konvergenzzeit t_{konv} ermittelt werden. Die vier nicht als konvergiert erkannten Fälle werden gesondert betrachtet.

Abbildung 5.20 zeigt die Häufigkeitsverteilung der euklidischen Abstandsfehler in der Tracking-Phase für alle konvergierten Messfahrten. Es ist erkennbar, dass die häufigsten Positionsfehler im Bereich zwischen 8 und 12 cm liegen. Aufgrund der Festlegung der Fehlerschranke auf 0.65 m können in der gezeigten Auswertung keine Einzelfehler existieren, die diesen Wert überschreiten. Tatsächlich weist die Häufigkeit der Fehler in der Nähe der Fehlerschranke bereits sehr geringe Werte auf, sodass deutlich wird, dass sich

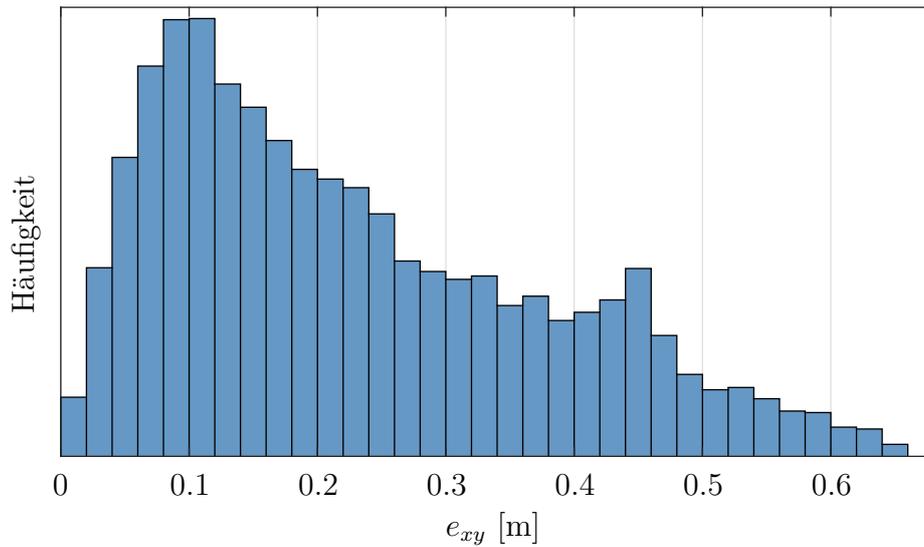


Abb. 5.20: Histogramm der euklidischen Tracking-Fehler bei der Echtzeitlokalisierung

in den meisten Fällen die Genauigkeit der Lokalisierung nach der Unterschreitung der Fehlerschranke sehr schnell in Richtung deutlich kleinerer Fehler verbessert.

Der Vergleich der Referenzposition mit der von der MCL bestimmten Position in den ENU-Koordinaten x und y in Abbildung 5.21 verdeutlicht anhand einer typischen Testfahrt das Arbeitsprinzip des parallelisierten Lokalisierungsalgorithmus. Insbesondere in der Konvergenzphase sind kurze Sequenzen von gleichmäßig und glatt aufeinander folgenden Messpunkten erkennbar, die von deutlichen Sprüngen unterbrochen werden. Hervorgerufen wird dieses Verhalten durch die mit unterschiedlicher Häufigkeit ausgeführten Verarbeitungsschritte des Partikelfilters (siehe Kapitel 3.4.3). Die mit fester Rate von 50 Hz getaktete modellbasierte Prädiktion des Fahrzeugzustandes greift auf die Odometrie- und IMU-Daten zu und ermittelt in jedem Zeitschritt die geschätzte Pose auf Basis des Zustandes im vorherigen Zeitschritt. Da auch die hierbei addierten Unsicherheiten als mittelwertfreies, Gaußsches Rauschen modelliert sind, ergibt sich durch diesen Mechanismus bei Mittelung über alle Partikelposen der dargestellte, glatte Verlauf. Jedes Messupdate zieht dagegen eine Gewichtung aller Partikel und ein anschließendes Resampling nach sich, bei dem die Partikel auf Basis der Übereinstimmung von LiDAR-Scan und digitaler Karte entfernt oder vervielfältigt werden. Dass dieser Verarbeitungsschritt zu einer sprunghaftigen Konzentration der Partikelwolke und damit zu einer Konvergenz in Richtung der wahren Lösung führt, ist anhand von Abbildung 5.21 deutlich erkennbar. In der Tracking-Phase sind dagegen keine diskreten Sprünge mehr erkennbar – hier ergibt sich eine gute Übereinstimmung zwischen der durch das Prädiktionsmodell ermittelten Posenhypothese und der Gewichtung der Partikel durch das Messupdate. Der Verlauf ist damit in dieser Phase weitestgehend glatt.

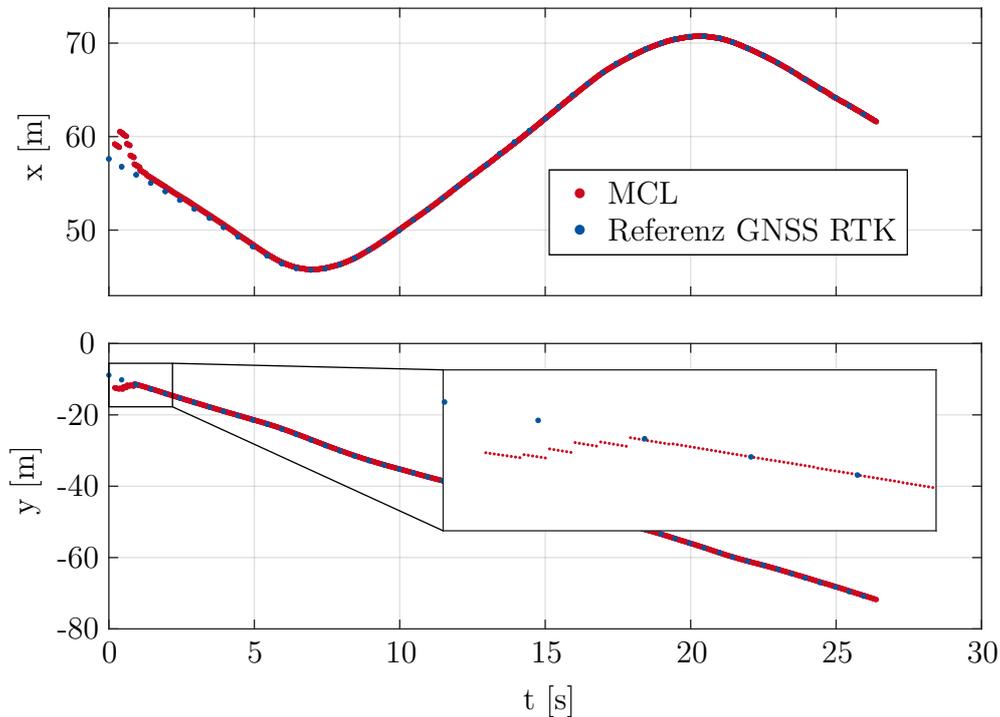


Abb. 5.21: Positionsschätzung der Monte-Carlo-Lokalisierung im Vergleich zur Referenz

5.4.2.2 Genauigkeit Orientierung

Wie bereits in Kapitel 5.4.1.1 bei der Auswertung der Trajektorienrekonstruktion dargestellt, existiert für die quantitative Bewertung der Lokalisierungsfehler im Hinblick auf die Orientierung kein zuverlässiges Referenzsignal, da der RTK-fähige GNSS-Empfänger lediglich Positionsinformationen liefern kann. Die hieraus mittels symmetrischer Differenzen ermittelten Orientierungsinformationen weisen ein deutliches Rauschen auf. Trotzdem kann anhand dieser Referenzdaten die Funktion der Orientierungsschätzung mittels der MCL qualitativ plausibilisiert werden.

Abbildung 5.22 zeigt die von der Echtzeitlokalisierung ermittelte Orientierung im Vergleich zur Referenz. Es ist erkennbar, dass nach der Konvergenzphase keine systematischen Abweichungen zwischen beiden Signalen auftreten. Die in Echtzeit geschätzte Orientierung entspricht mit sehr guter Übereinstimmung dem lokalen Mittelwert des Referenzsignals und weist einen glatten Verlauf auf. Im Hinblick auf die prinzipielle Funktion kann somit festgestellt werden, dass auch die Orientierungsschätzung des Lokalisierungssystems eine hohe Präzision und Auflösung erreicht.

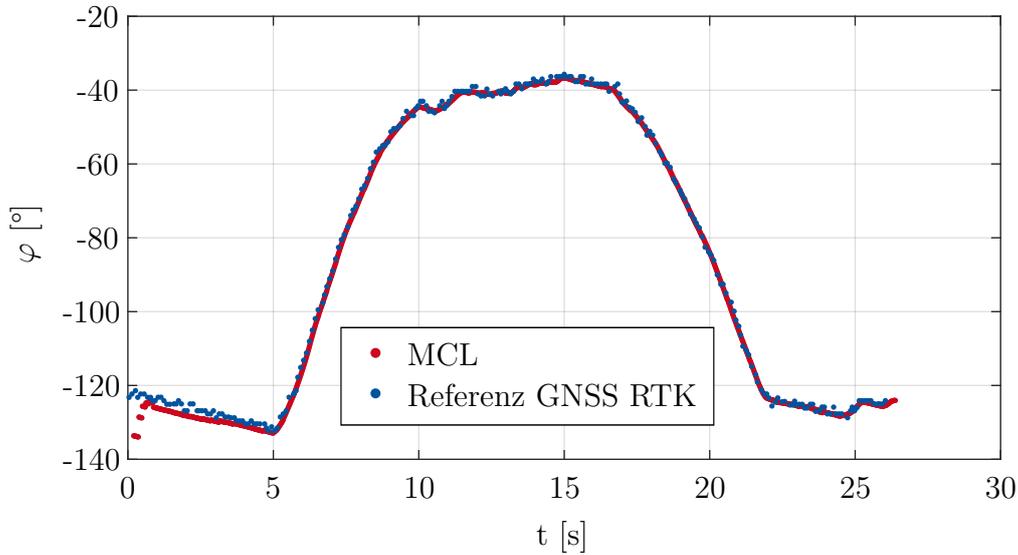


Abb. 5.22: Orientierungsschätzung der Monte-Carlo-Lokalisierung im Vergleich zur Referenz

5.4.2.3 Konvergenzgeschwindigkeit

Neben der Genauigkeit der Lokalisierung in der Tracking-Phase ist für die reale Anwendung des Lokalisierungsalgorithmus von großer Bedeutung, wie schnell eine zuverlässige Schätzung der Fahrzeugpose zur Verfügung steht. Anhand des zur Bestimmung der Positionsgenauigkeit identischen Datensatzes werden daher die Konvergenzzeiten mit der definierten Fehlerschranke $e_{konv} = 0.65$ m automatisch ermittelt.

Abbildung 5.23 zeigt das sich ergebende Histogramm der Konvergenzzeiten. Es ist erkennbar, dass bei der gewählten Parametrierung des Algorithmus die größte Häufigkeit im Intervall $t_{konv} \in [0\text{ s}, 2\text{ s}]$ zu verzeichnen ist. In 84 % der Fälle erfolgt die Konvergenz in weniger als 10 s, 94 % liegen unter 14 s. Ausreißer bei den Konvergenzzeiten weisen allerdings Werte von bis zu 24 s auf. Zu berücksichtigen ist bei dieser Auswertung, dass die Konvergenzzeiten wesentlich vom Initialisierungsfehler der anhand des konventionellen GNSS-Empfängers bestimmten Startposition abhängen. Zwar wird auch bei initialen Positionsfehlern in der Größenordnung von 10 m in den meisten Testfällen noch eine Konvergenz erzielt. Die hierfür benötigte Zeit ist dann aufgrund der für bessere Robustheit und Genauigkeit beim Tracking erforderlichen Trägheit des Partikelfilters aber entsprechend höher. Mittels der Parametrierung des Lokalisierungssystems kann also je nach zur Verfügung stehender initialer Positionsschätzung ein passender Kompromiss zwischen schneller Konvergenz auch bei hoher initialer Unsicherheit und genauer und gut gedämpfter Positionsschätzung in der Tracking-Phase getroffen werden.

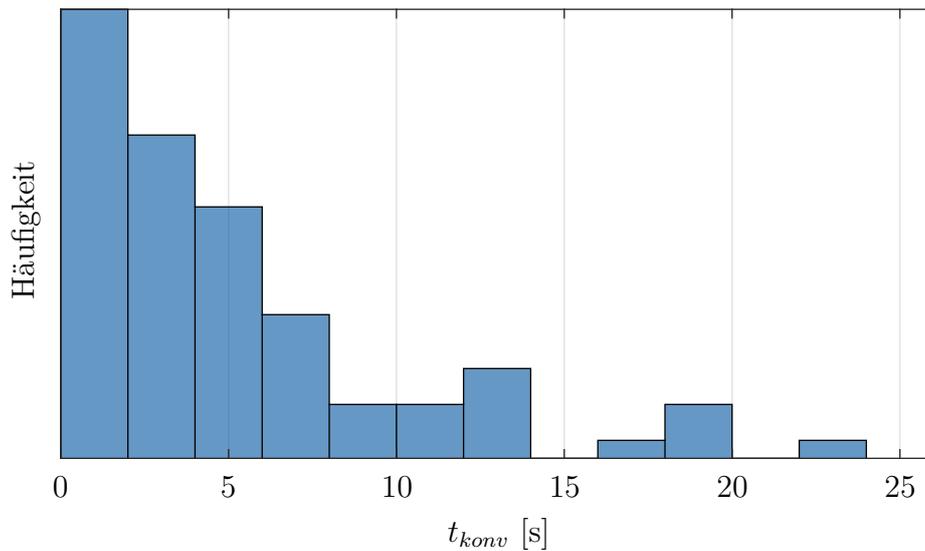


Abb. 5.23: Histogramm der Konvergenzzeiten bei der Echtzeitlokalisierung

5.4.2.4 Auswirkungen der Kartenaktualisierung auf die Lokalisierung

Die kontinuierliche Aktualisierung der digitalen Karte verfolgt das Ziel, dem Lokalisierungsalgorithmus stets eine möglichst vollständige und korrekte Karte zur Verfügung zu stellen. Es ist zu erwarten, dass die Genauigkeit und Robustheit der Lokalisierung auf diese Weise gesteigert werden kann. Zur Verifikation dieser Annahme werden verschiedene Szenarien betrachtet, in denen Abweichungen zwischen dem von der Fahrzeugsensorik wahrgenommenen Verkehrsumfeld und der digitalen Karte vorliegen. Es werden die Lokalisierungsergebnisse bei unvollständiger Karte mit Referenzergebnissen verglichen, bei denen die Karte das Umfeld vollständig abbildet.

Abbildung 5.24 zeigt einen Überblick über die drei getesteten Szenarien, wobei jeweils die vollständige und die unvollständige Karte im Vergleich dargestellt sind. In Szenario (a) ist ein zusätzliches kastenförmiges Objekt im kartierten Bereich platziert, das eine Größe von $1.58 \text{ m} \times 0.56 \text{ m}$ aufweist. Szenario (b) betrachtet den Einfluss der weitgehenden Verdeckung einer Kulissenwand durch nicht kartierte Hindernisse. Die Anordnung der Objekte entspricht Abbildung 5.11 aus Kapitel 5.4.1.1. In Szenario (c) wurde die statische Referenzkarte künstlich ausgedünnt, sodass nur noch etwa die Hälfte der in der Umgebung vorhandenen geometrischen Objekte in der Karte enthalten sind.

Zur Quantifizierung der Auswirkungen einer unvollständigen digitalen Karte auf die Lokalisierung werden für jedes Szenario mehrere Testfahrten ausgewertet. Wie bei den vorangegangenen Auswertungen wird auch hier jede Testfahrt 5 Mal simuliert, um eine Mittelung der stochastischen Ergebnisse vornehmen zu können. Als Gesamtmaß für

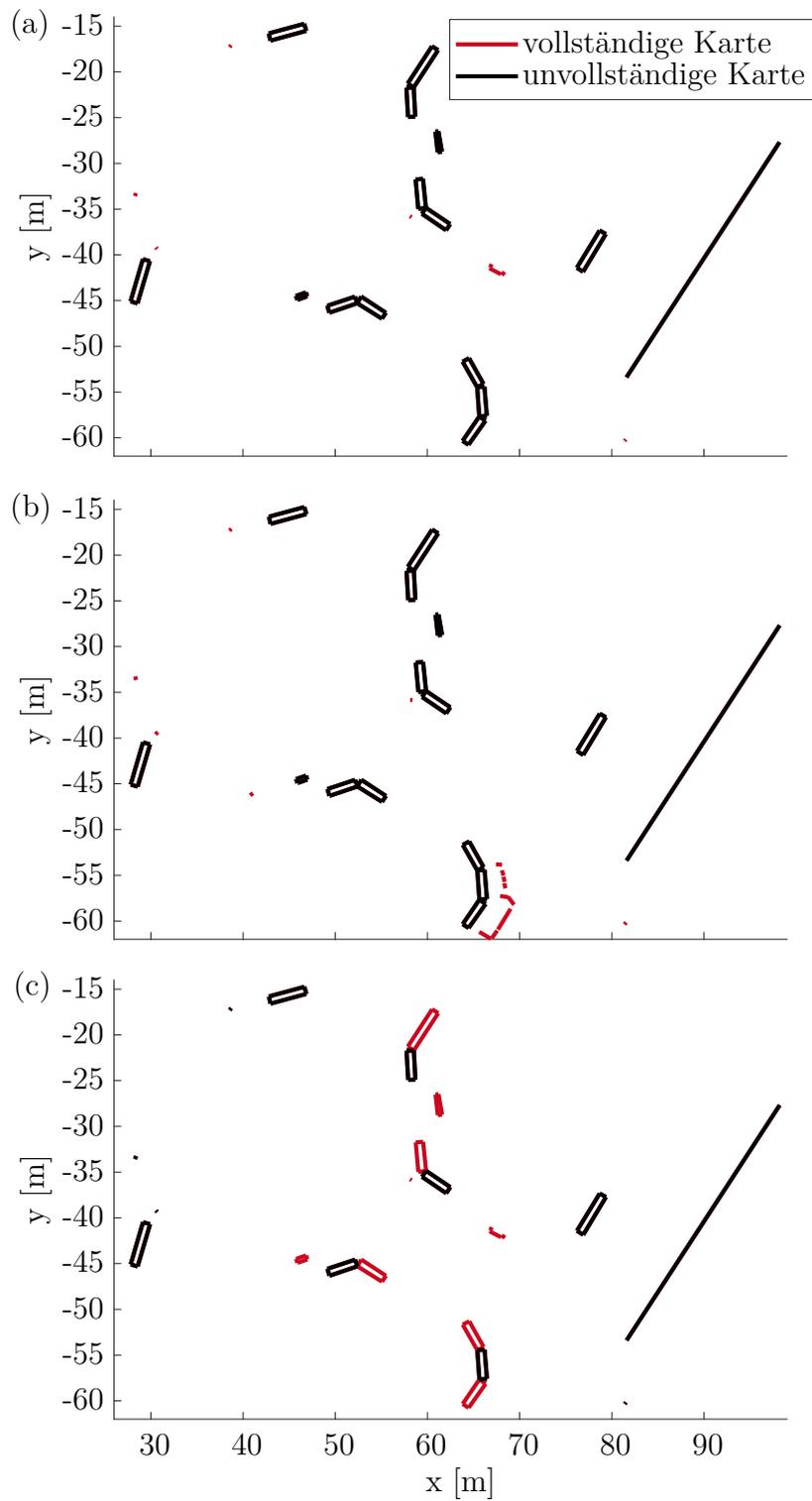


Abb. 5.24: Testszzenarien mit unvollständiger digitaler Karte

Tabelle 5.3: Änderung der mittleren Positionsfehler der Lokalisierung bei unvollständiger Karte

Szenario	Fehler vollst. Karte	Fehler unvollst. Karte	Änderung
(a)	0.3902 m	0.393 m	+0.707 %
(b)	0.444 m	1.132 m	+155 %
(c)	0.2339 m	0.3605 m	+54.1 %

die Genauigkeit der Lokalisierung, das sowohl die Konvergenz-, als auch die Tracking-Phase berücksichtigt, wird der mittlere Lokalisierungsfehler über die gesamte Messfahrt bestimmt. Tabelle 5.3 zeigt eine Übersicht der so ermittelten Fehler sowie den prozentualen Einfluss auf die Lokalisierungsgenauigkeit durch die fehlenden Kartenelemente. Die Fehler bei vollständiger Karte unterscheiden sich je nach Szenario aufgrund der unterschiedlichen Testfahrten, die in den jeweiligen Konfigurationen durchgeführt wurden.

Die Tabelle zeigt, dass sich bei den unterschiedlichen Szenarien sehr variable Einflüsse der unvollständigen Karte ergeben. Die Kartierung des relativ frei stehenden zusätzlichen Objekts in Szenario (a) hat keinen signifikanten Einfluss auf die Genauigkeit der Lokalisierung. Die Fehler liegen auf einem identischen Niveau unabhängig davon, ob das zusätzliche Objekt kartiert ist, oder nicht.

In Szenario (b) ist die Kartierung des parkenden PKW sowie des kastenförmigen Hindernisses dagegen für eine genaue Lokalisierung unbedingt notwendig. Fehlen diese Elemente in der Karte, so werden die Umrisse der vom Laserscanner erkannten halbstatistischen Objekte fälschlicherweise mit der dahinter befindlichen Kulissenwand in Überdeckung gebracht, sodass sich ein deutlicher Lokalisierungsfehler einstellt. Zwar wurde hierdurch in keinem Fall eine vollständige Divergenz des Filters beobachtet – außerhalb des kritischen Bereiches wurde stets wieder eine korrekte Überdeckung von LiDAR und digitaler Karte erzielt. Trotzdem kann abgeleitet werden, dass die Kartierung halbstatistischer Objekte, welche ein bekanntes statisches Objekt verdecken, für das genutzte Lokalisierungsverfahren von großer Wichtigkeit ist.

In Szenario (c) erfolgt die Lokalisierung mit einer insgesamt deutlich ausgedünnten digitalen Karte. Die Genauigkeit der Lokalisierung verschlechtert sich hierdurch zwar signifikant, im Vergleich zu Szenario (b) ist der Effekt allerdings deutlich kleiner, obwohl ein wesentlich größerer Teil der in der Realität vorhandenen Objekte nicht in der Karte verzeichnet ist. Die von der MCL ermittelte Positionsschätzung liegt auf einem im Mittel verschlechterten Niveau, systematische Fehler bei der Zuordnung der LiDAR-Daten konnten jedoch zu keinem Zeitpunkt beobachtet werden.

Die Ergebnisse zeigen, dass die Bereitstellung kontinuierlich aktualisierter Kartendaten inklusive halbstatistischer Elemente die Genauigkeit und Zuverlässigkeit der Lokalisierung wesentlich verbessern kann. Insbesondere in Fällen, bei denen viele sich gegenseitig

verdeckende Objekte im Sichtbereich der Fahrzeugsensorik liegen, führt eine präzise Kartierung zu deutlichen Vorteilen. Zugleich ist ersichtlich, dass das Lokalisierungsverfahren grundsätzlich robust gegenüber Abweichungen zwischen digitaler Karte und vom Fahrzeug erfasster Realität ist. Auch mit nur sehr unvollständig kartierter Umgebung ist eine Lokalisierung mit hoher Genauigkeit noch möglich.

5.4.2.5 Sonderfälle

Bei der automatischen Berechnung der Positions- und Orientierungsgenauigkeiten sowie der Konvergenzgeschwindigkeiten wurde für vier von 85 Datensätzen das Konvergenzkriterium nicht erfüllt. Die entsprechenden Datensätze können zwei verschiedenen Testszenarien zugeordnet werden. Für beide Testszenarien gilt, dass zwei von fünf Simulationsdurchläufen mit identischen Parametern nicht zu einer ausreichend genauen Konvergenz geführt haben. Das zugrunde liegende Problem tritt also je nach gezogenen Zufallszahlen innerhalb des Partikelfilters nur in einem Teil der Fälle auf.

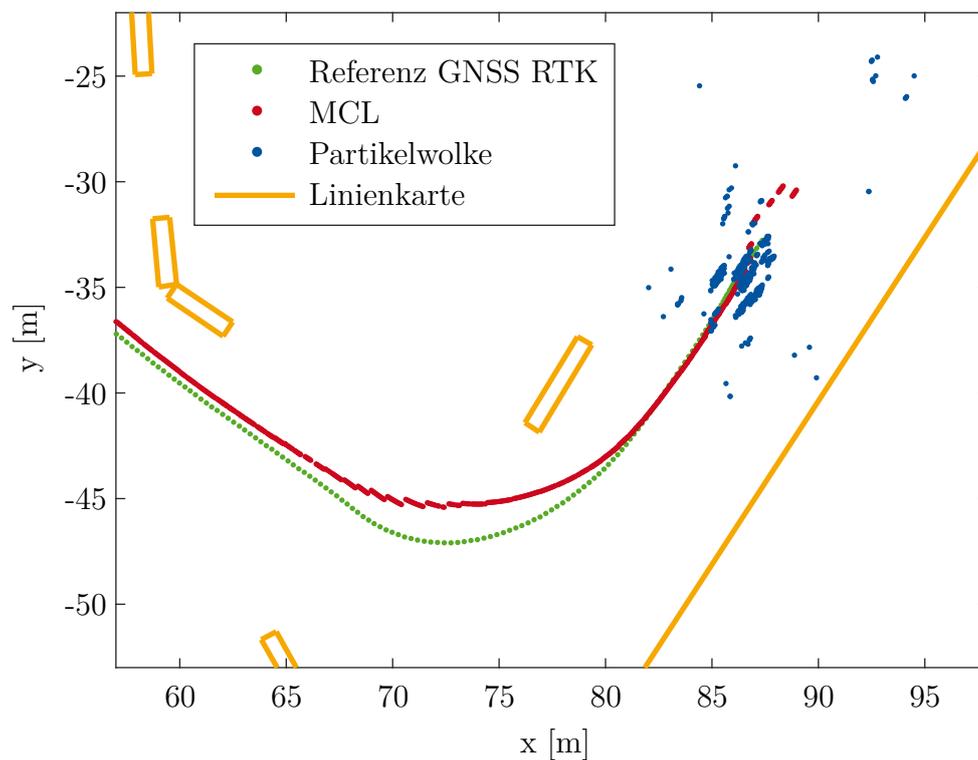


Abb. 5.25: Verteilung der Partikelwolke in Testfall mit schlechtem Konvergenzverhalten

Eine genauere Analyse der beiden Testszenarien ergibt, dass in beiden Fällen die Überschreitungen der Fehlerschranke dem Bereich der Zufahrt 4 zum kartierten Bereich (siehe Abbildung 5.1) zugeordnet werden können. Abbildung 5.25 visualisiert für eines der problematischen Szenarien den Zustand zu einem Zeitpunkt innerhalb der Konvergenzphase. Dargestellt ist neben der von der MCL geschätzten Trajektorie, der GNSS-basierten Referenz und der Linienkarte auch die Partikelwolke zum ausgewählten Zeitpunkt. Es ist erkennbar, dass die Partikelwolke einige geometrische Besonderheiten aufweist. Die Verteilung ist quer zur Bewegungsrichtung des Fahrzeuges durch mehrere, relativ klar abgegrenzte Cluster geprägt. Die Cluster, die als diskrete Positionshypothesen der multimodalen Wahrscheinlichkeitsverteilung interpretiert werden können, sind quer zur Fahrtrichtung relativ schmal, während sie in Bewegungsrichtung eine sehr langgestreckte Form aufweisen. Zudem wird deutlich, dass die Positionsschätzung des Partikelfilters quer zur Fahrtrichtung bereits nach kurzer Zeit nur noch geringe Fehler aufweist. Die wesentlichen Fehler treten in Fahrtrichtung auf und werden insbesondere bei der Einfahrt in die Rechtskurve sichtbar: Bei der geschätzten Trajektorie beginnt die Kurvenfahrt gegenüber der Referenz deutlich zu früh.

Die Besonderheit der Zufahrt 4, in der die Fehler auftreten, liegt in der Symmetrie der hier vorhandenen kartierten Elemente. Das größte vom Fahrzeug erkennbare Objekt ist eine exakt parallel zum geraden Straßenverlauf angebrachte Leitplanke. Auf der gegenüberliegenden Seite der Straße befindet sich eine Kulissenwand, die ebenfalls längs zur Fahrtrichtung ausgerichtet ist. Die quer verlaufende Stirnseite der Kulissenwand kann nur von im Vergleich sehr wenigen Punkten des Laserscanners erfasst werden. Insgesamt enthalten die sichtbaren Geometrien also nur sehr wenige Informationen, anhand derer die Fahrzeugposition längs zum Straßenverlauf abgeleitet werden kann, sodass in dieser Richtung eine größere Unsicherheit des Partikelfilters verbleibt. Erst bei der Einfahrt in die Kurve gelangen weitere charakteristische Objekte ins Sichtfeld des Sensors, sodass eine Korrektur der Position erfolgen kann. Die Korrekturschritte sind in diesem Bereich anhand der charakteristischen Sprünge innerhalb des Positionsverlaufes deutlich erkennbar – der Fehler nimmt ab.

Die Auswertung verdeutlicht eine wesentliche Voraussetzung für die Zuverlässigkeit der kartenbasierten Lokalisierung: Die digitale Karte und das Verkehrsumfeld müssen über geometrische Merkmale verfügen, die eine eindeutige Ausrichtung der Daten des Laserscanners zulassen. Aufgrund der großen Variabilität und dichten Verfügbarkeit von geometrischen Objekten in realen Innenstädten kann davon ausgegangen werden, dass in realen Anwendungen gegenüber dem Testszenario wesentlich günstigere Bedingungen für die kartenbasierte Lokalisierung herrschen. Zudem wird erneut deutlich, dass die Berücksichtigung zusätzlicher halbstatistischer Objekte in der digitalen Karte das Potenzial besitzt, die Robustheit und Genauigkeit der kartenbasierten Lokalisierung deutlich zu verbessern.

5.4.2.6 Fazit Lokalisierung

Insgesamt kann festgestellt werden, dass das Lokalisierungsverfahren den in Kapitel 3.1 aufgestellten Anforderungen entspricht. Der Algorithmus kann auf dem genutzten Prototypensteuergerät sicher in Echtzeit ausgeführt werden und er liefert georeferenzierte Daten zur Fahrzeugposition und -orientierung mit einer Rate von 50 Hz. Das bereitgestellte Kartenformat ermöglicht den Abgleich der Karteninformationen mit den LiDAR-Daten und unterstützt damit die Korrektur der Fahrzeugpose.

Die häufigste Genauigkeit der geschätzten Fahrzeugposition liegt zwischen 8 und 12 cm und damit in einem Bereich, der zur Unterstützung autonomer Fahrfunktionen geeignet ist. Zu berücksichtigen ist aber zugleich, dass der Algorithmus in seiner aktuellen Form keine Genauigkeitsgarantie liefern kann. Hier könnten zukünftig Erweiterungen vorgesehen werden, die etwa auf Basis von Kennwerten zur Übereinstimmung des Laserscans mit der Karte eine zusätzliche Plausibilisierungsstufe realisieren und somit inkonsistente oder unsichere Positionslösungen detektieren. Die Frage nach der Verlässlichkeit der Fahrzeuglokalisierung stellt sich auch in der Konvergenzphase des Algorithmus. Prinzipbedingt benötigt das Filter ausgehend von einer potenziell sehr ungenauen Initialpose zunächst einige Zeitschritte, bis eine Konvergenz zur korrekten Pose erreicht wird. In dieser Zeit treten häufig noch größere Lokalisierungsfehler auf. Einen wichtigen Anhaltspunkt zur Detektion dieser Phase kann die Struktur der Partikelwolke liefern. Während unmittelbar nach der Initialisierung häufig breit verteilte, multimodale Verteilungen zu beobachten sind, ist eine eng begrenzte, kreis- oder ellipsenförmige Partikelwolke ein Merkmal der Tracking-Phase.

Die Fusion von Odometrie- und IMU-Signalen innerhalb des Filters trägt wesentlich dazu bei, dass neben der Position auch die Orientierung des Fahrzeuges mit hoher Genauigkeit geschätzt werden kann. Durch die Parametrierung der zugrunde gelegten Unsicherheiten der Sensorsysteme ist zudem eine Abwägung der verschiedenen Anforderungen an das Lokalisierungssystem möglich: Werden im Verhältnis zur LiDAR-basierten Korrektur die Inertialsysteme des Fahrzeuges höher gewichtet, so stellt sich eine stärkere Dämpfung der ausgegebenen Posen ein. Zwar kann hierdurch die Konvergenzzeit verlängert werden, zugleich ergibt sich aber auch eine höhere Robustheit gegenüber Abweichungen zwischen Karte und Fahrzeugumgebung oder Messfehlern des Laserscanners.

Mit der hier vorgenommenen Parametrierung wird der Fokus auf die Robustheit des Algorithmus gelegt. Tatsächlich kann so in den durchgeführten Tests eine sehr zuverlässige Lokalisierung auch bei einer unvollständigen digitalen Karte erreicht werden. Die Vorteile der kontinuierlich adaptierten digitalen Karte manifestieren sich vor allem dann, wenn statische Elemente im realen Verkehrsumfeld durch halbstatische Objekte verdeckt werden oder die geometrische Konstellation der statischen Elemente aufgrund einer hohen Symmetrie keine eindeutige Positionsbestimmung zulässt. Daneben lässt sich auch eine

Tabelle 5.4: Datenmengen der zwischen Fahrzeug und Infrastruktur auszutauschenden Informationen

Datensatz	Größe
Daten Fahrzeugsensorik	359.3 KB
Daten LiDAR-Sensor	7.75 MB
Digitale Linienkarte	11.1 KB

allgemeine Verbesserung der Lokalisierungsgenauigkeit durch eine detailliertere digitale Karte nachweisen.

5.4.3 Gesamtsystem

Neben den Algorithmen zur Kartierung und Lokalisierung umfasst das entwickelte Gesamtsystem auch die fahrzeug- und infrastrukturseitigen Prozesse zur Kommunikation und Ablaufsteuerung, die insbesondere beim Einfahren in den kartierten Bereich sowie beim Verlassen desselben von Bedeutung sind. Zur Validierung dieser Prozesse werden Experimente mit dem vernetzten Gesamtsystem betrachtet, welche die realen Kommunikationsvorgänge zwischen Fahrzeug und Infrastruktur beinhalten.

In Tabelle 5.4 sind die mittleren Datenmengen angegeben, die sich bei den Experimenten im 4942 m² großen Testfeld ergeben. Es wird deutlich, dass die von der Infrastrukturseite bereitgestellte Linienkarte aufgrund der starken Abstraktion der Informationen nur eine vergleichsweise geringe Datenmenge umfasst. Insbesondere die Rohdaten des LiDAR-Sensors führen zu einer um mehrere Größenordnungen höheren Datenmenge, die 95.7% des vom Fahrzeug aufgezeichneten Gesamtdatensatzes ausmacht.

Tabelle 5.5 zeigt die erforderlichen Zeitbudgets für die verschiedenen Verarbeitungsschritte an den Kartengrenzen. Zu berücksichtigen ist, dass die Daten mit dem in Kapitel 5.1 beschriebenen Kommunikationssystem auf Basis von WLAN erhoben wurden – das finale, LTE-basierte Kommunikationssystem bietet das Potenzial deutlich geringerer Latenzen und höherer Übertragungsraten.

Ein erster Zeitverzug entsteht beim Einfahren in den kartierten Bereich durch die implementierte Hysterese an der Kartengrenze. Diese verhindert die fälschliche Erkennung von mehrfachen Ein- und Ausfahrten durch verrauschte Positionsdaten. Ab dem Zeitpunkt der sicheren Feststellung einer Fahrzeugposition innerhalb der Kartengrenzen erfolgt die Anmeldung am infrastrukturbasierten Kartenserver und die Anforderung sowie der Download der Karte. Dieser Vorgang wird im prototypischen Testsystem über das verschlüsselte Secure Copy (SCP)-Protokoll umgesetzt. Die Karte steht mit dieser Methode nach 0.6725 s zur Verfügung. Beim Verlassen des kartierten Bereiches entsteht

Tabelle 5.5: Zeitbudgets für Kommunikation und Datenverarbeitung an den Kartengrenzen

Verarbeitungsschritt	Zeitbudget
Hysterese Einfahrt	0.604 s
Kartendownload vom Server	0.6725 s
Hysterese Ausfahrt	0.6161 s
Serialisierung Sensordaten	0.1366 s
Upload LiDAR-Daten	2.796 s
Upload Daten Fahrzeugsensorik	0.6217 s

zunächst erneut eine Verzögerung durch die Hysterese, anschließend müssen die vom Fahrzeug aufgezeichneten Rohdaten auf dem Prototypencomputer in binäre protobuf-Dateien serialisiert werden. Der im Vergleich größte Zeitaufwand entsteht für den Upload der Daten zum Kartenserver. Die reine Datenmenge ist für diesen Zeitaufwand allerdings nicht alleine ausschlaggebend: Obwohl der Datensatz der LiDAR-Daten etwa die 22-fache Größe der restlichen Fahrzeugdaten aufweist, fällt für dessen Upload lediglich der 4,5-fache Zeitaufwand an.

Die Auswertung zeigt, dass im prototypischen Testsystem Latenzen bei der Kommunikation eine wichtige Rolle für das gesamte Zeitbudget spielen. Um den Zeitverzug bis zur Verfügbarkeit einer genauen Lokalisierung möglichst gering zu halten, ist somit der Einsatz latenzarmer Kommunikationstechnologien wie 5G in Kombination mit MEC sinnvoll und vielversprechend. Entsprechende Voruntersuchungen wurden in [1] bereits durchgeführt und veröffentlicht. Zugleich ist speziell für den Datenaustausch in Szenarien mit vielen gleichzeitig operierenden Fahrzeugen der Einsatz spezialisierter Kommunikationsprotokolle als Alternative zu SCP erforderlich.

Die zu übertragenden Datenmengen können vor dem Hintergrund der in modernen Mobilfunknetzen verfügbaren Datenraten als für den praktischen Einsatz realistisch eingestuft werden. Günstig ist, dass die Bereitstellung der digitalen Karte für die Fahrzeuglokalisierung mit im Vergleich sehr geringerem Kommunikationsaufwand möglich ist. Fahrzeuge können also leicht Nutznießer der digitalen Karte werden. Die aufwendigere Bereitstellung der Fahrzeugrohdaten könnte im Gegensatz dazu bedarfsgerecht organisiert werden. Der infrastrukturseitige Kartierungsrechner würde dann situativ entscheiden, ob weitere Fahrzeugdatensätze eine Verbesserung der Karte versprechen oder – etwa aufgrund von dichtem Verkehr – bereits genügend Informationen zur Verfügung stehen.

Zusammenfassend kann abgeleitet werden, dass die implementierten Datenformate sowie die Mechanismen zur Kommunikation die erwünschten Aufgaben im prototypischen Testsystem gut erfüllen können. Im Hinblick auf die Verbesserung der Reaktionszeiten an den Grenzen des kartierten Bereiches besteht zudem signifikantes Potenzial beim Einsatz moderner, mobilfunkbasierter Kommunikationstechnik.

6 Zusammenfassung

Mit der vorliegenden Dissertation wurde ein Beitrag geleistet zu kartengestützten Lokalisierungsverfahren für automatisierte Fahrzeuge. Die Besonderheit des hier vorgestellten Ansatzes besteht darin, dass ein vernetztes, innerstädtisches Szenario betrachtet wurde, in dem Fahrzeuge und Infrastruktur miteinander kommunizieren können. In diesem Umfeld wurden Algorithmen zur infrastrukturbasierten, kontinuierlichen Aktualisierung des für die Lokalisierung genutzten Kartenmaterials realisiert.

In der Arbeit wurden zwei wesentliche inhaltliche Säulen betrachtet: Einerseits war ein Ziel die Erforschung der infrastrukturseitigen Algorithmen, die auf Basis von Rohdaten aus Fahrzeugsensoren Änderungen im Verkehrsumfeld detektieren und in die digitale Karte integrieren. Zum anderen wurde auch der fahrzeugseitige Teil betrachtet. Die hier vorgesehenen Algorithmen haben die Aufgabe, eine Fusion der verschiedenen Fahrzeugsensorsysteme mit den Informationen der digitalen Karte in Echtzeit durchzuführen, um eine präzise und robuste Eigenlokalisierung des Fahrzeuges zu ermöglichen.

Basierend auf verschiedenen Ansätzen zur Sensorfusion und Lokalisierung aus der Literatur wurde die fahrzeugseitige Komponente mit einem Partikelfilter als wesentlichem methodischen Rückgrat umgesetzt. Die Methode erlaubt die Nutzung dichter Karten, sie ist also nicht auf diskrete Landmarken angewiesen. Zudem ist sie in der Lage, komplexe multimodale Wahrscheinlichkeitsverteilungen der geschätzten Zustände 2D-Position und Orientierung abzubilden und so mehrere Positionshypothesen zugleich zu verfolgen. Um einen echtzeitfähigen Betrieb mit hoher Samplerate zu erreichen, wurden parallelisierte Prozesse entworfen, die eine unabhängige Ausführung von Prädiktions- und Korrekturschritt auf mehreren Rechenkernen mit anschließender Synchronisation der Ergebnisse erlauben. Als wesentlicher Umgebungssensor wurde ein 360°-Laserscanner eingesetzt.

Der infrastrukturbasierte Kartieralgorithmus wurde so konzipiert, dass er auf Basis aller zur Verfügung stehenden Sensorinformationen eines Fahrzeuges die Fahrzeugtrajektorie durch die Lösung eines GraphSLAM-Optimierungsproblemekonstruktion rekonstruiert. Das mehrschrittige Verfahren wird unter Verwendung der groben Positionsschätzung eines konventionellen GNSS-Systems initialisiert. Anschließend werden durch den Abgleich von LiDAR-Daten mit bereits kartierten Objekten weitere Randbedingungen generiert, anhand derer die Schätzung wesentlich verbessert werden kann. Mit hoher Frequenz verfügbare Inertialmessdaten des Fahrzeuges stützen das Verfahren zusätzlich. Als zweiten

Schritt beinhaltet das Kartierungsverfahren die iterative Aktualisierung einer probabilistischen, gitterbasierten Belegungskarte, aus der schließlich mittels einer automatischen Liniensuche die Komponenten der abstrahierten Linienkarte abgeleitet werden.

Die beschriebenen Verfahren wurden im Rahmen der Arbeit als prototypisches Testsystem implementiert, um einen Proof of Concept zur Funktion und Wirksamkeit des Ansatzes zu erbringen. Die Einflüsse der verschiedenen Parameter auf das Systemverhalten und die Ergebnisse wurden im Detail diskutiert und es wurde anhand von simulativen Studien eine Basisparametrierung vorgenommen. Für die experimentellen Tests wurde auf ein prototypisches Testfahrzeug und ein vom öffentlichen Verkehr abgetrenntes Automobiltestzentrum zurückgegriffen. Im genutzten Testfeld wurde innerstädtische Bebauung mittels schmaler Kulissenwände simuliert. Parkende PKW und weitere portable Hindernisse dienten als zusätzliche, zu kartierende Objekte. Mittels drahtloser Kommunikation erfolgte die Vernetzung zwischen Fahrzeug und Infrastruktur, sodass anhand verschiedener Testfahrten durch das kartierte Areal die quantitative Auswertung aller Teilschritte des Systems inklusive der Datenübertragung erfolgen konnte.

Anhand der experimentellen Ergebnisse wurde nachgewiesen, dass Änderungen im Verkehrsumfeld im Rahmen der automatischen Aktualisierung in die digitale Karte übernommen werden. Sowohl das Entfernen, als auch das Hinzufügen von Objekten erfolgt innerhalb der durch die Parametrierung vorgegebenen zeitlichen Grenzen und es wird eine Genauigkeit im Zentimeterbereich erzielt. Der Vergleich der beiden Ansätze zur Ausrichtung von Laserscans und digitaler Karte – ICP und Scan-Matching – zeigt, dass mit der Wahl des Verfahrens ein Kompromiss zwischen Genauigkeit und Rechenaufwand bei der Trajektorienrekonstruktion getroffen werden kann. Durch die probabilistische Verarbeitung der Umfeldinformationen in der gitterbasierten Belegungskarte ergibt sich ein robustes Verhalten der Kartierung gegenüber Ungenauigkeiten bei der Trajektorienrekonstruktion. Fehler in diesem Schritt führen nicht zu einer direkten Degradierung der Qualität der digitalen Karte, sondern können durch einen statistischen Mittelungseffekt über mehrere Datensätze ausgeglichen werden.

Im Bezug auf die Lokalisierung zeigen die experimentellen Ergebnisse zunächst die gute Nutzbarkeit der abstrahierten digitalen Karte für die Korrektur der Fahrzeugpose basierend auf LiDAR-Signalen. Das implementierte Partikelfilter erzielt eine signifikante Verbesserung der globalen Positionsgenauigkeit gegenüber dem konventionellen GNSS-Empfänger bei zusätzlicher Verfügbarkeit einer Orientierungsschätzung. Die Ergebnisse stehen dabei mit einer Rate von 50 Hz im Echtzeitbetrieb zur Verfügung. Bei der Auswertung wurde neben der Genauigkeit der Lokalisierung auch der Zeitaufwand bis zur Konvergenz des Verfahrens ausgehend von der mit großer Unsicherheit behafteten GNSS-Position betrachtet. Durch die vorgenommene Parametrierung wurde im auftretenden Zielkonflikt zwischen Robustheit des Verfahrens und Konvergenzgeschwindigkeit der Fokus auf den robusten Betrieb gelegt. Probleme bei der Konvergenz des Verfahrens wurden bei zu starker Symmetrie der im Fahrzeugumfeld befindlichen kartierten Objekte

und bei großflächigen Verdeckungen kartierter Objekte durch unbekannte Hindernisse nachgewiesen. Die kontinuierliche Aktualisierung des Kartenmaterials sowie die Kartierung halbstatistischer Objekte besitzt hier ein großes Potenzial zur Verbesserung der Lokalisierungsgenauigkeit.

Insgesamt zeigen die erzielten Ergebnisse, dass das entwickelte System sehr vielversprechende Eigenschaften für den Einsatz in realen innerstädtischen Szenarien aufweist. Die kartenbasierte Lokalisierung ermöglicht die hochgenaue, globale Lokalisierung auch ohne die durchgehende Verfügbarkeit satellitenbasierter Signale und die kontinuierliche, automatisierte Aktualisierung der digitalen Karte stellt jederzeit eine verlässliche Datenbasis zur Verfügung.

6.1 Ausblick

Im Vergleich zum im Rahmen der Arbeit genutzten Testfeld ist in realen Innenstädten von wesentlich dichteren und komplexeren Konstellationen kartierbarer Geometrien auszugehen, von denen Kartierung und Lokalisierung stark profitieren können. Zugleich kann das vom Fahrzeug wahrnehmbare Verkehrsumfeld in solchen Szenarien zum großen Teil von bewegten Fahrzeugen und anderen dynamischen Objekten geprägt sein, die für die Lokalisierung nicht nutzbar sind. Die Übertragung in den Realverkehr bietet also sowohl deutliche Potenziale für eine Verbesserung der Leistungsfähigkeit des Systems, als auch zusätzliche Herausforderungen, die evaluiert und adressiert werden müssen.

Im Hinblick auf die Kommunikation ist bei einer Umsetzung im größeren Maßstab auf Basis von Mobilfunktechnologien die Bandbreite von Bedeutung, die für die sinnvolle Nutzung des Kartendienstes mit zahlreichen verbundenen Fahrzeugen erforderlich ist. Aus Kostengründen kann hier eine weitere Reduktion der benötigten Datenmengen erforderlich sein, wobei insbesondere die Rohdaten der Fahrzeugsensoren in Betracht kommen. Denkbar sind alternative Schemata zur Aufteilung des Rechenaufwandes zwischen PKW und Infrastruktur. Während die in dieser Arbeit entwickelte Lösung eine vollständige Auslagerung der Kartierungsaufgaben auf die Infrastrukturseite vorsieht, wäre grundsätzlich auch eine Vorverarbeitung auf dem Fahrzeugrechner möglich, sodass der Infrastruktur bereits abstrahierte und damit potenziell komprimierte Daten zugesendet werden könnten.

In der vorliegenden Arbeit wird ein stark abstrahiertes, geometrisches Layer einer digitalen Karte mit dem Ziel der Fahrzeuglokalisierung untersucht. Teile der entwickelten Methoden können allerdings auch für die kontinuierliche Aktualisierung zusätzlicher semantischer Kartenebenen genutzt werden. Anhand der mit hoher Genauigkeit rekonstruierten Fahrzeugtrajektorie können etwa erkannte Verkehrsschilder, Fahrbahnmarkierungen oder

Lichtzeichenanlagen präzise verortet werden und mögliche Änderungen der Verkehrsführung innerhalb eines entsprechenden semantischen Kartenlayers übernommen werden.

Erweiterungsmöglichkeiten bestehen auch hinsichtlich der Nutzung alternativer Sensorsysteme für die Umfelderkennung. Neben LiDAR kommen auch RADAR- oder Kamerasysteme in aktuellen Fahrzeugen zum Einsatz und liefern geometrische Informationen zum Fahrzeugumfeld. Die im Rahmen dieser Arbeit entwickelte abstrahierte Kartendarstellung ist grundsätzlich nicht sensorspezifisch. Mittels adaptierter Vorverarbeitungsstufen könnte die gezeigte Funktionalität also auch auf Basis von kostengünstigeren Sensoren umgesetzt werden. Im Allgemeinen müsste – insbesondere bei Kameradaten – mit im Vergleich zur LiDAR-Technik deutlich größeren Unsicherheiten gerechnet werden. Zugleich könnte durch die Kombination der Vorteile verschiedener Messprinzipien aber auch die Robustheit, Genauigkeit und der Funktionsumfang des Systems deutlich erweitert werden.

A Anhang

A.1 Spezifikation der Sensorsysteme

Tabelle A.1: Spezifikation der Fahrzeugsensorik [65]

Radodometer [8]	
Hersteller und Bezeichnung	ams AS5040
Typ	Hall-Effekt-Sensor
Auflösung	1024 Ticks/Umdrehung
Interface (eingebetteter Rechner)	CAN
Inertiale Messeinheit [9]	
Hersteller und Bezeichnung	Analog Devices ADIS16405BMLZ
Typ	MEMS-Sensor
Messgrößen	3x Drehrate, 3x Beschleunigung, 3x Magnetische Feldstärke, 1x Temperatur
Interface (eingebetteter Rechner)	CAN
Messbereich (Drehrate)	$\pm 300^\circ \text{s}^{-1}$
Auflösung (Drehrate)	$0.05^\circ/\text{s}/\text{LSB}$
GNSS [82]	
Hersteller und Bezeichnung	uBlox EVK-M8T
Typ	Einfrequenz-GNSS-Empfänger
Genauigkeit (typ.)	2 - 3 m
Sample Rate (typ.)	5 Hz
Interface (eingebetteter Rechner)	USB
GNSS (Referenz) [75]	
Hersteller und Bezeichnung	Septentrio AsteRx3
Typ	Mehrfrequenz-GNSS-Empfänger mit RTK-Korrektur
Korrekturdaten	SAPOS Nordrhein-Westfalen
Genauigkeit horizontal	6 mm + 0.5 ppm

Genauigkeit vertikal	1 cm + 1 ppm
Sample Rate (typ.)	10 Hz
Interface (eingebetteter Rechner)	RS232
<hr/>	
LiDAR [83]	
<hr/>	
Hersteller und Bezeichnung	Velodyne VLP-16
Typ	3D LiDAR-Sensor
Reichweite (typ.)	100 m
Messgenauigkeit (typ.)	bis zu ± 3 cm
Sichtfeld (horiz.)	360°
Sichtfeld (vert.)	$\pm 15^\circ$
Auflösung (horiz., typ.)	0.4°
Auflösung (vert., typ.)	2°
Sample Rate (typ.)	20 Hz
Interface (eingebetteter Rechner)	Ethernet (100 Mbit/s)
<hr/>	

A.2 Spezifikation des Echtzeitrechners

Tabelle A.2: Spezifikation des Fahrzeugrechners [58]

Hersteller und Bezeichnung	Neosys Nuvo 5108 VTC
Prozessor	Intel i7-6700TE (8M Cache, 2.4/3.4 GHz)
Arbeitsspeicher	16 GB DDR4-2133 SDRAM
GPU	Onboard Intel HD graphics 530
USB	4x USB 3.0, 4x USB 2.0
Serieller Anschluss	3x RS-232
CAN	1x CAN 2.0
Ethernet	10x Gigabit Ethernet (RJ45)
WiFi	1x Dualband-Adapter Edimax EW-7811UAC, 802.11a/b/g/n/ac
<hr/>	

Eigene Veröffentlichungen

- [1] QUACK, T. ; BÖSINGER, M. ; HESSELER, F.-J. ; ABEL, D. : Infrastructure-based digital maps for connected autonomous vehicles. In: *at - Automatisierungstechnik* 66 (2018), Nr. 2, S. 183–191
- [2] QUACK, T. ; HESSELER, F.-J. ; ABEL, D. : Digitale Karten für die hochgenaue Lokalisierung von vernetzten Fahrzeugen im städtischen Umfeld. In: *Positionierung und Navigation für intelligente Transportsysteme (DGON POSNAV)*. Berlin, Deutschland, 2018
- [3] QUACK, T. ; HESSELER, F.-J. ; ABEL, D. : Fast real-time localization with sparse digital maps for connected automated vehicles in urban areas. In: *IFAC-PapersOnLine* 52 (2019), Nr. 5, S. 366–371
- [4] QUACK, T. M. ; REITER, M. ; ABEL, D. : Digital Map Generation and Localization for Vehicles in Urban Intersections using LiDAR and GNSS Data. In: *IFAC-PapersOnLine* 50 (2017), Nr. 1, S. 251–257

Betreute studentische Arbeiten

- [5] BAUER, D. ; BETREUER: QUACK, T. : *Automatic Creation of Digital, Three Dimensional Maps for Localization of Autonomous Vehicles*. Institut für Regelungstechnik, RWTH Aachen University, Masterarbeit (unveröffentlicht), 2017
- [6] SCHEFFE, P. ; BETREUER: QUACK, T. : *Kontinuierlich lernende digitale Karten für die Lokalisierung vernetzter automatisierter Fahrzeuge in urbanen Verkehrsräumen*. Institut für Regelungstechnik, RWTH Aachen University, Masterarbeit (unveröffentlicht), 2018

Literaturverzeichnis

- [7] ALSAYED, Z. ; BRESSON, G. ; NASHASHIBI, F. ; VERROUST-BLONDET, A. : PML-SLAM: A solution for localization in large-scale urban environments. In: *7th Workshop on Planning, Perception and Navigation for Intelligent Vehicles – 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Hamburg, Germany, 2015
- [8] AMS AG: *AS5040 10-Bit 360° Programmable Magnetic Rotary Encoder Data Sheet*. https://ams.com/documents/20143/36005/AS5040_DS000374_3-00.pdf. Version: 2019, Abruf: 01.10.2019
- [9] ANALOG DEVICES INC.: *Triaxial Inertial Sensor with Magnetometer ADIS16400/ADIS16405 Data Sheet*. https://www.analog.com/media/en/technical-documentation/data-sheets/ADIS16400_16405.pdf. Version: 2019, Abruf: 01.10.2019
- [10] ATC GMBH: *Aldenhoven Testing Center: Tracks*. <https://www.atc-aldenhoven.de/de/strecken.html>. Version: 2019, Abruf: 11.02.2019
- [11] BELLEKENS, B. ; SPRUYT, V. ; MAARTEN WEYN, R. B.: A survey of rigid 3D pointcloud registration algorithms. In: *Fourth International Conference on Ambient Computing, Applications, Services and Technologies*. Rome, Italy, 2014, S. 8–13
- [12] BERNS, K. ; PUTTKAMER, E. v.: *Autonomous Land Vehicles: Steps towards Service Robots*. Wiesbaden : Vieweg + Teubner, 2009
- [13] BESL, P. J. ; MCKAY, N. D. ; BESL, P. J.: A method for registration of 3-D shapes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992), Nr. 2, S. 239–256
- [14] BONETTI, P. : *HERE introduces HD Live Map to show the path to highly automated driving*. <http://360.here.com/2016/01/05/here-introduces-hd-live-map-to-show-the-path-to-highly-automated-driving/>. Version: 2016, Abruf: 15.03.2019
- [15] BUNDESREGIERUNG DER BUNDESREPUBLIK DEUTSCHLAND: *5G-Strategie für Deutschland: Eine Offensive für die Entwicklung Deutschlands zum Leitmarkt für 5G-Netze und -Anwendungen*. <https://www.bmvi.de/>

- SharedDocs/DE/Publikationen/DG/098-dobrindt-5g-strategie.pdf?__blob=publicationFile. Version:2017, Abruf: 30.09.2019
- [16] BUNDESREGIERUNG DER BUNDESREPUBLIK DEUTSCHLAND: *Aktionsplan Forschung für autonomes Fahren: Ein übergreifender Forschungsrahmen von BMBF, BMWi und BMVI*. https://www.bmbf.de/upload_filestore/pub/Aktionsplan_Forschung_fuer_autonomes_Fahren.pdf. Version:2019, Abruf: 29.09.2019
- [17] BURKI, M. ; DYMCZYK, M. ; GILITSCHENSKI, I. ; CADENA, C. ; SIEGWART, R. ; NIETO, J. : Map Management for Efficient Long-Term Visual Localization in Outdoor Environments. In: *2018 IEEE Intelligent Vehicles Symposium*. Changshu, China, 2018
- [18] CACILO, A. ; SCHMIDT, S. ; WITTLINGER, P. ; HERRMANN, F. ; BAUER, W. ; SAWADE, O. ; DODERER, H. ; HARTWIG, M. ; SCHOLZ, V. : *Hochautomatisiertes Fahren auf Autobahnen – Industriepolitische Schlussfolgerungen: Studie im Auftrag des Bundesministeriums für Wirtschaft und Energie (BMWi)*. https://www.bmw.de/Redaktion/DE/Downloads/H/hochautomatisiertes-fahren-auf-autobahnen.pdf%3F__blob%3DpublicationFile%26v%3D1. Version:2015, Abruf: 30.09.2019
- [19] CADENA, C. ; CARLONE, L. ; CARRILLO, H. ; LATIF, Y. ; SCARAMUZZA, D. ; NEIRA, J. ; REID, I. ; LEONARD, J. J.: Past, Present, and Future of Simultaneous Localization and Mapping: Towards the Robust-Perception Age. In: *IEEE Transactions on Robotics* 32 (2016), Nr. 6, S. 1309–1332
- [20] CASTELLANOS, J. A. ; MONTIEL, J. ; NEIRA, J. ; TARDOS, J. D.: The SPmap: A probabilistic framework for simultaneous localization and map building. In: *IEEE Transactions on Robotics and Automation* 15 (1999), Nr. 5, S. 948–952
- [21] CHEN, Y. ; DAVIS, T. A. ; HAGER, W. W.: Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. In: *ACM Transactions on Mathematical Software* (2008), S. 1–14
- [22] DAYOUB, F. ; CIELNIAK, G. ; DUCKETT, T. : Long-term experiments with an adaptive spherical view representation for navigation in changing environments. In: *Robotics and Autonomous Systems* 59 (2011), Nr. 5, S. 285–295
- [23] DMA: Department of Defense World Geodetic System 1984. In: *DMA Technical Report TR 8350.2 and Supplement TR 830.2-A,B* (1987)
- [24] DOUCET, A. ; JOHANSEN, A. M.: A tutorial on particle filtering and smoothing: Fifteen years later. In: CRISAN, D. (Hrsg.) ; ROZOVSKY, B. (Hrsg.): *Nonlinear Filtering Handbook*. Oxford : Oxford University Press, 2011, S. 656–704

- [25] DOUCET, A. ; FREITAS, N. de ; GORDON, N. : An Introduction to Sequential Monte Carlo Methods. In: DOUCET, A. (Hrsg.) ; FREITAS, N. de (Hrsg.) ; GORDON, N. (Hrsg.): *Sequential Monte Carlo Methods in Practice*. New York : Springer, 2001, S. 3–14
- [26] EUROPÄISCHES PARLAMENT: *Verordnung (EG) Nr. 661/2009 vom 13. Juli 2009 über die Typp Genehmigung von Kraftfahrzeugen, Kraftfahrzeuganhängern und von Systemen, Bauteilen und selbstständigen technischen Einheiten für diese Fahrzeuge hinsichtlich ihrer allgemeinen Sicherheit*. https://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=uriserv:OJ.L_.2009.200.01.0001.01.DEU
- [27] EUROPÄISCHES PARLAMENT: *Verordnung (EU) 215/758 vom 29. April 2015 über Anforderungen für die Typp Genehmigung zur Einführung des auf dem 112-Notruf basierenden bordeigenen eCall-Systems in Fahrzeugen und zur Änderung der Richtlinie 2007/46/EG*. <https://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32015R0758>
- [28] FISCHLER, M. A. ; BOLLES, R. C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In: *Communications of the ACM* 24 (1981), Nr. 6, S. 381–395
- [29] FOX, D. ; BURGARD, W. ; DELLAERT, F. ; THRUN, S. : Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In: *National Conference on artificial intelligence (AAAI)*, 1999, S. 343–349
- [30] GAI, P. ; VIOLANTE, M. : Automotive embedded software architecture in the multi-core age. In: *2016 21th IEEE European Test Symposium (ETS)*, 2016, S. 1–8
- [31] GASSER, T. ; ARZT, C. ; AYOUBI, M. ; BARTELS, A. ; BÜRKLE, L. ; EIER, J. ; FLEMISCH, F. ; HÄCKER, D. ; HESSE, T. ; HUBER, W. ; LOTZ, C. ; MAURER, M. ; RUTH-SCHUMACHER, S. ; SCHWARZ, J. ; VOGT, W. : *Rechtsfolgen zunehmender Fahrzeugautomatisierung*. <https://bast.opus.hbz-nrw.de/opus45-bast/frontdoor/deliver/index/docId/541/file/F83.pdf>. Version: 2012, Abruf: 30.09.2019
- [32] GOOGLE DEVELOPERS: *Procol Buffers – Developer Guide*. <https://developers.google.com/protocol-buffers/docs/overview>. Version: 2019, Abruf: 02.10.2019
- [33] GREWAL, M. S. ; WEILL, L. R. ; ANDREWS, A. P.: *Global positioning systems, inertial navigation and integration*. 2. Auflage. Hoboken, USA : Wiley, 2007
- [34] GRISSETTI, G. ; STACHNISS, C. ; BURGARD, W. : Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, S. 2432–2437

-
- [35] GRISETTI, G. ; KÜMMERLE, R. ; STACHNISS, C. ; BURGARD, W. : A Tutorial on Graph-Based SLAM. In: *IEEE Intelligent Transportation Systems Magazine* 2 (2010), Nr. 4, S. 31–43
- [36] GUSTAFSSON, F. ; GUNNARSSON, F. ; BERGMAN, N. ; FORSSELL, U. ; JANSSON, J. ; KARLSSON, R. ; NORDLUND, P. J.: Particle filters for positioning, navigation, and tracking. In: *IEEE Transactions on Signal Processing* 50 (2002), Nr. 2, S. 425–437
- [37] HATA, A. Y. ; RAMOS, F. T. ; WOLF, D. F.: Monte Carlo Localization on Gaussian Process Occupancy Maps for Urban Environments. In: *IEEE Transactions on Intelligent Transportation Systems* (2017), S. 1–10
- [38] HERE: *HERE HD Live Map: The most intelligent sensor for autonomous driving*. https://www.here.com/sites/g/files/odxslz166/files/2018-11/HERE_HD_Live_Map_one_pager.pdf. Version: 2019, Abruf: 15.03.2019
- [39] HERTENSTEIN, P. ; GRÜNE, C. : *Technology driven M&A in the automotive industry: From automobile to autonomous*. [https://www.ey.com/Publication/vwLUAssets/ey-autobauer-investieren-verstaerkt-in-neue-technologien-und-geschaeftsmodelle/\\$File/ey-autobauer-investieren-verstaerkt-in-neue-technologien-und-geschaeftsmodelle.pdf](https://www.ey.com/Publication/vwLUAssets/ey-autobauer-investieren-verstaerkt-in-neue-technologien-und-geschaeftsmodelle/$File/ey-autobauer-investieren-verstaerkt-in-neue-technologien-und-geschaeftsmodelle.pdf). Version: 2019, Abruf: 29.09.2019
- [40] HOLZ, D. ; ICHIM, A. E. ; TOMBARI, F. ; RUSU, R. B. ; BEHNKE, S. : Registration with the Point Cloud Library: A Modular Framework for Aligning in 3-D. In: *IEEE Robotics Automation Magazine* 22 (2015), Nr. 4, S. 110–124
- [41] IBISCH, A. ; STUMPER, S. ; ALTINGER, H. ; NEUHAUSEN, M. ; TSCHENTSCHER, M. ; SCHLIPSING, M. ; SALINEN, J. ; KNOLL, A. : Towards autonomous driving in a parking garage: Vehicle localization and tracking using environment-embedded LIDAR sensors. In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. Gold Coast City, Australia, 2013, S. 829–834
- [42] KEHR, S. ; QUIÑONES, E. ; BÖDDEKER, B. ; SCHÄFER, G. : Parallel execution of AUTOSAR legacy applications on multicore ECUs with timed implicit communication. In: *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. San Francisco, USA, 2015, S. 1–6
- [43] KHALEGHI, B. ; KHAMIS, A. ; KARRAY, F. O. ; RAZAVI, S. N.: Multisensor data fusion: A review of the state-of-the-art. In: *Information Fusion* 14 (2013), Nr. 1, S. 28–44
- [44] KRAIL, M. : *Energie- und Treibhausgaswirkungen des automatisierten und vernetzten Fahrens im Straßenverkehr*. <https://www.isi.fraunhofer.de/content/dam/isi/dokumente/ccn/2019/energie-treibhausgaswirkungen-vernetztes-fahren.pdf>. Version: 2019, Abruf: 29.09.2019

- [45] KRZOSSA, T. : *5G Edge Computing: Die kleinsten Rechenzentren im Netz*. <https://www.vodafone.de/newsroom/netz/5g-mobile-edge-computing-so-funktiniert-mec/>. Version: 2019, Abruf: 30.09.2019
- [46] KÜMMERLE, J. ; SONS, M. ; POGGENHANS, F. ; KÜHNER, T. ; LAUER, M. ; STILLER, C. : Accurate and Efficient Self-Localization on Roads using Basic Geometric Primitives. In: *2019 International Conference on Robotics and Automation (ICRA)*. Montreal, Canada, 2019, S. 5965–5971
- [47] KÜMMERLE, R. ; GRISETTI, G. ; STRASDAT, H. ; KONOLIGE, K. ; BURGARD, W. : g2o: A General Framework for Graph Optimization. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China, 2011, S. 3607–3613
- [48] LENZ, B. ; WINNER, H. ; GERDES, J. C. ; MAURER, M. : *Autonomes Fahren: Technische, rechtliche und gesellschaftliche Aspekte*. Berlin, Heidelberg : Springer Vieweg, 2015
- [49] LEVINSON, J. ; MONTEMERLO, M. ; THRUN, S. : Map-Based Precision Vehicle Localization in Urban Environments. In: *Robotics: Science and Systems*. Atlanta, USA, 2007
- [50] LEVINSON, J. ; THRUN, S. : Robust vehicle localization in urban environments using probabilistic maps. In: *2010 IEEE International Conference on Robotics and Automation (ICRA)*, 2010, S. 4372–4378
- [51] LUNDGREN, M. ; STENBORG, E. ; SVENSSON, L. ; HAMMARSTRAND, L. : Vehicle self-localization using off-the-shelf sensors and a detailed map. In: *2014 IEEE Intelligent Vehicles Symposium (IV)*. Ypsilanti, USA, 2014, S. 522–528
- [52] MAO, Y. ; YOU, C. ; ZHANG, J. ; HUANG, K. ; LETAIEF, K. B.: A Survey on Mobile Edge Computing: The Communication Perspective. In: *IEEE Communications Surveys and Tutorials* (2017), Nr. 99, S. 1
- [53] MASSOW, K. ; KWELLA, B. ; PFEIFER, N. ; HÄUSLER, F. ; PONTOW, J. ; RADUSCH, I. ; HIPPEL, J. ; DÖLITZSCHER, F. ; HAUEIS, M. : Deriving HD maps for highly automated driving from vehicular probe data. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, S. 1745–1752
- [54] MATTERN, N. ; WANIELIK, G. : Camera-based vehicle localization at intersections using detailed digital maps. In: *2010 IEEE/ION Position Location and Navigation Symposium (PLANS)*. Indian Wells, USA, 2010, S. 1100–1107
- [55] MECSSWARE GMBH: *Mobile Edge Cloud Server*. <https://www.mecssware.com/products.html>. Version: 2019, Abruf: 01.10.2019

-
- [56] MEYER-DELIUS, D. : *Probabilistic Modeling of Dynamic Environments for Mobile Robots: Dissertation*. Freiburg im Breisgau : Albert-Ludwigs-Universität, 2011
- [57] MÜHLFELLNER, P. ; BÜRKI, M. ; BOSSE, M. ; DERENDARZ, W. ; PHILIPPSEN, R. ; FURGALE, P. : Summary Maps for Lifelong Visual Localization. In: *Journal of Field Robotics* 33 (2016), Nr. 5, S. 561–590
- [58] NEOSYS TECHNOLOGY INC.: *Nuvo-5100VTC Series Data Sheet*. https://www.neosys-tech.com/Resource/Product_Document/Nuvo-5100VTC-series/Nuvo-5100VTC_Datasheet.pdf. Version: 2019, Abruf: 01.10.2019
- [59] O’CALLAGHAN, S. T. ; RAMOS, F. T.: Gaussian process occupancy maps. In: *The International Journal of Robotics Research* 31 (2011), Nr. 1, S. 42–62
- [60] OLSON, E. B.: Real-time correlative scan matching. In: *2009 IEEE International Conference on Robotics and Automation (ICRA)*, 2009, S. 4387–4393
- [61] PINK, O. ; STILLER, C. : Automated map generation from aerial images for precise vehicle localization. In: *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2010, S. 1517–1522
- [62] POGGENHANS, F. ; SALSCHIEDER, N. O. ; STILLER, C. : Precise Localization in High-Definition Road Maps for Urban Regions. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, S. 2167–2174
- [63] PRESSE- UND INFORMATIONSAMT DER BUNDESREGIERUNG: *Straßenverkehrsgesetz – Automatisiertes Fahren auf dem Weg*. <https://www.bundesregierung.de/breg-de/aktuelles/automatisiertes-fahren-auf-dem-weg-326108>. Version: 2019, Abruf: 29.09.2019
- [64] QIN, B. ; CHONG, Z. J. ; BANDYOPADHYAY, T. ; ANG, M. H. ; FRAZZOLI, E. ; RUS, D. : Curb-intersection feature based Monte Carlo Localization on urban roads. In: *2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012, S. 2640–2646
- [65] REITER, M. ; ALRIFAEI, B. ; ABEL, D. : Model Scale Experimental Vehicle as Test Platform For Autonomous Driving Applications. In: *FISITA 2014 World Automotive Congress*. Maastricht, Netherlands, 2014
- [66] ROHDE, J. ; JATZKOWSKI, I. ; MIELENZ, H. ; ZÖLLNER, J. M.: Vehicle pose estimation in cluttered urban environments using multilayer adaptive Monte Carlo localization. In: *2016 19th International Conference on Information Fusion (FUSION)*, 2016, S. 1774–1779
- [67] ROSEN, D. M. ; MASON, J. ; LEONARD, J. J.: Towards lifelong feature-based mapping in semi-static environments. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, S. 1063–1070

- [68] RUSINKIEWICZ, S. ; LEVOY, M. : Efficient Variants of the ICP Algorithm. In: *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, 2001
- [69] RUSU, R. B. ; COUSINS, S. : 3D is here: Point Cloud Library (PCL). In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2011, S. 1–4
- [70] SAE INTERNATIONAL: *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. <https://www.beuth.de/de/norm/sae-j-3016/292598163>. Version: 2018, Abruf: 30.09.2019
- [71] SCHINDLER, A. : *Vehicle self-localization using high-precision digital maps: Dissertation*. Passau : Universität Passau, 2013
- [72] SCHIOTKA, A. ; SUGER, B. ; BURGARD, W. : Robot localization with sparse scan-based maps. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, S. 642–647
- [73] SEFATI, M. ; DAUM, M. ; SONDERMANN, B. ; KREISKOTHER, K. D. ; KAMPKER, A. : Improving vehicle localization using semantic and pole-like landmarks. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, S. 13–19
- [74] SEIF, H. G. ; HU, X. : Autonomous Driving in the iCity – HD Maps as a Key Challenge of the Automotive Industry. In: *Engineering 2* (2016), Nr. 2, S. 159–162
- [75] SEPTENTRIO SATELLITE NAVIGATION N.V.: *AsteRx3: GNSS Multi-frequency Receiver Data Sheet*. https://www.navtechgps.com/assets/1/7/AsteRx3_DS_V260110_19_DS1.PDF. Version: 2019, Abruf: 01.10.2019
- [76] SIMON, D. : *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Hoboken, USA : Wiley-Interscience, 2006
- [77] SONS, M. ; STILLER, C. : Efficient Multi-Drive Map Optimization towards Life-long Localization using Surround View. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, S. 2671–2677
- [78] SPANGENBERG, R. ; GOEHRING, D. ; ROJAS, R. : Pole-based localization for autonomous vehicles in urban scenarios. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, S. 2161–2166
- [79] STATISTISCHES BUNDESAMT DEUTSCHLAND: *Verkehrsunfälle – Unfallursachen bei Unfällen mit Personenschaden*. https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/_inhalt.html. Version: 2019, Abruf: 29.09.2019
- [80] THRUN, S. ; BURGARD, W. ; FOX, D. : *Probabilistic robotics*. Cambridge, USA : MIT Press, 2006

-
- [81] TOMTOM: *HD map with ROADDNA: High definition map with sensor-agnostic localization*. http://download.tomtom.com/open/banners/HD_Map_with_RoadDNA_Product_Info_Sheet.pdf. Version: 2019, Abruf: 15.03.2019
- [82] U-BLOX AG: *NEO-M8 Data Sheet*. https://www.u-blox.com/sites/default/files/NEO-M8-FW3_DataSheet_%28UBX-15031086%29.pdf. Version: 2019, Abruf: 01.10.2019
- [83] VELODYNE LIDAR: *Puck Real-Time 3D LiDAR Sensor Data Sheet*. https://www.mapix.com/wp-content/uploads/2018/07/63-9229_Rev-H_Puck-Datasheet_Web-1.pdf. Version: 2019, Abruf: 01.10.2019
- [84] VERBAND DER AUTOMOBILINDUSTRIE: *Automatisierung – Von Fahrerassistenzsystemen zum automatisierten Fahren*. <https://www.vda.de/dam/vda/publications/2015/automatisierung.pdf>. Version: 2015, Abruf: 30.09.2019
- [85] VODAFONE GMBH: *5G Mobility Lab - die Zukunft des Fahrens*. <https://www.vodafone.de/innovationpark/5g-mobility-lab.html>. Version: 2019, Abruf: 11.02.2019
- [86] WALCOTT-BRYANT, A. ; KAESS, M. ; JOHANNSSON, H. ; LEONARD, J. J.: Dynamic pose graph SLAM: Long-term mapping in low dynamic environments. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, S. 1871–1878
- [87] WENDEL, J. : *Integrierte Navigationssysteme: Sensordatenfusion, GPS und inertielle Navigation*. 2. Auflage. München : Oldenbourg, 2011
- [88] WINNER, H. (Hrsg.) ; HAKULI, S. (Hrsg.) ; WOLF, G. (Hrsg.): *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*. 1. Auflage. Wiesbaden : Vieweg + Teubner, 2009
- [89] WOLCOTT, R. W. ; EUSTICE, R. M.: Visual localization within LIDAR maps for automated urban driving. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, S. 176–183
- [90] XU, G. ; XU, Y. : *GPS: Theory, algorithms and applications*. 3. Auflage. Berlin and Heidelberg : Springer, 2016