

Universität der Bundeswehr München
Fakultät für Luft- und Raumfahrttechnik
Institut für Technik Autonomer Systeme

Umgebungserfassung für Fahrerassistenzsysteme mit hierarchischen Belegungskarten

Dipl.-Inf. Matthias Roland Schmid

Vollständiger Abdruck der bei der
Fakultät für Luft- und Raumfahrttechnik
der Universität der Bundeswehr München
zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

eingereichten Dissertation.

Vorsitzender: Prof. Dr. Berthold Färber
1. Berichterstatter: Prof. Dr.-Ing. Hans-Joachim Wünsche
2. Berichterstatter: Prof. Dr.-Ing. Klaus C. J. Dietmayer

Diese Dissertation wurde am 11. Juni 2012 bei der Universität der Bundeswehr München, 85577 Neubiberg eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 20. Juni 2012 angenommen.

Tag der Prüfung:
26. November 2012

Danksagung

Die vorliegende Dissertation entstand im Rahmen meiner dreijährigen Tätigkeit als wissenschaftlicher Mitarbeiter an der Universität der Bundeswehr München.

Herrn Prof. Dr.-Ing. Hans-Joachim Wünsche danke ich für die Betreuung meiner Arbeit, den fachlichen Rat und die stetige Unterstützung, die mir dennoch eine große Freiheit bei der Themenwahl und der Ausarbeitung dieser Dissertation ließ.

Herrn Prof. Dr.-Ing. Klaus C. J. Dietmayer danke ich für die Übernahme des Koreferates und das Interesse an meiner Arbeit.

Den ehemaligen Kollegen des Instituts Technik Autonomer Systeme möchte ich danken für die stets angenehme Arbeitsatmosphäre, die spannenden Diskussionen und die zahlreichen Anregungen, die in diese Arbeit einfließen.

Im Forschungszentrum der Daimler AG in Ulm fand die Umsetzung dieser Arbeit statt. Alle Arbeitsmittel inklusive der Versuchsfahrzeuge stellte die Abteilung Umgebungserfassung zur Verfügung. Für die sehr gute Zusammenarbeit und Unterstützung möchte ich allen ehemaligen Kollegen dort danken. Hervorgehoben sei Dr.-Ing. Jürgen Dickmann, der mir die Mitarbeit in spannenden Projekten, auch standortübergreifend, ermöglichte, sowie Dr.-Ing. Mirko Mählich, der mit seinen Ideen und inspirierenden Gesprächen den Grundstein zu dieser Arbeit legte.

Für die wertvollen studentischen Beiträge zu dieser Arbeit danke ich Christian Baur, Savas Ates, Dmitriy Kuzikov und Yongkie Wiyogo.

Im Besonderen danke ich schließlich meiner Familie, die mich stets unterstützt hat und immer hinter mir stand.

Wolfsburg, den 20. Dezember 2012

Umgebungserfassung für Fahrerassistenzsysteme mit hierarchischen Belegungskarten

Zusammenfassung

Die Zahl der im Straßenverkehr Verunglückten in Deutschland ist rückläufig. Die Fahrzeughersteller tragen ihren Teil zur Sicherheit im Straßenverkehr bei, indem sie Fahrern immer komplexere Fahrerassistenz- und Sicherheitssysteme anbieten, um sicher und entspannt ans Ziel zu kommen.

Für moderne Fahrerassistenzsysteme ist eine zuverlässige und umfassende Umgebungserfassung unerlässlich. Um möglichst viele unterschiedliche Systeme mit einer gemeinsamen Umgebungserfassung bedienen zu können, ist es notwendig, dass diese eine möglichst sensorunabhängige, anwendungsübergreifende, konsistente und umfassende Repräsentation der Umgebung liefert.

Belegungskarten ermöglichen die Analyse von Freiraum, die Fahrbahnverlaufsschätzung, die Detektion bewegter Objekte und die Selbstlokalisierung. In dieser Arbeit wird daher eine dreidimensionale Belegungskarte zur Umgebungsmodellierung für Fahrerassistenzsysteme eingesetzt.

Um den effizienten Einsatz von dreidimensionalen Karten zu ermöglichen wird ein Verfahren vorgestellt, mit dem sich der Detaillierungsgrad der Umgebungsmodellierung dynamisch und anwendungsgesteuert anpassen lässt.

Um die Umgebungserfassung zu vervollständigen wird sowohl die rasterbasierte Selbstlokalisierung in zwei und drei Dimensionen als auch die Detektion und Verfolgung bewegter Objekte behandelt.

Als weitere Anwendungen für die vorgeschlagene Umgebungsrepräsentation werden eine Laserscanner-basierte Höhenschätzung für Brücken und eine Parklückendetektion und -vermessung beschrieben und evaluiert.

Environment Perception for Driver Assistance Systems Using Hierarchical Occupancy Grids

Summary

The number of casualties from road traffic is decreasing in Germany. Vehicle manufacturers contribute to safety by providing drivers with more and more complex driver assistance and safety systems, making driving safer and more relaxing.

A reliable and comprehensive environment perception is essential for modern driver assistance systems. To support as many different applications as possible with a single environment representation, a sensor-independent and application-independent consistent environment representation must be applied.

Occupancy grids enable free space analysis, road course estimation, detection of moving objects and self-localization among other things. Therefore, this thesis uses a three-dimensional occupancy grid to model the environment for driver assistance systems.

To allow for an efficient application of three-dimensional occupancy grids, a dynamic method to control the occupancy grid detail level is proposed. The desired resolution is controlled by the applications.

To complete the environment perception, a grid-based simultaneous self-localization and mapping approach for two and three dimensions as well as the detection and tracking of moving objects are described and evaluated.

In addition, exemplary applications for the proposed environment representation are characterized and analyzed: lidar-based height estimation for bridges and parking spot detection and measuring.

Inhaltsverzeichnis

Symbolverzeichnis	XI
1 Einleitung	1
1.1 Motivation	1
1.2 Anforderungen an die Umgebungserfassung	3
1.3 Einsatzgebiete für Belegungskarten	4
1.4 Zielsetzung	6
1.5 Aufbau der Arbeit	8
2 Grundlagen	9
2.1 Wahrscheinlichkeitstheorie	9
2.2 Zustandsschätzung	12
2.3 Bayes-Filter	14
2.3.1 Das binäre Bayes-Filter mit statischem Zustand	15
2.3.2 Das lineare Kalman-Filter	18
2.3.3 Das erweiterte Kalman-Filter	19
2.3.4 Sequentielle Monte-Carlo-Methoden	19
2.4 Umgebungserfassende Sensorik im automobilen Umfeld	23
2.4.1 Radarsensoren	23
2.4.2 Lidar Sensoren	24
2.4.3 Stereokamera	27
3 Umgebungsmodellierung durch hierarchische Belegungskarten	29
3.1 Motivation	30
3.2 Stand der Technik	31
3.3 Koordinatensysteme	37
3.3.1 Kartenkoordinatensystem	37
3.3.2 Fahrzeugkoordinatensystem	38
3.3.3 Sensorkoordinatensysteme	39
3.3.4 Koordinatentransformationen	40
3.4 Inverses Sensormodell	41
3.4.1 Messrauschen	41

3.4.2	Gemessener Freiraum	45
3.4.3	Motion-Scan-Effekt	48
3.5	Datenstruktur	49
3.5.1	Dünn besetzte Voxeligitter	49
3.5.2	Hierarchische Datenstrukturen	50
3.5.3	Kombination von Voxeligittern und hierarchischen Datenstrukturen	54
3.5.4	Zugriff auf die Datenstruktur	55
3.6	Schätzung des Belegungszustandes	58
3.6.1	Binäres Bayes-Filter	58
3.6.2	Evidenztheorie	59
3.6.3	Zeitliches Abklingen	61
3.7	Operationen auf der Datenstruktur	61
3.7.1	Verfeinern eines Kartenausschnittes	62
3.7.2	Reduktion der Auflösung eines Kartenausschnittes	62
3.7.3	Integration von Messungen	63
3.7.4	Löschen freier Bereiche	64
3.7.5	Mengenoperationen auf Karten	68
3.8	Bestimmung des Detaillierungsgrades der Belegungskarte	70
3.8.1	Detaillierung abhängig vom Einsatzbereich der Umgebungskarte	70
3.8.2	PreCrash-Funktionen	71
3.8.3	Navigation mit Punktlandmarken	72
3.8.4	Parkfunktionen	73
3.9	Evaluierung der Effizienz der Algorithmen	73
3.9.1	Mögliche Konfigurationen der Belegungskarte	73
3.9.2	Randbedingungen	74
3.9.3	Berechnungsaufwand	75
3.9.4	Speicherbedarf	80
3.10	Zusammenfassung und Ausblick	84
4	Simultane Selbstlokalisierung und Kartierung	89
4.1	Motivation	89
4.2	Stand der Technik	92
4.2.1	Kalman-SLAM	92
4.3	Schätzung des Fahrzeugzustandes	96
4.3.1	Zustandsraum	97
4.3.2	Prozessmodell	97
4.4	Odometrie	101
4.4.1	Modellierung einer Messung	101
4.4.2	Integration der Messungen	101
4.5	Rasterbasierte Selbstlokalisierung	102

4.5.1	Definition der Gütefunktion	102
4.5.2	Definition der Messung	103
4.5.3	Nichtlineare Optimierungsverfahren	105
4.6	Ergebnisse	109
4.6.1	Versuchsträger zur Realdatenauswertung	109
4.6.2	Bestimmung von Mess- und Prozessrauschen	110
4.6.3	Auswertung mit realen Messdaten	113
4.7	Dreidimensionaler Kartenabgleich	122
4.8	Zusammenfassung und Ausblick	124
5	Detektion und Verfolgung bewegter Objekte	127
5.1	Motivation	127
5.2	Stand der Technik	129
5.2.1	Objektbasierte Verfolgung mit Lidar-Sensoren	129
5.2.2	Detektion bewegter Objekte in Belegungskarten mit Lidar-Sensoren . .	130
5.2.3	Detektion bewegter Objekte mit weiteren Sensoren	132
5.2.4	Verfolgung bewegter Objekte in Belegungskarten	133
5.3	Detektion bewegter Objekte	134
5.3.1	Überblick	134
5.3.2	Die Messkarte	134
5.3.3	Die statische Belegungskarte	135
5.3.4	Belegungskarte mit dynamischen Messungen	136
5.3.5	Zusammenfassung	139
5.4	Verfolgung bewegter Objekte	140
5.4.1	Systemzustand	140
5.4.2	Bewegungsmodell	141
5.4.3	Zustandsschätzung	141
5.4.4	Partikelfilter	142
5.4.5	Kalman-Filter	148
5.4.6	Aktualisierung der statischen Karte	149
5.5	Auswertung	150
5.5.1	Szenario	150
5.5.2	Auswertung der Positionsschätzung	152
5.5.3	Auswertung der Geschwindigkeitsschätzung	153
5.5.4	Verfolgung mehrerer Objekte	154
5.6	Zusammenfassung und Ausblick	155

6	Weitere Applikationen	159
6.1	Höhenschätzung	160
6.1.1	Einleitung	160
6.1.2	Stand der Technik	160
6.1.3	Verfahren	160
6.1.4	Eingesetzter Versuchsträger: E-Klasse.	161
6.1.5	Ergebnisse	163
6.1.6	Variante: Höhenschätzung mit einzeiligem Laserscanner	166
6.2	Detektion und Vermessung von Parklücken	168
6.2.1	Einleitung	168
6.2.2	Stand der Technik	168
6.2.3	Ansatz zur Parklückendetektion	169
6.2.4	Adaption des Detaillierungsgrades für die Belegungskarte	176
6.2.5	Eingesetzter Versuchsträger: A-Klasse	176
6.2.6	Ergebnisse	178
6.3	Zusammenfassung und Ausblick	186
7	Zusammenfassung und Ausblick	187
A	Literaturverzeichnis	193
A.1	Verwendete Literatur	193
A.2	Eigene Publikationen und Patente	205
A.3	Betreute Abschlussarbeiten	205

Symbolverzeichnis

Abkürzungen

3D	Drei Dimensionen
ACC	Abstandsregeltempomat (engl. Adaptive Cruise Control)
ADAS	Modernes Fahrerassistenzsystem (engl. Advanced Driver Assistance System)
ADASIS	Schnittstellenspezifikationen für moderne Fahrerassistenzsysteme (engl. Advanced Driver Assistant Systems Interface Specifications)
ADMA	Automotive Dynamic Motion Analyzer
ADTF	Software zur Entwicklung von Fahrerassistenzsystemen (engl. Automotive Data and Time-Triggered Framework)
API	Programmierschnittstelle (engl. Application Programming Interface)
CA	Bewegungsmodell mit konstanter Beschleunigung (engl. Constant Acceleration)
CAN	Asynchroner serieller Feldbus (engl. Controller Area Network)
CT	Bewegungsmodell mit konstanter Kurvenfahrt (engl. Constant Turn)
CUDA	Programmierschnittstelle von NVidia (engl. Compute Unified Device Architecture)
CV	Bewegungsmodell mit konstanter Geschwindigkeit (engl. Constant Velocity)
DATMO	Detektion und Verfolgung von bewegten Objekten (engl. Detection and Tracking of Moving Objects)
DGPS	Globales Navigationssatellitensystem mit Differentialsignal (engl. Differential Global Positioning System)
DST	Evidenztheorie nach Dempster und Shafer (engl. Dempster-Shafer Theory)
EKF	Erweitertes Kalman-Filter
EMP	Frei (engl. Empty)
engl.	Englisch
ESP	Elektronisches Stabilitätsprogramm

FMCW	Frequenzmoduliertes Dauerstrichradar (engl. Frequency Modulated Continuous Wave Radar)
FPGA	Spezieller integrierter Schaltkreis (engl. Field Programmable Gate Array)
GNN	‘Globaler nächster Nachbar’-Assoziationsverfahren
GPS	Globales Navigationssatellitensystem (engl. Global Positioning System)
GPU	Grafikprozessor (engl. Graphics Processing Unit)
GSL	GNU Numerik-Bibliothek (engl. Gnu Scientific Library)
GSM	Standard für ein volldigitales Mobilfunknetz (engl. Global System for Mobile Communications)
ICP	Iterativer Nächster-Nachbar Algorithmus (engl. Iterative Closest Point Algorithm)
IMM	Paralleles Verfolgen unterschiedlicher Bewegungsmodelle (engl. Interacting Multiple Model)
IMU	Inertialmesseinheit (engl. Inertial Measurement Unit)
IPM	Inverse Perspective Mapping
JPDA	Vereinheitlichte probabilistische Datenassoziation (engl. Joint Probabilistic Data Association)
Lidar	Methode zur optischen Abstandsmessung (engl. Light Detection and Ranging)
LKW	Lastkraftwagen
LM	Landmarke
LOD	Detailierungsgrad (engl. Level of Detail)
OCC	Belegt (engl. Occupied)
PC	PreCrash-System
PDF	Wahrscheinlichkeitsdichtefunktion (engl. Probability Density Function)
PKW	Personenkraftwagen
PL	Parklücke
PMD	Photomischdetektor (engl. Photonic Mixer Device)
RMSE	Statistischer Gesamtfehler (engl. Root Mean Square Error)
ROI	Bereich von Interesse (engl. Region of Interest)
SAPOS	Satellitenpositionierungsdienst
SIR	Sequentielle Monte-Carlo Methode (engl. Sequential Importance Resampling)
SLAM	Simultane Selbstlokalisierung und Kartierung (engl. Simultaneous Self-Localization and Mapping)
SRR	Nahbereichsradar (engl. Short Range Radar)
TBD	Signalverfolgung mit anschließender Detektion (engl. Track-Before-Detect)
V2I	Fahrzeug-Infrastruktur Kommunikation (engl. Vehicle to Infrastructure Communication)

V2V	Fahrzeug-Fahrzeug Kommunikation (engl. Vehicle to Vehicle Communication)
WC	ungünstigster Fall (engl. Worst Case)

Konventionen

α_h	Horizontale Strahlaufweitung eines Laserstrahls
α_v	Vertikale Strahlaufweitung eines Laserstrahls
c	Konstante
\mathbf{C}_m	Kovarianzmatrix Messunsicherheit
c_{\max}	Definierte maximale Hierarchietiefe eines Octrees
$\text{cost}(\cdot)$	Kostenfunktion
\mathbf{C}	Kovarianzmatrix
d	Index eines Messpunktes
$\text{diag}(\cdot)$	Diagonalmatrix, es sind lediglich die Elemente der Hauptdiagonale angegeben
$\text{dist}(a, b)$	Euklidischer Abstand von a nach b
DYN	Klasse von Messpunkten von dynamischen Objekten
$h(\cdot)$	Gütefunktion, gibt die Überlagerung einer Messung mit der Belegungskarte an
\mathbf{I}	Einheitsmatrix
i	Allgemeiner Index, Index eines Blattknotens / einer Zelle
j	Index eines Octrees
\mathbf{K}	Belegungskarte
\mathbf{K}_{dyn}	Karte mit dynamischen Messungen
\mathbf{K}_{meas}	Messkarte
$\mathbf{K}_{\text{static}}$	Statische Belegungskarte
k	Index in der zeitdiskreten Betrachtung
l	Geradenmerkmal
l	Parklückenlänge
LOD	Menge von Funktionen, die den Detaillierungsgrad angeben
logit	Logarithmisches Quotenverhältnis (engl. Log-Odds-Ratio)
M	Anzahl an Partikeln
m	Zell- beziehungsweise Objektmittelpunkt
(m)	Index eines Partikels
$\mathcal{N}(\mu, \sigma)$	Normalverteilung mit Mittelwert μ und Standardabweichung σ
\mathbb{N}	Menge der natürlichen Zahlen
n	Knoten in einem Octree
\mathbf{N}	Menge von Knoten eines Octrees
\mathcal{O}	Menge von Octrees
O	Octree

P	Eckpunkt einer detektierten Parklücke
\mathcal{P}	Menge aller quaderförmigen Segmente, die eine Kontur einer Belegungskarte beschreiben
\mathbf{p}	Punkt im kartesischen Koordinatensystem
p	Wahrscheinlichkeit $p(X = x)$, dass Zufallsvariable X Ereignis x annimmt
Q	Umschreibender Quader um ein Objekt
q	Gewichtungsfaktor
R	Rotationsmatrix
\mathbb{R}	Menge der reellen Zahlen
r	Radialer Abstand
S	Quaderförmiges Segment, das einen Teil der Kontur einer Belegungskarte beschreibt
s	Schrittweite
STAT	Klasse von Messpunkten von statischen Objekten
T	Teilbaum eines Octrees
Δt	Zeitdifferenz
V	Volumen
\mathcal{V}	Dünn besetztes Voxelgitter, das Octrees als Elemente enthält
v	Geschwindigkeit
v_{rad}	Radialgeschwindigkeit
w	Parklückenbreite
\mathcal{X}	Menge von Partikeln
χ	Partikel
\mathbf{x}_F	Pose, umfasst Position und Orientierung

Griechische Buchstaben

η	Normierungsfaktor
μ	Mittelwert
φ	Azimutwinkel
Φ	Wankwinkel
$\dot{\Phi}$	Wankrate
Ψ	Gierwinkel
$\dot{\Psi}$	Gierrate
σ	Standardabweichung
θ	Elevationswinkel
Θ	Nickwinkel
$\dot{\Theta}$	Nickrate

Kalman-Filter

B	Eingangsmatrix
F	Zustandsübergangsmatrix
<i>f</i>	Zustandsübergangsfunktion
G	Beobachtungsmatrix
<i>g</i>	Beobachtungsfunktion
J	Jacobi-Matrix
K	Kalman Verstärkungsmatrix
P	Prozessfehler-Kovarianzmatrix
\mathbf{P}_k^*	A priori Prozessfehler-Kovarianzmatrix
\mathbf{P}_k	A posteriori Prozessfehler-Kovarianzmatrix
Q	Kovarianzmatrix Prozessrauschen
R	Kovarianzmatrix Messrauschen
S	Kovarianzmatrix Residuum
<i>s</i>	Prozessrauschen
u	Steuervektor
w	Messrauschen
x	Systemzustand
$\hat{\mathbf{x}}$	Schätzung des Systemzustandes
$\hat{\mathbf{x}}_k^*$	A priori Systemzustand
$\hat{\mathbf{x}}_k$	A posteriori Systemzustand
y	Messung
$\bar{\mathbf{y}}$	Residuum

Koordinatensysteme

ECEF	Erdfestes kartesisches Koordinatensystem (engl. Earth Centered Earth Fixed)
ENU	ENU-Koordinatensystem (engl. East-North-Up)
FK	Fahrzeugkoordinatensystem
KK	Kartenkoordinatensystem
OK	Objektkoordinatensystem
PK	Parklückenkoordinatensystem
SK	Sensorkoordinatensystem

1

Einleitung

1.1 Motivation

Die Zahl der im Straßenverkehr Verunglückten in Deutschland ist rückläufig. Dies ist zum einen der Politik zu verdanken, die durch Regularien und Forschungsprojekte einwirkt. Zum anderen haben Automobilhersteller ein Interesse, ihre Fahrzeuge so sicher wie möglich zu gestalten. Die Daimler AG beispielsweise verfolgt die Vision vom unfallfreien Fahren in naher Zukunft [20].

Als Hauptursache von Unfällen in Deutschland im Jahr 2010 hat das statistische Bundesamt Fahrfehler der Fahrzeugführer ermittelt. Diese bestehen neben nicht angepasster Geschwindigkeit im Wesentlichen aus Fehlern beim Abbiegen, Wenden, Rückwärtsfahren sowie beim Ein- und Anfahren [118]. Fahrfehler entstehen hauptsächlich, wenn Fahrer in Eile sind, unter Stress stehen, übermüdet sind oder einen mangelnden Überblick über die Verkehrssituation haben. Technische Mängel der Fahrzeuge spielen lediglich eine untergeordnete Rolle.

Automobilhersteller stellen dem Fahrer deshalb mehr und mehr Assistenzsysteme zur Verfügung, um die Sicherheit im Straßenverkehr zu erhöhen. Dabei gibt es zwei unterschiedliche Arten von Systemen: Komfortsysteme wie zum Beispiel die Verkehrszeichenerkennung oder den Nachtsicht-Assistenten, die zu einem vorausschauenden und entspannten Fahren beitragen sollen und Sicherheitssysteme, deren Ziel es ist, drohende Gefahren abzuwenden.

Sicherheitssysteme im Fahrzeug können wiederum in zwei Klassen eingeteilt werden: Passive Sicherheitssysteme haben zum Ziel, die Folgen von drohenden oder unvermeidbaren Unfällen zu reduzieren. Aktive Sicherheitssysteme versuchen durch einen aktiven Eingriff Unfälle zu verhindern oder Unfallfolgen zu mindern. Ein aktives Sicherheitssystem im erweiterten Sinn stellt das autonome Fahren dar: Hierbei übernimmt das Fahrzeug selbstständig alle Fahraufgaben. Fahrfehler des Menschen werden somit ausgeschlossen.

Passive Sicherheit

Das Ziel der passiven Sicherheit ist es, die Schwere von Unfällen oder deren Folgen zu verringern. Den Anfang der passiven Sicherheit stellt die Erfindung des Sicherheitsgurts 1903 dar. In einem Fahrzeugmodell von Mercedes-Benz wurde 1959 erstmals eine Sicherheitskarosserie realisiert, die Einführung des Airbags 1981 ist ein weiterer Meilenstein für die passive Sicherheit [117].

Für passive Sicherheitssysteme wie den Gurtstraffer oder den Airbag ist lediglich der Zustand des eigenen Fahrzeuges relevant: Wird eine starke Verzögerung sowie eine Verformung der Frontpartie des Fahrzeuges detektiert, lösen diese Systeme aus.

PreCrash-Systeme gehen einen Schritt weiter. Hier erfolgt eine Reduktion der Unfallschwere über das Auslösen von reversiblen Rückhaltesystemen zu einem besonders günstigen Zeitpunkt [55], nach Möglichkeit deutlich vor der eigentlichen Kollision. Ein im Rahmen des Experimental-Sicherheitsfahrzeuges 2009 der Daimler AG betrachtetes System ist „Pre-Safe Pulse Plus“: Hier werden die Insassen bei einer drohenden seitlichen Kollision aus dem Gefahrenbereich gerückt, um einen möglichst großen Abstand zum Unfallgegner zu schaffen [15]. Um diese Systeme möglich zu machen, ist eine zuverlässige Erfassung des Umfelds des Fahrzeuges erforderlich.

Aktive Sicherheit und Komfortsysteme

Ein prominentes Beispiel der Systeme, die aktiv in das Geschehen eingreifen, ist der Abstandsregeltempomat ACC (engl. Adaptive Cruise Control). Dieser führt das Fahrzeug selbstständig entlang der Längsachse, basierend auf einer Erfassung der Umgebung. Ein weiteres Beispiel ist der Fußgängerschutz. Die Daimler AG arbeitet an einem Ausweichassistenten, der je nach Situation automatisch ausweicht oder eine Vollbremsung durchführt, wenn ein Fußgänger plötzlich auf die Straße tritt [15].

Die Funktion aktiver Sicherheitssysteme basiert im Wesentlichen auf dem Wissen über die Umgebung des eigenen Fahrzeuges. Hierzu wird mit einem oder mehreren Sensoren die Umge-

bung des Fahrzeugs erfasst und analysiert, so dass das Fahrzeug über alle relevanten Objekte in der Umgebung informiert ist. Bei potentiellen Gefahrensituationen kann der Fahrer damit rechtzeitig gewarnt werden; er hat mehr Zeit zum Handeln. Zudem lassen sich im Fahrzeug Maßnahmen treffen, um die Auswirkungen einer möglichen Kollision zu entschärfen oder eine Kollision ganz zu verhindern, beispielsweise durch rechtzeitiges Abbremsen des Fahrzeuges. Ein Überblick über Fahrerassistenzsysteme wird in [145] gegeben.

Autonomes Fahren

Das autonome Fahren im privaten individuellen Straßenverkehr ist noch eine Zukunftsvision. Beim autonomen Fahren übernimmt das Fahrzeug selbst alle Fahraufgaben. Wenn solche Systeme ausreichend abgesichert werden können, wären diese in der Lage die Sicherheit im Straßenverkehr weiter zu erhöhen: Immerhin werden menschliche Fahrfehler als Fehlerquelle ausgeschlossen. Die Daimler AG lässt Fahrtests heute schon automatisiert durchführen, weil menschliche Fahrer sicherheitskritische Versuche nicht reproduzierbar fahren können [16].

Als nächster Zwischenschritt zum vollständigen autonomen Fahren ist die automatische Querführung denkbar, das heißt das Folgen des Fahrbahnverlaufs sowie das automatische Ein- und Ausscheren bei Überholvorgängen. Autonome Fahrzeuge wären von besonderem Interesse für junge oder ältere Fahrzeugführer, die ansonsten noch nicht oder nicht mehr fahren können. Aber auch alle anderen Fahrzeugführer könnten ihre Zeit besser nutzen, wenn sie sich nicht mehr auf den Verkehr konzentrieren müssen. Für autonome Fahrzeuge ist eine vollständige Erfassung rund um das eigene Fahrzeug unerlässlich.

1.2 Anforderungen zukünftiger Fahrerassistenz- und Sicherheitssysteme an die Umgebungserfassung

Fahrerassistenz- und Sicherheitssysteme werden zunehmend komplexer, weshalb die Anforderungen an solche Systeme wachsen. Insbesondere ist eine zuverlässige und vollständige Erfassung der Umgebung für moderne Fahrerassistenz- und Sicherheitssysteme grundlegend.

Bei Versuchsträgern oder Fahrzeugen, die bei Wettbewerben wie der DARPA Urban Challenge (siehe beispielsweise [69]) zum autonomen Fahren zum Einsatz kommen, werden meist hochgenaue Sensoren verwendet, die aufgrund der Bauform oder anderen Einschränkungen nicht immer serientauglich sind. Im Gegensatz dazu sollen Seriensensoren kompakt, verdeckt verbaubar, zuverlässig und kostengünstig sein. Aufgrund der unterschiedlichen Anforderungen und Randbedingungen müssen bei Seriensensoren in Bezug auf die Genauigkeit und das

Messrauschen gegenüber prototypischen High-End Sensoren teilweise Kompromisse eingegangen werden.

Die Situation, in der sich das eigene Fahrzeug befindet, soll mit Hilfe der Umgebungserfassung möglichst exakt modelliert werden. Dazu zählt die Kenntnis des Zustandes aller Verkehrsteilnehmer in der Umgebung, die Abmaße aller statischen Hindernisse, der Fahrbahnverlauf sowie die Position und Orientierung (Pose) des eigenen Fahrzeugs relativ zu diesen Objekten. Weitere Anforderungen an die Umgebungserfassung sind Zuverlässigkeit und Ausfallsicherheit, welche durch redundante Systeme und komplementäre Sensorik erfüllt werden können.

1.3 Einsatzgebiete für Belegungskarten

Bei Belegungskarten (engl. Occupancy Grids) wird die Umgebung des Fahrzeugs in der Regel durch ein Raster in Zellen gleicher Größe unterteilt. Für jede Zelle des Rasters wird auf Basis von Sensormessungen geschätzt, ob diese belegt ist oder nicht. Somit kann die Umgebung des Fahrzeugs je nach Zellgröße unterschiedlich fein repräsentiert werden. Eine Belegungskarte kann für unterschiedlichste Anwendungen im Bereich der Fahrerassistenz- und Sicherheitssysteme eingesetzt werden, die im Folgenden beschrieben werden.

Freiraumanalyse

In einer Belegungskarte, in der Hindernisse der Umgebung abgelegt sind, lässt sich der Freiraum ermitteln. Dieser dient unter anderem zur Pfadplanung bei autonomen Fahraufgaben oder zur Bestimmung einer Ausweichtrajektorie bei drohenden Kollisionen. Die Freiraumanalyse wird in Abschnitt 6.2.3 eingehend behandelt.

Selbstlokalisierung

Bei unzureichender Kenntnis der Lage des Sensorfahrzeuges beziehungsweise der gefahrenen Strecke seit dem letzten Messzeitpunkt kann eine Belegungskarte zur Lokalisierung genutzt werden. Dazu gibt es mehrere Verfahren: Die rasterbasierte Selbstlokalisierung, die merkmalsbasierte Selbstlokalisierung und die Registrierung der Umgebungskarte auf eine digitale Karte. In der Literatur (unter anderem in [116]) werden auch Verfahren zum Loop-Closing beschrieben. Dabei wird die erstellte Karte korrigiert, indem ein bereits in der Vergangenheit vermessenes Objekt mit der erneuten Messung desselben Objektes in Übereinklang gebracht wird. Mit einer dreidimensionalen Karte ist zusätzlich die Ermittlung des Nick- und Wankwinkels möglich. Durch die Selbstlokalisierung lassen sich Odometriedaten stützen, vor allem

in Bereichen, in denen keine oder nur ungültige GPS-Positionsinformationen verfügbar sind. Die Selbstlokalisierung wird ausführlich in Kapitel 4 behandelt.

Fahrbahnverlaufsschätzung

Zur Feststellung der Relevanz von Objekten ist es hilfreich zu wissen, ob sich diese auf der Fahrbahn befinden. Diese Aufgabe wird häufig dadurch gelöst, dass die Randbebauung in einer Belegungskarte modelliert wird. In dieser können dann Vorhersagen über den Fahrbahnverlauf getroffen werden. Durch eine Kombination mit einer digitalen Karte kann die Reichweite der Vorhersagen in der Regel deutlich erhöht werden. Weitere Hinweise und Literaturstellen finden sich in Abschnitt 4.2.1.

Höhenschätzung

Häufig werden zweidimensionale Belegungskarten betrachtet. Die Verwendung von dreidimensionalen Belegungskarten hat jedoch einen deutlichen Vorteil: Hier bilden sich auch Strukturen wie Brücken korrekt ab. Die Schätzung der Höhe einer solchen über die Fahrbahn hängenden Struktur wird möglich. Mit diesem Wissen kann eine Brücke sicher von Hindernissen auf der Fahrbahn unterschieden werden. Auf die Höhenschätzung wird in Abschnitt 6.1 eingegangen. Weiterhin kann durch die dreidimensionale Modellierung auch auf die Überfahrbarkeit von Zielen geschlossen werden. Niedrige Bordsteine sind beispielsweise überfahrbar, zeigen sich in einer zweidimensionalen Karte aber unter Umständen als Hindernis.

Detektion und Verfolgung bewegter Objekte

Mit Hilfe einer Belegungskarte lassen sich ebenso bewegte Objekte detektieren. In der Literatur werden im Wesentlichen zwei Ansätze unterschieden: Die Detektion bewegter Objekte über Artefakte, die in der Karte entstehen [141] und die Detektion durch die Bildung einer Differenzkarte [76]. Ebenso ist die Detektion bewegter Objekte über eine Modellierung durch die Evidenztheorie nach Dempster und Shafer (DST) möglich, siehe Moras [87]. Ziel der Detektion bewegter Objekte ist einerseits, Artefakte bewegter Objekte aus der Belegungskarte zu entfernen. Andererseits ist die Verfolgung bewegter Objekte für viele Funktionen essenziell. In der Belegungskarte lassen sich bewegte Objekte verfolgen durch eine Registrierung von dynamisch klassifizierten Zellen. Über diese Registrierung lassen sich lokale Objektkarten aufbauen, die der Bestimmung der exakten Außenkontur von bewegten Objekten dienen. Kapitel 5 widmet sich diesem Thema ausführlich.

1.4 Zielsetzung

Es gibt Fahrerassistenzsysteme, bei denen alle Komponenten in einem Gehäuse integriert sind: Vom Sensor über die Signalverarbeitung bis zur Abbildung der Funktion. Solche Systeme sind für einfache Funktionen geeignet, bei denen wenige Sensoren genutzt werden. Beispielsweise wird in [113] ein Spurwechselassistent beschrieben, bei dem die Warnindikatoren direkt vom Sensor angesteuert werden. Komplexere Funktionen sind in der Umgebungserfassung jedoch meist anspruchsvoller, es wird häufig ein erweiterter Sichtbereich und/oder komplementäre Sensorik benötigt, um steigende Sicherheitsstandards zu erfüllen. Um Redundanz zu erhalten und um den meist eingeschränkten Sichtbereich eines Sensors zu erhöhen, werden in der Regel mehrere Sensoren eingesetzt. Dies führt zu dem Problem, dass die Daten unterschiedlicher Sensoren zu einem konsistenten Gesamtbild zusammengeführt werden müssen. Die Fusion der Daten mehrerer Sensoren wird typischerweise in einer zentralen Einheit durchgeführt.

Umgekehrt sollten unterschiedliche Funktionen alle notwendigen Informationen aus dem Umgebungsmodell extrahieren können, das durch diese zentrale Einheit zur Verfügung gestellt wird. Im Gegensatz zu mehreren anwendungsspezifischen Modellen können durch eine solche zentrale Einheit Rechenzeit und Speicher, die Kosten bedeuten, eingespart werden. Der Gewinn ist umso größer, je mehr Funktionen bedient werden können. Des Weiteren entsteht Einsparpotenzial, wenn dieselbe zentrale Einheit zur Umgebungserfassung durch ihre Flexibilität in mehreren Baureihen eingesetzt werden kann und mit unterschiedlichen Sensorkonfigurationen zusammenarbeitet.

Um die in Abschnitt 1.2 gestellten Anforderungen zu erfüllen wird vorgeschlagen, ein System zur umfassenden dreidimensionalen Umgebungserfassung zu realisieren, mit dem möglichst viele Fahrerassistenzfunktionen abgedeckt werden können. Dieses soll so generisch und flexibel wie möglich sein. Weiterhin sollte es nicht stark von Applikationen abhängig sein, das heißt der Einsatz von applikationsspezifischen Schwellwerten und Modellen sollte weitestgehend vermieden werden. Hierzu bieten sich dreidimensionale Belegungskarten an, wie im vorigen Abschnitt 1.3 aufgezeigt. Belegungskarten sind sehr gut geeignet, um unbekannte Umgebungen modellfrei in hoher Genauigkeit darzustellen.

Komplementär ausgelegte Sensoren und eine Erfassung der Umgebung des Fahrzeuges in 360° bedingen einen hohen Speicherbedarf und eine hohe Rechenzeit. Dabei steigen der Speicherbedarf und die Rechenzeit naturgemäß mit der Detailtreue, mit der die Umgebung modelliert wird. Je nach Anwendung muss die Umgebung jedoch nicht in jedem Punkt mit dem gleichen Detaillierungsgrad dargestellt werden. Für ein PreCrash-System genügt es, den Bereich direkt um das eigene Fahrzeug exakt zu modellieren. Für einen Parkassistenten ist es ausreichend, Parklücken exakt zu vermessen. In Bereichen, die für momentan aktive Funktionen nicht relevant sind, kann demnach Speicher und Rechenzeit eingespart werden, indem der Detaillierungsgrad reduziert wird. Mit anderen Worten ergeben sich Rechenzeit- und Spei-

chervorteile, wenn die Bereiche der Umgebung, die für Anwendungen relevant sind, bekannt sind und bei der Umgebungsmodellierung in höherer Auflösung berücksichtigt werden als andere Bereiche. Hierzu fließt lediglich ein Modell für potenziell relevante Bereiche ein, es wird jedoch nicht die gesamte Umgebung durch einfache geometrische Modelle wie Quader abstrahiert. Die Modellierung der Umgebung ist gitterbasiert diskret, so dass auch unvorhersehbare Gegebenheiten repräsentiert werden können.

Ziel der vorliegenden Arbeit ist es, eine auf Belegungskarten basierende Umgebungserfassung zu realisieren. Hierbei kann Anwendungswissen in die Umgebungsmodellierung einfließen, indem die Auflösung der Belegungskarte lokal dynamisch angepasst wird. Um die Unabhängigkeit des Ansatzes von speziellen Sensoren zu demonstrieren, werden beispielhaft unterschiedliche Sensoren zur Umgebungserfassung angebunden: Laserscanner, Radarsensoren und eine Stereokamera. Als demonstrationsfähige Anwendungen für die Umgebungserfassung werden eine robuste Selbstlokalisierung, die Schätzung der Höhe von Brücken, die Detektion und Vermessung von Parklücken sowie die Detektion und Verfolgung bewegter Objekte umgesetzt.

Abbildung 1.1 zeigt die hierzu vorgeschlagene Architektur. Basierend auf den Distanzprofilen beispielsweise eines Laserscanners werden simultan eine Selbstlokalisierung, eine Detektion bewegter Objekte und eine Kartierung durchgeführt. Hierzu ist die genaue Pose des Sensorfahrzeugs erforderlich. Zur Ermittlung der Pose des Sensorfahrzeuges kann entweder eine Schätzung der Inertialsensorik verwendet werden oder Odometriedaten. Die Selbstlokalisierung und die Unterscheidung zwischen statischen und dynamischen Objekten bilden einen Kreislauf. Für Lokalisierungsalgorithmen sind bewegte Objekte störend, diese müssen also bekannt sein, um entfernt werden zu können. Umgekehrt ist für die Detektion bewegter Objekte die exakte Pose und Dynamik des Sensorfahrzeuges zu kennen.

In dieser Arbeit wird dieses Problem iterativ gelöst: Die Selbstlokalisierung liefert trotz der Artefakte bewegter Objekte eine Positionsschätzung. Mit dem so geschätzten Zustand des Sensorfahrzeugs kann die Detektion bewegter Objekte durchgeführt werden. Um die Ergebnisse zu verbessern, wird diese Vorgehensweise iterativ wiederholt. Die Detektion bewegter Objekte liefert als Ergebnis zum einen eine statische Karte, die zur Freiraumanalyse und Pfadplanung verwendet werden kann. Zum anderen liefert sie Hinweise auf bewegte Objekte, die in einem Verfahren zur Zustandsschätzung verfolgt werden. Als Besonderheit kann die Anwendung vorgeben, welche Bereiche der Umgebung von besonderem Interesse sind. Hier wird der Detaillierungsgrad dynamisch erhöht, während er in allen anderen Bereichen gering gehalten wird um Rechenzeit und Speicherbedarf zu optimieren.

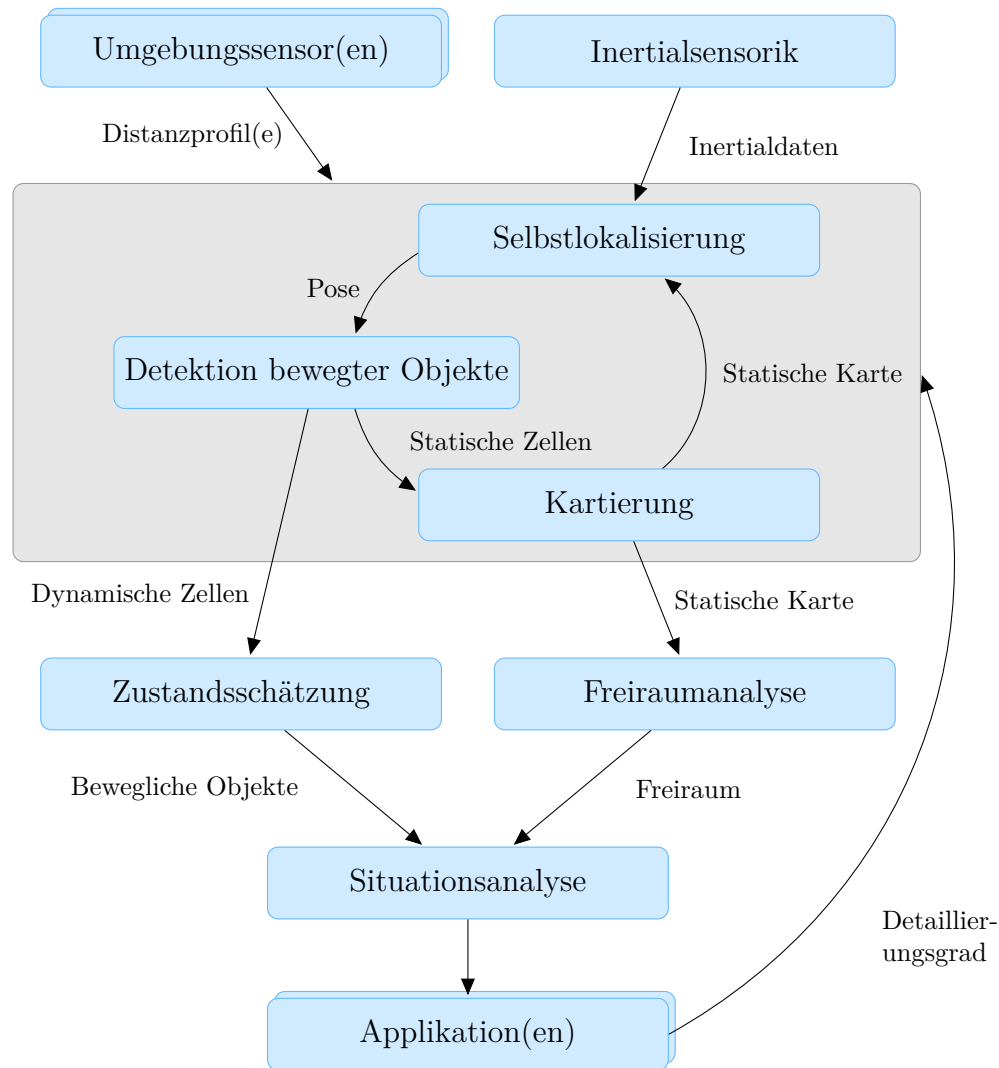


Abbildung 1.1 – Vorgeschlagene Architektur zur Umgebungserfassung mit hierarchischen Belegungskarten. Als Besonderheit wird der Detaillierungsgrad der Umgebungsmodellierung durch die Applikation gesteuert.

1.5 Aufbau der Arbeit

In Kapitel 2 werden theoretische Grundlagen erläutert. Anschließend werden in Kapitel 3 der Aufbau und die Funktionsweise der dreidimensionalen hierarchischen Belegungskarte eingeführt. Essentiell für den korrekten Aufbau einer Belegungskarte ist die Kenntnis über die Pose des Sensorfahrzeugs. In Kapitel 4 werden deshalb mögliche Lokalisierungsansätze für Belegungskarten aufgezeigt und diskutiert. Da Belegungskarten nur statische Objekte korrekt beschreiben, wird die Detektion und die Verfolgung dynamischer Objekte in Kapitel 5 behandelt. Unterschiedliche Experimente und deren Auswertung werden in Kapitel 6 vorgestellt. Abschließend folgen in Kapitel 7 eine Zusammenfassung der Arbeit und ein Ausblick.

2

Grundlagen

In diesem Kapitel werden die Grundlagen der in den folgenden Kapiteln vorgestellten Verfahren erläutert und die Notation eingeführt. Abschnitt 2.1 führt in die Wahrscheinlichkeitstheorie ein. Abschnitt 2.2 beschreibt die Probleme der Zustandsschätzung im Allgemeinen, Abschnitt 2.3 fasst unterschiedliche Verfahren zur Zustandsschätzung zusammen, die alle auf dem Satz von Bayes basieren. Abschließend werden in Abschnitt 2.4 Sensoren vorgestellt, die im Kontext Fahrerassistenz- und Sicherheitssysteme beziehungsweise Referenzsysteme zur Umgebungserfassung eingesetzt werden.

2.1 Wahrscheinlichkeitstheorie

In diesem Abschnitt werden Grundlagen zur Wahrscheinlichkeitstheorie und die Notation eingeführt.

Sei X eine *Zufallsvariable* und x ein Ereignis, das X annehmen kann. Dann bezeichnet

$$p(X = x) \tag{2.1}$$

die *Wahrscheinlichkeit*, dass X das Ereignis x annimmt. Wenn eindeutig ist, um welche Zufallsvariable es sich handelt, so wird $p(X = x)$ zu $p(x)$ abgekürzt.

Sowohl diskrete als auch kontinuierliche Wahrscheinlichkeiten summieren sich zu Eins:

$$\sum_x p(x) = 1 \quad (2.2)$$

$$\int p(x) dx = 1 \quad (2.3)$$

Insbesondere summieren sich die Wahrscheinlichkeiten, dass Ereignis x und dessen *Gegeneignis* \bar{x} eintritt, zu Eins. Daher gilt:

$$p(\bar{x}) = 1 - p(x) \quad (2.4)$$

Schätzverfahren arbeiten typischerweise in einem kontinuierlichen Raum. Hier besitzen Zufallsvariablen eine *Wahrscheinlichkeitsdichtefunktion* (PDF, engl. Probability Density Function). Ein Beispiel für eine PDF ist die eindimensionale *Normalverteilung* $\mathcal{N}(\mu, \sigma)$ mit dem Erwartungswert μ und der Standardabweichung σ :

$$p(x) = (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2} \frac{(x - \mu)^2}{\sigma^2}\right\}$$

Die mehrdimensionale Normalverteilung wird auch *multivariate Normalverteilung* genannt. Sie ist gegeben durch:

$$p(\vec{x}) = \det(2\pi\mathbf{C})^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\vec{x} - \vec{\mu})^T \mathbf{C}^{-1}(\vec{x} - \vec{\mu})\right\} \quad (2.5)$$

Hier steht $\vec{\mu}$ für den mehrdimensionalen Erwartungswert und \mathbf{C} für die Kovarianzmatrix. Die *Verbundwahrscheinlichkeit* zweier Zufallsvariablen X und Y ist definiert durch:

$$p(x, y) = p(X = x \text{ und } Y = y) \quad (2.6)$$

Wenn X und Y statistisch unabhängig sind, gilt:

$$p(x, y) = p(x) p(y) \quad (2.7)$$

Die *bedingte Wahrscheinlichkeit* wird folgendermaßen notiert:

$$p(x|y) = p(X = x \text{ gegeben, dass } Y = y) \quad (2.8)$$

und steht für die Wahrscheinlichkeit, dass x eintritt, unter der Bedingung, dass Ereignis y bereits eingetreten ist.

Für $p(y) \neq 0$ ist die bedingte Wahrscheinlichkeit wie folgt definiert:

$$p(x|y) = \frac{p(x, y)}{p(y)} \quad (2.9)$$

Sind X und Y unabhängig, gilt demnach:

$$p(x|y) = \frac{p(x)p(y)}{p(y)} = p(x) \quad (2.10)$$

Mit diesen Definitionen können zwei Sätze aufgestellt werden. Der *Satz der totalen Wahrscheinlichkeit* lautet:

$$p(x) = \sum_y p(x|y) p(y) \quad (\text{diskreter Fall}) \quad (2.11)$$

$$p(x) = \int p(x|y) p(y) dy \quad (\text{kontinuierlicher Fall}) \quad (2.12)$$

Wegen $p(x, y) = p(y, x)$ folgt aus Gleichung (2.9) der *Satz von Bayes*. Er stellt unter der Voraussetzung $p(y) \neq 0$ eine Beziehung zwischen $p(x|y)$ und $p(y|x)$ her und wird meist in der folgenden Form geschrieben:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \quad (2.13)$$

Der Nenner in (2.13) hängt nicht von x ab, deshalb wird $p(y)^{-1}$ häufig verkürzend als Normalisierungsfaktor η geschrieben:

$$p(x|y) = \eta p(y|x)p(x) \quad (2.14)$$

Ein *Box-Whisker-Plot* dient der kompakten Darstellung der robusten Streuungsmaße einer Zufallsvariablen. Abbildung 2.1 zeigt den Box-Whisker-Plot einer 10000-elementigen Stichprobe der Zufallsvariablen $X \sim \mathcal{N}(0, 1)$. Der rote Querstrich kennzeichnet den Median der Stichprobe. Das blaue Rechteck wird durch das untere und das obere Quartil begrenzt und enthält demnach 50% der Stichproben. Die gestrichelte schwarze Linie wird durch den 1,5-

fachen Interquartilsabstand begrenzt. Dies entspricht bei einer normalverteilten Stichprobe annähernd $\pm 2,7\sigma$, beziehungsweise einer Abdeckung von 99,3% der Stichprobe. Außerhalb liegende Stichproben werden als Ausreißer aufgefasst und als rote „+“-Symbole gekennzeichnet.

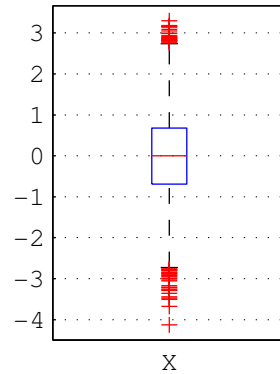


Abbildung 2.1 – Boxplot einer 10000-elementigen Stichprobe einer normalverteilten Zufallsvariablen mit $\mu = 0$ und $\sigma = 1$.

Der *statistische Gesamtfehler* RMSE (engl. Root Mean Square Error) eines Schätzwertes $\hat{\mathbf{x}}$ ist definiert als:

$$\text{RMSE}(\hat{\mathbf{x}}) = \sqrt{E((\hat{\mathbf{x}} - \mathbf{x})^2)} \quad (2.15)$$

Hierbei bezeichnet \mathbf{x} den wahren Wert und E den Erwartungswert.

Ist der wahre Wert nicht bekannt, kann mit folgender Formel auch der Fehler zwischen zwei Schätzwerten $\hat{\mathbf{x}}_1 = (x_{1,1}, \dots, x_{1,n})$ und $\hat{\mathbf{x}}_2 = (x_{2,1}, \dots, x_{2,n})$ bestimmt werden:

$$\text{RMSE}(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2) = \sqrt{\frac{\sum_{i=1}^n (x_{1,i} - x_{2,i})^2}{n}} \quad (2.16)$$

2.2 Zustandsschätzung

Auf Basis des Satzes von Bayes kann der Zustand von Systemen mittels Beobachtungen geschätzt werden. Sowohl der Systemzustand als auch Messungen werden dann als Wahrscheinlichkeiten aufgefasst: Ein Zustand \mathbf{x} wird durch die Wahrscheinlichkeit $p(\mathbf{x})$ und eine Messung \mathbf{y} durch $p(\mathbf{y})$ repräsentiert. Wenn von einer Messung \mathbf{y} auf den Zustand \mathbf{x} eines Systems geschlossen werden kann, dann heißt $p(\mathbf{x})$ die a priori Wahrscheinlichkeit und $p(\mathbf{x}|\mathbf{y})$ die a posteriori Wahrscheinlichkeit.

Die Schätzung $\hat{\mathbf{x}}$ des Wertes von \mathbf{x} , welche die Wahrscheinlichkeit $p(\mathbf{x}|\mathbf{y})$ maximiert, wird „Maximum a posteriori“-Schätzung genannt. Umgekehrt wird die Maximierung der Wahrscheinlichkeit $p(\mathbf{y}|\mathbf{x})$ als „Maximum Likelihood“-Schätzung bezeichnet. Diese beiden Wahrscheinlichkeiten können durch den Satz von Bayes ineinander überführt werden.

Der Zustand \mathbf{x} beschreibt die Eigenschaften eines Systems zu einem bestimmten Zeitpunkt. Falls sich diese Eigenschaften über die Zeit ändern heißt das System dynamisch, andernfalls statisch. Dynamische Systeme können in lineare und nichtlineare Systeme aufgeteilt werden. Steuerdaten, die das System beeinflussen, werden in dem Steuervektor \mathbf{u} zusammengefasst. Die resultierenden Eigenschaften des Systems sind meist nicht alle direkt schätzbar. Letztere können aber über zeitliche Zusammenhänge zwischen aufeinanderfolgenden Messungen \mathbf{y} beobachtet werden.

Die rekursive Änderung eines Systemzustands in einer zeitdiskreten Betrachtung kann über die Zustandsübergangsfunktion f formalisiert werden als:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{s}_{k-1}). \quad (2.17)$$

Der diskrete Zeitindex wird hierbei mit k bezeichnet, das Prozessrauschen mit \mathbf{s} . Eine Messung kann formalisiert werden als:

$$\mathbf{y}_k = g(\mathbf{x}_k, \mathbf{w}_k). \quad (2.18)$$

Hier steht g für die Messgleichung und \mathbf{w} für das Messrauschen. In einem linearen System mit additivem Rauschen lauten diese Gleichungen:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{s}_{k-1} \quad (2.19)$$

$$\mathbf{y}_k = \mathbf{G}\mathbf{x}_k + \mathbf{w}_k \quad (2.20)$$

dabei bezeichnet \mathbf{F} die Zustandsübergangsmatrix, \mathbf{B} die Eingangsmatrix und \mathbf{G} die Messmatrix.

Das Ziel der Zustandsschätzung ist es, die bestmögliche Schätzung $\hat{\mathbf{x}}_k$ für \mathbf{x}_k zu bestimmen, gegeben eine Menge von Messungen $\mathbf{y}_{1:k}$ und eine Sequenz von Steuervektoren $\mathbf{u}_{1:k-1}$. Hierbei bezeichnet $\mathbf{x}_{1:k}$ eine Serie von aufeinanderfolgenden Zuständen zu den Zeitpunkten $1, \dots, k$.

Mit der Wahrscheinlichkeitstheorie kann der Zustandsübergang als Wahrscheinlichkeitsdichtefunktion modelliert werden:

$$p(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}) \quad (2.21)$$

Wenn der Zustand \mathbf{x}_k alle Informationen aus der Vergangenheit beinhaltet, also der nächste Zustand \mathbf{x}_{k+1} lediglich von \mathbf{x}_k abhängt, so wird dieser Prozess als Markov-Kette bezeichnet. Der Zustand \mathbf{x}_k hängt dann nur vom vorherigen Zustand \mathbf{x}_{k-1} und dem vorherigen Steuervektor \mathbf{u}_{k-1} ab. Eine Messung \mathbf{y}_k ist unabhängig von vorangegangenen Messungen, sie hängt nur vom aktuellen Systemzustand ab. Ein solches System kann als dynamisches Bayes-Netzwerk dargestellt werden, siehe Abbildung 2.2. Diese Abbildung verdeutlicht, dass die Informationen der vorherigen Messungen $\mathbf{y}_{1:k-1}$ und vorangegangener Steuervektoren $\mathbf{u}_{1:k-2}$ in \mathbf{x}_{k-1} enthalten sind. Daher kann die PDF (2.21) wie folgt vereinfacht werden:

$$p(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}). \quad (2.22)$$

Die PDF (2.22) wird auch Zustandsübergangswahrscheinlichkeitsdichtefunktion genannt. Die Wahrscheinlichkeit $p(\mathbf{y}_k | \mathbf{x}_{0:k}, \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1})$ wird auch Messwahrscheinlichkeitsdichtefunktion genannt. Diese kann ebenso vereinfacht werden:

$$p(\mathbf{y}_k | \mathbf{x}_{0:k}, \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}) = p(\mathbf{y}_k | \mathbf{x}_k). \quad (2.23)$$

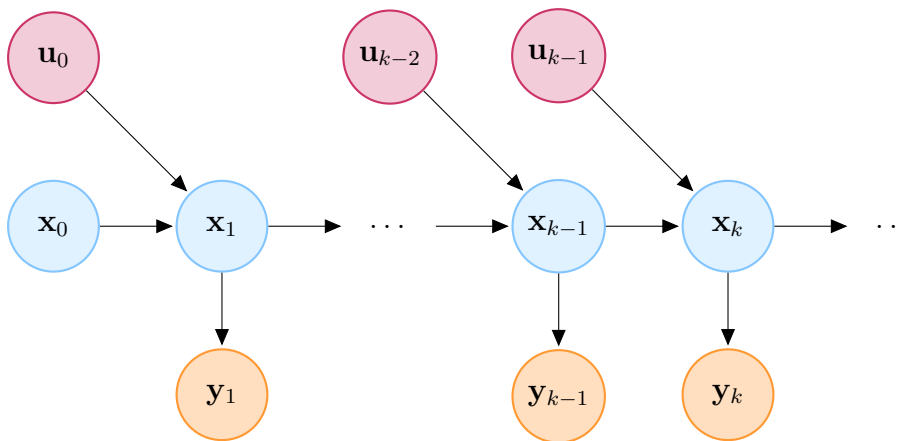


Abbildung 2.2 – Dynamisches Bayes-Netzwerk.

2.3 Bayes-Filter

Die rekursive Bayessche Zustandsschätzung, auch genannt Bayes-Filter, ist ein allgemeiner Zustandsschätzer, der auf der Wahrscheinlichkeitstheorie basiert. Ein Bayes-Filter schätzt eine PDF.

Die *a posteriori* Wahrscheinlichkeitsdichteverteilung des Systemzustands \mathbf{x}_k zum Zeitpunkt k ist definiert als die bedingte Wahrscheinlichkeit des Zustandes unter der Voraussetzung, dass die vorangegangenen Messungen und Steuervektoren bekannt sind:

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}, \mathbf{u}_{1:k-1}). \quad (2.24)$$

Diese PDF beinhaltet die Informationen der aktuellen Messung. Zur Zustandsschätzung wird hingegen häufig eine Wahrscheinlichkeit benötigt, bevor die aktuelle Messung \mathbf{y}_k einfließt. Diese Verteilung wird auch *a priori* PDF oder Prädiktion genannt und wie folgt definiert:

$$p(\mathbf{x}_k | \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}). \quad (2.25)$$

Die Innovation, also die Aktualisierung durch eine Messung, wird durchgeführt, indem die *a posteriori* PDF aus der *a priori* PDF berechnet wird. Dazu wird der prädizierte Zustand mit der tatsächlichen Messung verglichen. Die mathematische Formulierung für das Bayes-Filter besteht aus zwei Teilen, aus der Prädiktion und aus der Innovation.

Prädiktion:

$$p(\mathbf{x}_k | \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-2}) d\mathbf{x}_{k-1} \quad (2.26)$$

Innovation:

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}, \mathbf{u}_{1:k-1}) = \eta p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}) \quad (2.27)$$

Für eine mathematische Herleitung sei auf [130] verwiesen.

2.3.1 Das binäre Bayes-Filter mit statischem Zustand

Ein Beispiel für ein Bayes-Filter ist das binäre Bayes-Filter mit statischem Zustand. Dieses kommt bei der Schätzung der Belegungswahrscheinlichkeit der in Kapitel 3 beschriebenen Belegungskarte zum Einsatz. Binär heißt in diesem Fall, dass der zu schätzende Zustand lediglich zwei Werte annehmen kann: Null oder Eins. Dabei wird angenommen, dass sich dieser Zustand nicht ändert. Deshalb erhält \mathbf{x} keinen zeitlichen Index:

$$\mathbf{x} \in \{0, 1\} \quad (2.28)$$

Die folgende Herleitung ist an Thrun [130] angelehnt. Da der Zustand statisch ist, hängt die Schätzung zum Zeitpunkt k lediglich von den bisherigen Messungen ab:

$$p(\mathbf{x}|\mathbf{y}_{1:k}, \mathbf{u}_{1:k-1}) = p(\mathbf{x}|\mathbf{y}_{1:k}) \quad (2.29)$$

Der Ausdruck (2.29) kann mit Hilfe des Satzes von Bayes (2.13) wie folgt umgestellt werden:

$$p(\mathbf{x}|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}, \mathbf{y}_{1:k-1}) p(\mathbf{x}|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \quad (2.30)$$

Eine Einzelmessung \mathbf{y}_k ist nur vom Zustand \mathbf{x} abhängig, nicht jedoch von vorangegangenen Messungen, deshalb gilt:

$$p(\mathbf{y}_k|\mathbf{x}, \mathbf{y}_{1:k-1}) = p(\mathbf{y}_k|\mathbf{x}) \quad (2.31)$$

Damit lässt sich (2.30) vereinfachen zu:

$$p(\mathbf{x}|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}) p(\mathbf{x}|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \quad (2.32)$$

Der Ausdruck $p(\mathbf{y}_k|\mathbf{x})$ wird *Messmodell* genannt. Die Anwendung des Satzes von Bayes (2.13) auf das Messmodell ergibt:

$$p(\mathbf{y}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y}_k) p(\mathbf{y}_k)}{p(\mathbf{x})} \quad (2.33)$$

Diese Gleichung kann dann eingesetzt werden in (2.32):

$$p(\mathbf{x}|\mathbf{y}_{1:k}) = \frac{p(\mathbf{x}|\mathbf{y}_k) p(\mathbf{y}_k) p(\mathbf{x}|\mathbf{y}_{1:k-1})}{p(\mathbf{x}) p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \quad (2.34)$$

Es gilt analog für das Gegenereignis:

$$p(\bar{\mathbf{x}}|\mathbf{y}_{1:k}) = \frac{p(\bar{\mathbf{x}}|\mathbf{y}_k) p(\mathbf{y}_k) p(\bar{\mathbf{x}}|\mathbf{y}_{1:k-1})}{p(\bar{\mathbf{x}}) p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \quad (2.35)$$

Durch eine Division von (2.34) durch (2.35) kürzen sich schwierig zu berechnende Faktoren heraus:

$$\begin{aligned}
\frac{p(\mathbf{x}|\mathbf{y}_{1:k})}{p(\bar{\mathbf{x}}|\mathbf{y}_{1:k})} &= \frac{p(\mathbf{x}|\mathbf{y}_k)}{p(\bar{\mathbf{x}}|\mathbf{y}_k)} \frac{p(\mathbf{x}|\mathbf{y}_{1:k-1})}{p(\bar{\mathbf{x}}|\mathbf{y}_{1:k-1})} \frac{p(\bar{\mathbf{x}})}{p(\mathbf{x})} \\
&= \frac{p(\mathbf{x}|\mathbf{y}_k)}{1 - p(\mathbf{x}|\mathbf{y}_k)} \frac{p(\mathbf{x}|\mathbf{y}_{1:k-1})}{1 - p(\mathbf{x}|\mathbf{y}_{1:k-1})} \frac{1 - p(\mathbf{x})}{p(\mathbf{x})}
\end{aligned} \tag{2.36}$$

Die Schätzung des Zustandes kann durch die Repräsentation des *Quotenverhältnisses* durchgeführt werden. Das Quotenverhältnis des Zustandes \mathbf{x} ist definiert durch:

$$\frac{p(\mathbf{x})}{p(\bar{\mathbf{x}})} = \frac{p(\mathbf{x})}{1 - p(\mathbf{x})} \tag{2.37}$$

Das *logarithmische Quotenverhältnis* logit eines Zustandes \mathbf{x} ist weiterhin definiert durch:

$$\text{logit}(\mathbf{x}) := \log \frac{p(\mathbf{x})}{1 - p(\mathbf{x})} \tag{2.38}$$

Aus dem Quotenverhältnis $\text{logit}(\mathbf{x})$ erhält man die Wahrscheinlichkeit durch folgende Formel:

$$p(\mathbf{x}) = 1 - \frac{1}{1 + \exp\{\text{logit}(\mathbf{x})\}} \tag{2.39}$$

Damit kann das logarithmische Quotenverhältnis von (2.36) gebildet werden:

$$\begin{aligned}
\text{logit}_k(\mathbf{x}) &= \log \frac{p(\mathbf{x}|\mathbf{y}_k)}{1 - p(\mathbf{x}|\mathbf{y}_k)} + \log \frac{p(\mathbf{x}|\mathbf{y}_{1:k-1})}{1 - p(\mathbf{x}|\mathbf{y}_{1:k-1})} + \log \frac{1 - p(\mathbf{x})}{p(\mathbf{x})} \\
&= \log \frac{p(\mathbf{x}|\mathbf{y}_k)}{1 - p(\mathbf{x}|\mathbf{y}_k)} + \text{logit}_{k-1}(\mathbf{x}) - \log \frac{p(\mathbf{x})}{1 - p(\mathbf{x})}
\end{aligned} \tag{2.40}$$

Der initiale Zustand wird mit logit_0 bezeichnet:

$$\text{logit}_0(\mathbf{x}) = \log \frac{p(\mathbf{x})}{1 - p(\mathbf{x})} \tag{2.41}$$

Damit lautet die rekursive Formulierung für das binäre Bayes-Filter mittels logarithmischem Quotenverhältnis:

$$\text{logit}_k(\mathbf{x}) = \log \frac{p(\mathbf{x}|\mathbf{y}_k)}{1 - p(\mathbf{x}|\mathbf{y}_k)} + \text{logit}_{k-1}(\mathbf{x}) - \text{logit}_0(\mathbf{x}) \tag{2.42}$$

2.3.2 Das lineare Kalman-Filter

Das Kalman-Filter [68, 142] ist ein rekursives Verfahren um den Zustand eines linearen Systems zu schätzen. Die Schätzung basiert sowohl auf der Modellierung als auch auf Beobachtungen des Prozesses. Das Kalman-Filter ist ein Bayes-Filter; der Zustand wird geschätzt indem rekursiv jeweils ein Prädiktions- und ein Innovationschritt durchgeführt werden. Der Prozessfehler wird in einer Kovarianzmatrix \mathbf{P} , der Prozessfehlerkovarianzmatrix geschätzt. Der geschätzte Zustand $\hat{\mathbf{x}}$ und die Prozessfehlerkovarianzmatrix \mathbf{P} repräsentieren die a posteriori PDF, wie sie für das Bayes-Filter in 2.26 definiert wurde. Das Kalman-Filter ist ein optimales Filter, das den mittleren quadratischen Fehler minimiert.

In den Zustandsgleichungen (2.19) und (2.20) für ein lineares System wird beim Kalman-Filter sowohl das Prozessrauschen \mathbf{s}_k als auch das Messrauschen \mathbf{w}_k als weiß, mittelwertfrei, normalverteilt und gegenseitig unkorreliert angenommen:

$$\mathbf{s}_k \sim \mathcal{N}(0, \mathbf{Q}_k) \quad (2.43)$$

$$\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{R}_k) \quad (2.44)$$

Die Kovarianzmatrix \mathbf{Q}_k stellt das Prozessrauschen dar, wogegen die Kovarianzmatrix \mathbf{R}_k das Messrauschen beschreibt. Zur Vereinfachung werden \mathbf{Q} und \mathbf{R} wie in [142] als konstant angenommen.

Mit diesen Annahmen können die Formeln für das Kalman-Filter aufgestellt werden, Abbildung 2.3 stellt diese übersichtlich grafisch dar. Hierbei bezeichnet \mathbf{K} die Kalman-Verstärkungsmatrix, $\bar{\mathbf{y}}$ das Residuum, \mathbf{S} die Kovarianzmatrix des Residuums und \mathbf{I} die Einheitsmatrix.

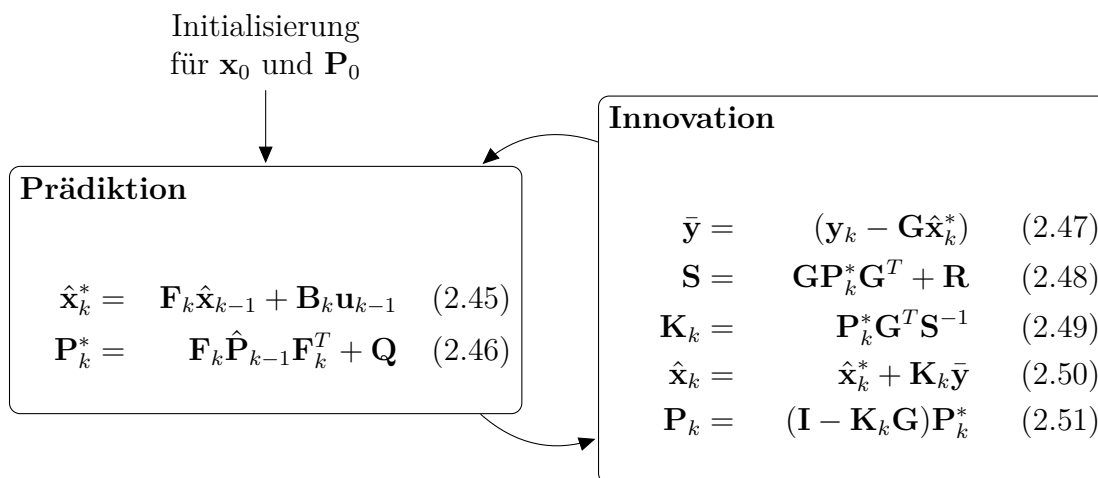


Abbildung 2.3 – Das Kalman-Filter.

Um bei der Implementierung numerischen Problemen vorzubeugen, die durch die Subtraktion in (2.51) auftreten können, wird stattdessen die Joseph-Form verwendet:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{G}) \mathbf{P}_k^* (\mathbf{I} - \mathbf{K}_k \mathbf{G})^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \quad (2.52)$$

2.3.3 Das erweiterte Kalman-Filter

Mit dem linearen Kalman-Filter, das im letzten Abschnitt beschrieben wurde, können lediglich lineare Prozesse abgebildet werden. Um nichtlineare Prozesse beschreiben zu können, kann das Kalman-Filter zum so genannten erweiterten Kalman-Filter (EKF) ergänzt werden.

Kann der Zustandsübergang nicht linear durch eine Matrix \mathbf{F} (2.19) beschrieben werden, so wird dieser über eine nichtlineare Zustandsübergangsfunktion f durchgeführt. Diese kann zwar genutzt werden, um den Zustand zu präzisieren (2.45), sie kann jedoch nicht als Matrix dargestellt werden um die Prozessfehlerkovarianzmatrix (2.46) zu präzisieren. Das Gleiche gilt für die Messgleichung g . Beide Funktionen können jedoch über Jacobi-Matrizen approximiert werden. Für die Zustandsübergangs- und die Messmatrix gilt dann:

$$\mathbf{F}_k = \frac{\partial f_k}{\partial \mathbf{x}_k} \quad (2.53)$$

$$\mathbf{G}_k = \frac{\partial g_k}{\partial \mathbf{x}_k} \quad (2.54)$$

Durch die Approximierung ist das EKF kein optimaler Schätzer. Wenn die Funktionen f und g im Punkt \mathbf{x} nicht gut linearisiert werden können, kann ein nichtlineares Schätzproblem besser durch das Unscented Kalman-Filter gelöst werden. Hierfür sei auf Julier [65] verwiesen.

2.3.4 Sequentielle Monte-Carlo-Methoden

Analytische Methoden wie das Kalman-Filter stellen für nichtlineare Systeme lediglich eine suboptimale Lösung dar. Einen alternativen Ansatz stellen numerische Verfahren dar, bei denen die Stützstellenwahl für die Zustandsschätzung deterministisch oder stochastisch erfolgt. Bei deterministischen Methoden wird der Zustandsraum diskret abgebildet. Eberhardt behandelt beispielsweise die Zustandsschätzung basierend auf einer Baumstruktur [29]. Für eine Übersicht sei auf Bergman [6] verwiesen.

Stochastische Simulationsverfahren wie die sequentiellen Monte-Carlo-Verfahren haben den Vorteil, dass die Stichproben aus den Bereichen des Zustandsraums gezogen werden, die

hauptsächlich zur Monte-Carlo-Integration beitragen. Die Monte-Carlo-Integration wird im folgenden Abschnitt eingeführt.

Monte-Carlo-Integration

Monte-Carlo-Methoden zur numerischen Integration betrachten Integrale der Form

$$I = \int_{\mathbb{R}^n} f(x) \pi(x) dx, \quad (2.55)$$

dabei ist $\pi(x)$ eine nicht-negative Funktion, deren Integral 1 ergibt:

$$\int_{\mathbb{R}^n} \pi(x) dx = 1. \quad (2.56)$$

Die Eigenschaften von π lassen es zu, π als PDF zu betrachten. Im Kontext der Bayes-Filterung ist π die a posteriori PDF: $\pi(\mathbf{x}) = p(\mathbf{x}|\mathbf{y})$. Eine weitere Annahme für Monte-Carlo-Methoden ist, dass aus $f(x)$ eine große Anzahl M an Stichproben $\chi^{(m)}$, $m = 1, \dots, M$, gezogen werden kann, deren Verteilung der Wahrscheinlichkeitsdichte von $\pi(x)$ entsprechen. Die Monte-Carlo Approximation für das Integral (2.55) wird als Mittelwert aller Stichproben $\chi^{(m)}$ bestimmt:

$$f_M = \frac{1}{M} \sum_m f(\chi^{(m)}) \quad (2.57)$$

Für $M \rightarrow \infty$ geht der Approximierungsfehler gegen Null. Gegenüber anderen numerischen Integrationsverfahren hat die Monte-Carlo Integration den Vorteil, dass der Schätzfehler unabhängig von der Dimension des Zustands ist [6].

Zur Bayesschen Zustandsschätzung wird die Monte-Carlo-Integration sequentiell angewandt, daher der Name sequentielle Monte-Carlo-Methoden. Die einzelnen Stichproben, mit denen die PDF approximiert wird, werden auch „Partikel“ genannt und das Filter zur Zustandsschätzung daher *Partikel-Filter*. Wie beim Bayes-Filter ist das Ziel des Partikel-Filters, die a posteriori Wahrscheinlichkeit $p(\mathbf{x}_k|\mathbf{y}_k)$ zu schätzen, wobei diese Wahrscheinlichkeit rekursiv aus $p(\mathbf{x}_{k-1}|\mathbf{y}_{k-1})$ hervorgeht. Ein Partikel χ entspricht einer diskreten Stichprobe, die einen möglichen Zustand des Systems repräsentiert. Die Notation für die Menge aller Partikel \mathcal{X} zum Zeitpunkt k ist:

$$\mathcal{X}_k := \{\chi_k^{(m)} : m \in [1, M]\} \quad (2.58)$$

Sequential Importance Resampling

Beim sequentiellen gewichteten Ziehen von Stichproben, genannt SIR-Partikel-Filter (engl. Sequential Importance Resampling) [26], wird eine Verteilung $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ durch eine Menge von M gewichteten Partikeln approximiert:

$$\left\{ \left(q_k^{(m)}, \chi_k^{(m)} \right) : m \in [1, M] \right\} \quad (2.59)$$

Das Gewicht eines Partikels wird mit $q_k^{(m)}$ bezeichnet und es gilt:

$$\sum_m q_k^{(m)} = 1 \quad (2.60)$$

Der Algorithmus für das SIR-Partikel-Filter lautet:

1. Resampling: Ziehe für alle $m \in M$ einen Index i mit Wahrscheinlichkeit $q_k^{(m)}$ aus $[1, M]$ und ersetze $\chi_k^{(m)}$ durch $\chi_k^{(i)}$. Partikel mit hohem Gewicht werden also häufiger gezogen.
2. Prädiziere für alle $m \in M$ Partikel m und füge Rauschen hinzu, indem aus der Zustandsübergangsverteilung gezogen wird:

$$\chi_k^{(m)*} \sim p\left(\mathbf{x}_k | \chi_{k-1}^{(m)}, \mathbf{u}_{k-1}\right) \quad (2.61)$$

3. Führe für alle $m \in M$ eine Gewichtung gemäß dem Sensormodell durch:

$$q_k^{(m)} = p\left(\mathbf{y}_k | \chi_k^{(m)*}\right) \quad (2.62)$$

4. Normalisiere das Gewicht $q_k^{(m)}$ für alle $m \in M$ um (2.60) zu erfüllen:

$$q_k^{(m)} = \frac{q_k^{(m)}}{\sum_i q_k^{(i)}} \quad (2.63)$$

Die Funktionsweise des Algorithmus wird exemplarisch in Abbildung 2.4 veranschaulicht. Die einzelnen Schritte des Algorithmus sind von oben nach unten dargestellt. Der Algorithmus kann auch ohne Schritt 1 (Resampling) eingesetzt werden. Das zugehörige Filter nennt sich Sequential Importance Sampling (SIS)-Filter. Dieses neigt jedoch dazu mit der Zeit zu degenerieren und damit die a posteriori Wahrscheinlichkeitsdichteverteilung nicht mehr gut zu approximieren [25], und wird daher hier nicht weiter betrachtet.

Hol [61] vergleicht mehrere Ansätze zum Resampling und beschreibt das Systematic Resampling als zu bevorzugende Variante aufgrund der geringen Komplexität und der hohen Qua-

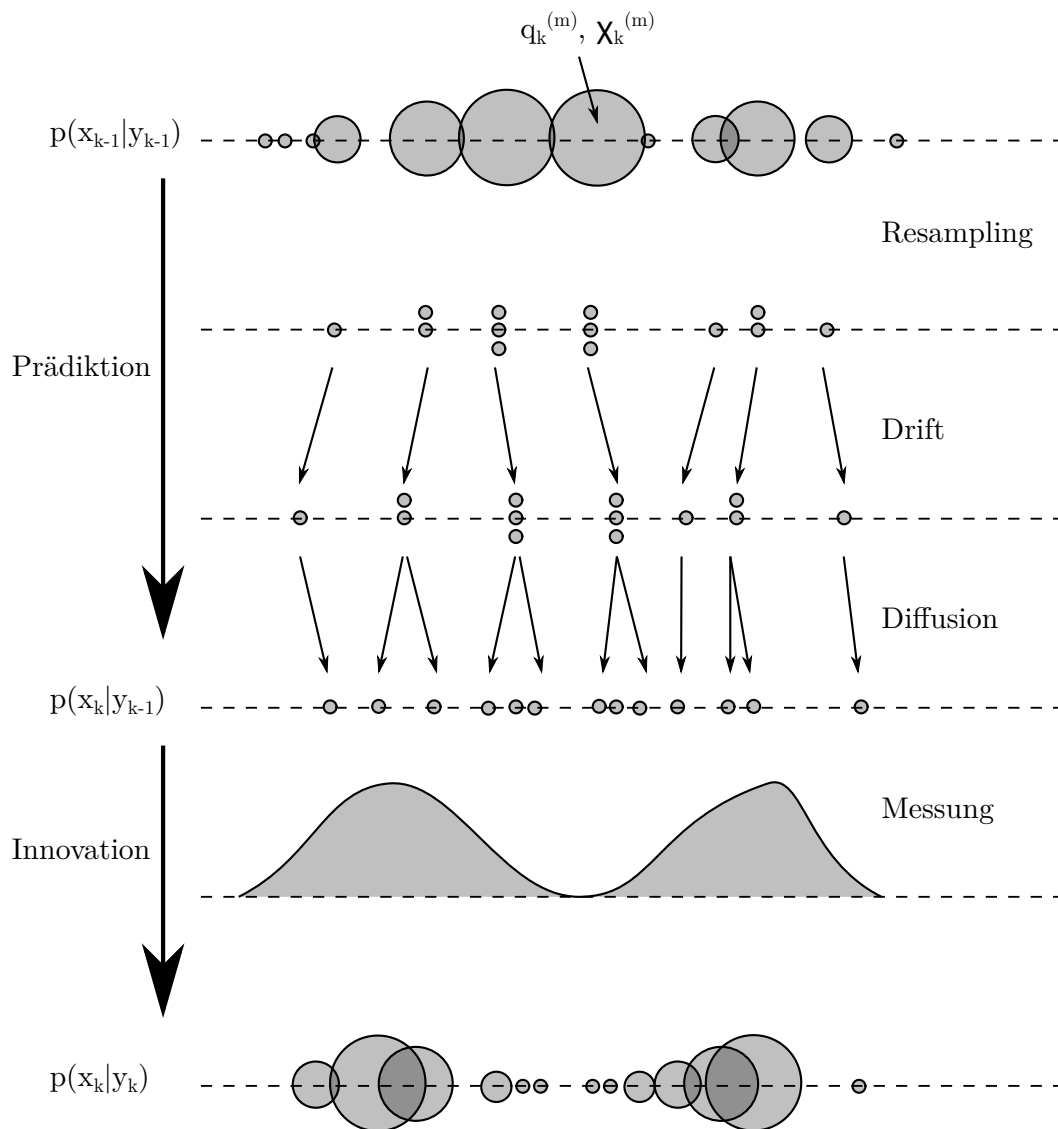


Abbildung 2.4 – Grafische Veranschaulichung des SIR-Partikel-Filters. Von oben nach unten sind die einzelnen Schritte des Algorithmus dargestellt.

lität der resultierenden Partikelmenge. Beim Systematic Resampling wird eine Zufallszahl \tilde{u} gleichverteilt aus dem Intervall $(0, 1]$ gezogen. Aus dieser werden systematisch weitere Zahlen u_m erzeugt, welche Partikel aus der durch die Gewichtungen gegebenen Verteilungsfunktion auswählen:

$$u_m = \frac{(m-1) + \tilde{u}}{M}, \quad m = 1, \dots, M \quad (2.64)$$

Anschließend wird für jedes m das Partikel $\chi_k^{(m)}$ durch das Partikel $\chi_k^{(i)}$ ersetzt, so dass gilt: $u_m \in \left[\sum_{s=1}^{i-1} q_s, \sum_{s=1}^i q_s \right)$. Das Systematic Resampling wird grafisch in Abbildung 2.5 veranschaulicht.

Das SIR-Partikel-Filter in dieser Form ist auch unter dem Namen „Condensation-Algorithmus“ [64] oder „Bootstrap Filter“ bekannt.

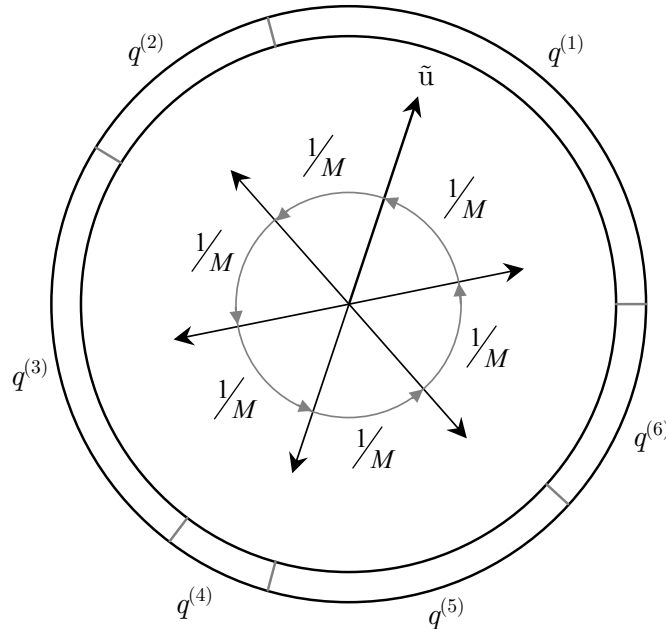


Abbildung 2.5 – Grafische Veranschaulichung des Systematic Resampling nach [8]. Es wird eine Zufallszahl \tilde{u} bestimmt, alle weiteren Zahlen u_m werden effizient durch die fortlaufende Addition von $\frac{1}{M}$ bestimmt. Die so ermittelten Werte dienen der gewichteten Auswahl von Partikeln. In diesem Beispiel wird $\chi^{(1)}$ verdoppelt, wogegen $\chi^{(6)}$ verworfen wird.

2.4 Umgebungserfassende Sensorik im automobilen Umfeld

In diesem Abschnitt werden Sensoren zur Umgebungserfassung beschrieben, die im Kontext Fahrerassistenz-, Sicherheits- und Referenzsysteme relevant sind. Insbesondere werden die in dieser Arbeit verwendeten Sensoren vorgestellt und deren Messprinzip erläutert.

2.4.1 Radarsensoren

Nahbereichsradare

Bei den eingesetzten Nahbereichsradaren (SRR, engl. Short Range Radar) handelt es sich um Sensoren, die auch in den aktuellen Baureihen der Daimler AG zum Einsatz kommen.

Tabelle 2.1 – Spezifikation der Nahbereichsradare

Eigenschaft	Nahbereichsradar
Prinzip	Pulse-Doppler Amplituden-Monopuls
Scanfrequenz	25 Hz
Sichtbereich	80°
Winkelgenauigkeit	$\pm 5^\circ \dots 10^\circ$
Reichweite	0.2 – 30 m
Genauigkeit der Abstandsmessung	$\pm 5 \text{ cm} \dots 7.5 \text{ cm}$

Die Nahbereichsradare arbeiten mit dem Amplituden-Monopulsverfahren zur Winkelmessung. Aus einem ausgesendeten Messimpuls werden durch zwei seitlich versetzte Antennen zwei Empfangssignale gebildet. Durch den Vergleich der Amplitude der Empfangssignale in beiden Kanälen kann ein Winkel berechnet werden, der sich aus der lateralen Ablage eines Objektes ergibt. Die Frequenz liegt bei 24,125 GHz und die Bandbreite beträgt 5 GHz. Eine detaillierte Sensorspezifikation ist in Tabelle 2.1 angegeben. Eine weiterführende Beschreibung ist bei Wenger [144] zu finden.

2.4.2 Lidar Sensoren

Lidar (engl. Light Detection and Ranging)-Sensoren sind aktive Sensoren zur Distanzmessung. Die Ermittlung der Distanz erfolgt über die Messung der Zeitdifferenz zwischen dem Senden und dem Empfangen eines Lichtpulses, der an einem Objekt reflektiert wird. Das Licht, das emittiert wird, liegt in der Regel im nahen Infrarot-Bereich bei 905 nm.

Lidar-Sensoren lassen sich in zwei Kategorien einteilen: in Mehrstrahl-Lidar und in Laserscanner. Bei einem Mehrstrahl-Lidar messen mehrere feststehende Dioden parallel, während bei einem Laserscanner die Lichtpulse einer oder mehrerer Dioden durch eine geeignete Mechanik die Umgebung überstreichen, um einen größeren Sichtbereich zu erhalten. In der Praxis kann durch die Ablenkung der Lichtpulse durch einen Spiegel ein Sichtbereich von bis zu 240° erzielt werden. Mit einem Laserscanner, bei dem die Laserdioden in einem rotierenden Gehäuse angebracht sind, lässt sich ein Sichtbereich von 360° erreichen. Bei einem Laserscanner erfolgen die einzelnen Messungen leicht zeitversetzt.

Beim Einsatz von Lidar- und Radarsensoren im automobilen Umfeld sollte beachtet werden, dass Nickbewegungen des Sensorfahrzeuges, die unter anderem bei Bremsvorgängen entstehen, ausgeglichen werden müssen. Dazu kann ein Laserscanner mit mehreren übereinanderliegenden Scanebenen eingesetzt werden. Bei einzeiligen Laserscannern oder Lidar-Sensoren

sollte die vertikale Strahlaufweitung groß genug sein, um Nickbewegungen ausgleichen zu können. In Böhlau [9] wird ein einzeliger Laserscanner mit einer vertikalen Strahlaufweitung von 3° beschrieben. Es folgt ein Überblick über Lidar-Sensoren, die in der Forschung verwendet werden.

Hella IDIS 1.0

Dieser Lidar-Sensor mit 16 fest angeordneten Dioden wurde für ACC-Funktionen entwickelt, siehe Abbildung 2.6a. Neben den Spezifikationen in Tabelle 2.2 finden sich mehr Details in [124].



(a) IDIS 1.0 Lidar



(b) IDIS Scanner

Abbildung 2.6 – Hella Lidar-Sensoren, Quelle: Hella

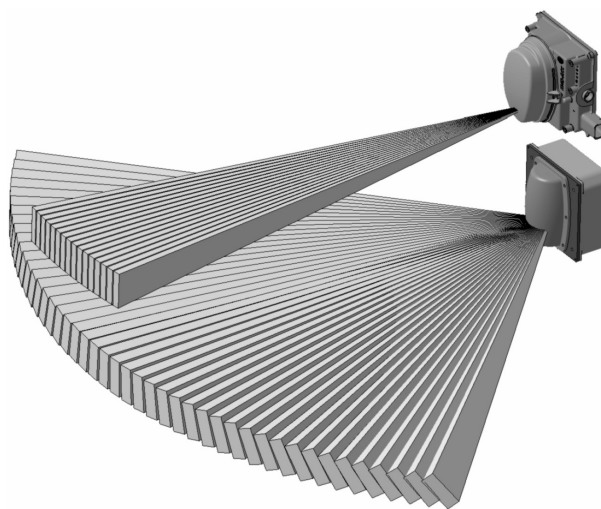


Abbildung 2.7 – Strahlgeometrie der Hella Sensoren: oben IDIS 1.0 und unten IDIS Scanner, Quelle: Böhlau [9].

Hella IDIS Scanner

Der einzeilige Hella IDIS Laserscanner arbeitet mit einer Diode und einem rotierenden Spiegel, um die Lichtpulse abzulenken. Dabei entsteht ein horizontaler Sichtbereich von bis zu 170° . Die Strahlgeometrie ist dabei variabel von einer äquidistanten Auflösung von einem Grad bis zu einer Konfiguration mit einer Auflösung von 2° und 4° am Randbereich. Abbildung 2.7 zeigt die Strahlgeometrie des Scanners im Vergleich zum IDIS 1.0. Deutlich zu erkennen ist der höhere Erfassungsbereich und die kippenden Strahlkeulen im Randbereich. Die vertikale Aufweitung beträgt 3° und ist damit in gewissem Rahmen geeignet, Nickbewegungen des Fahrzeuges auszugleichen. Eine sehr detaillierte Beschreibung über den Aufbau und die Funktionsweise findet sich in [9]. Ebenso wurden Tests durchgeführt, welche die Tauglichkeit für den Einsatz im Automobilbereich zeigen.

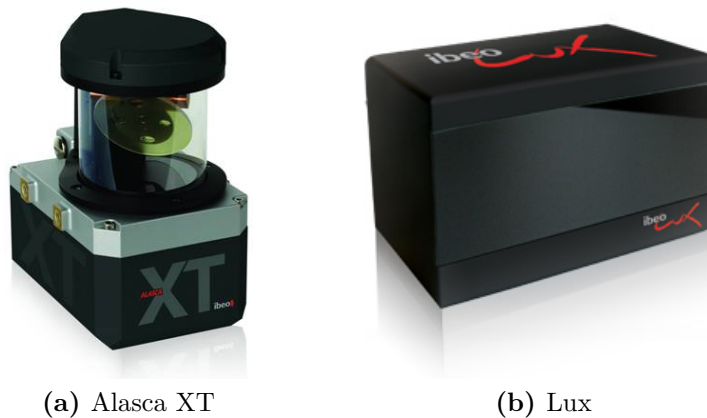


Abbildung 2.8 – Ibeo Laserscanner, Quelle: Ibeo

Ibeo Alasca XT

Der Laserscanner Alasca XT von Ibeo besitzt vier übereinander liegende Scanebenen, um die Nickempfindlichkeit zu reduzieren. Die Scanebenen haben einen vertikalen Abstand von $0,8^\circ$, so dass vertikal insgesamt ein Sichtbereich von $3,2^\circ$ abgedeckt wird. Analog zum Hella IDIS Scanner kippen die Strahlkeulen zunehmend zum Rand des horizontalen Sichtbereiches auf Grund des Messprinzips, so dass sich der vertikale Sichtbereich reduziert. Abbildung 2.8a zeigt den Scanner ohne Schutzgehäuse. Die technischen Spezifikationen sind in Tabelle 2.2 aufgelistet. Weitere Details und eine Beschreibung der Funktionsweise des Scanners finden sich in [72, 78].

Ibeo LUX

Der Ibeo LUX ist eine Weiterentwicklung des Alasca XT, jedoch mit kompakterem Gehäuse und einem daraus resultierenden verminderten Sichtbereich von 110° [63]. Die kompaktere Bauweise soll die Integration in ein Fahrzeug erleichtern. Die Spezifikation findet sich ebenso in Tabelle 2.2, Abbildung 2.8b zeigt den Scanner. Ein baugleicher Sensor wird auch von der SICK AG unter dem Namen SICK LD-MRS für Industrieanwendungen vertrieben.

Velodyne HDL-64E

Der Velodyne High Definition Lidar (Abbildung 2.9) ist ein 3D-Laserscanner, der durch seine hohe Auflösung genaue Distanzprofile mit Rundumsicht (360°) erzeugt [131]. Durch die präzise Vermessung der Umgebung eignet er sich sehr gut als Referenzsensor für automotivtaugliche Sensoren. Die Spezifikation findet sich in Tabelle 2.2.

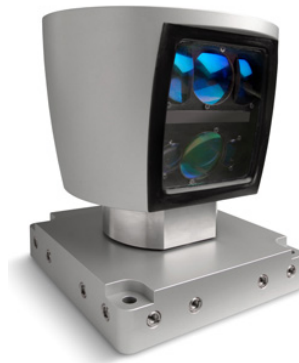


Abbildung 2.9 – Velodyne HDL-64E, Quelle: Velodyne.

2.4.3 Stereokamera

Eine Stereokamera ist ein passiver Umgebungssensor, bei dem zwei Kameras die gleiche Szene aus leicht unterschiedlichen Perspektiven wahrnehmen. Die zu den Bildebenen gehörenden Brennpunkte sind durch die Basis b voneinander getrennt. Werden zwei Linsen mit der Brennweite f genutzt, so gilt für den Abstand r eines Objektes

$$r = b \frac{f}{d}, \quad (2.65)$$

wobei d für die Disparität steht. Durch die Bestimmung der Disparitäten eines Objektes im Stereobild kann also dessen Abstand ermittelt werden. Wird der Abstand für jeden Punkt

Tabelle 2.2 – Spezifikation von Lidar-Sensoren im automobilen Kontext, Herstellerangaben

Hersteller	Hella	Hella	IBEO	IBEO	Velodyne
Bezeichnung	IDIS 1.0	IDIS Scanner	Alasca XT	Lux	HDL-64E
Laser-Dioden	16	1	4	4	64
Sichtbereich horiz.	16°	90–170°	160–240°	85–110°	360°
Auflösung horiz.	1°	1–3,1°	0,1–1°	0,1–1°	0,09–0,27°
Sichtbereich vert.	3°	0,5–3,2°	0–3,2°	3,2°	26,8°
Auflösung vert.	3°	0,5–3,2°	0–0,8°	3,2°	0,4°
Distanzgenauigkeit	±1%	0,1 m	0,04 m	0,1 m	0,02 m
Abtastfrequenz	16 Hz	20 Hz	12,5/25 Hz	12,5/25 Hz	5/10/15 Hz
Reichweite	150 m	150 m	200 m	110 m	120 m
Anzahl Ziele	1	2	4	3	1
Schnittstelle	CAN	CAN	TCP/IP,CAN	TCP/IP,CAN	UDP

des Bildes ermittelt, so ergibt sich ein Tiefenbild wie in Abbildung 2.10 oben. Die ermittelte Tiefe ist farblich kodiert.

Badino [4] beschreibt eine Abstraktion für ein solches Tiefenbild: Die Umgebungsmodellierung durch *Stixel*. Abbildung 2.10 aus Pfeiffer [99] zeigt neben dem dichten Tiefenbild (oben) die zugehörige Stixelrepräsentation (unten). In dieser Arbeit wird lediglich auf die abstrakte Stixel-Repräsentation zurückgegriffen.

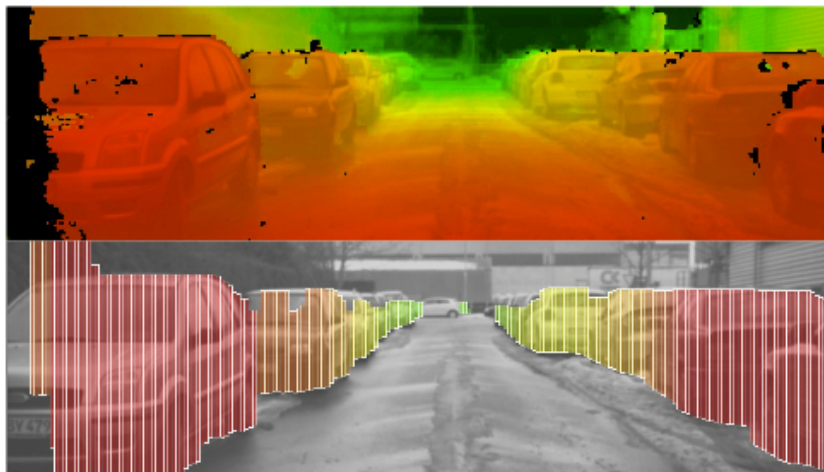


Abbildung 2.10 – Umgebungserfassung mit einer Stereokamera. Oben: Ein dichtes Tiefenbild. Unten: Die zugehörige abstrakte Stixel-Repräsentation. Quelle: Pfeiffer [99]

3

Umgebungsmodellierung durch hierarchische Belegungskarten

In diesem Kapitel wird die Modellierung der statischen Umgebung eines Fahrzeuges durch dreidimensionale hierarchische Belegungskarten beschrieben. Die Modellierung unterstützt dabei eine dynamische Anpassung des Detaillierungsgrades, so dass relevante Bereiche feiner aufgelöst werden können als andere. Dies dient der Erhöhung der Genauigkeit in anwendungsrelevanten Bereichen, um beispielsweise die Position von Landmarken möglichst exakt zu bestimmen oder den befahrbaren Freiraum möglichst präzise zu erfassen. Eine hierarchische Datenstruktur macht den Einsatz einer dreidimensionalen Karte in der Praxis erst möglich.

Eine Belegungskarte setzt einerseits voraus, dass lediglich statische Objekte integriert werden und andererseits, dass die Sensorlage relativ zur Karte bekannt ist. Die Unsicherheit in der Schätzung der Pose wird als normalverteilt modelliert. Diese zwei Annahmen sollen für dieses Kapitel als erfüllt betrachtet werden. In den beiden folgenden Kapiteln 4 und 5 werden Ansätze beschrieben, die Lösungen bieten, um diese Annahmen zu erfüllen.

3.1 Motivation

Die Umgebungserfassung für Fahrerassistenz- und Sicherheitssysteme basiert in der Regel auf kostengünstigen und ins Design integrierten Sensoren. Bei diesen Sensoren tritt zum Teil ein relativ hohes Messrauschen auf. Mit Belegungskarten lässt sich dieses teilweise kompensieren. In einer Karte, die den Raum in diskrete Zellen unterteilt, werden Sensormessungen über die Zeit akkumuliert. Damit kann Rauschen reduziert und die Genauigkeit erhöht werden. Ein weiterer Vorteil der Akkumulierung von Sensormessungen besteht in der virtuellen Erweiterung des Sichtfeldes, da Bereiche, die aus dem Sensorsichtfeld geraten, noch für eine gewisse Zeit in der Karte gehalten werden.

Mit zweidimensionalen Karten kann ein Großteil der komplexen Funktionen wie Lokalisierung, Freiraumanalyse und Detektion bewegter Objekte abgedeckt werden. Eine Modellierung der Umgebung in drei Dimensionen legt jedoch den Grundstein für zusätzliche Möglichkeiten: Die Betrachtung der Höhe eines Hindernisses macht die Entscheidung, ob dieses über- oder unterfahrbar ist, erst möglich. Eine Selbstlokalisierung in Steilkurven und auf Rampen ist generell nur in drei Dimensionen möglich. Des Weiteren können überhängende Strukturen wie Brücken korrekt abgebildet und deren Höhe geschätzt werden. Ein 3D-Laserscanner wie zum Beispiel der Velodyne HDL-64E (siehe Abschnitt 2.4.2), der ein sphärisches Distanzprofil liefert, entfaltet mit einer dreidimensionalen Karte erst sein volles Potenzial.

In dieser Arbeit wird deshalb der Einsatz einer dreidimensionalen Belegungskarte vorgeschlagen, um so viele Funktionen wie möglich mit einer Art der Umgebungsrepräsentation bedienen zu können.

Eine dreidimensionale Repräsentation der Umgebung benötigt naturgemäß mehr Speicherplatz und Rechenzeit als eine ebene Modellierung. Es bleibt zu zeigen, dass die Anforderungen an die Hardware sich reduzieren, indem hierarchische Datenstrukturen eingesetzt werden.

Kapitelübersicht

Abschnitt 3.2 gibt einen ausführlichen Überblick über den Stand der Technik der Kartierung mit Belegungskarten. In Abschnitt 3.3 werden die verwendeten Koordinatensysteme definiert. Anschließend wird in Abschnitt 3.4 ein generisches inverses Sensormodell beschrieben, mit dem Messungen der verwendeten aktiven Sensoren in die Belegungskarte integriert werden können. Abschnitt 3.5 führt schließlich die Datenstruktur ein, die eine dünn besetzte Belegungskarte realisiert, bei der im Betrieb die Auflösung einzelner Bereiche für die Anwendung angepasst werden kann. Die beschriebenen Ansätze gelten gleichermaßen für sowohl zwei- als auch dreidimensionale Belegungskarten. In Abschnitt 3.6 wird auf die Schätzung des Belegungszustandes basierend auf der Wahrscheinlichkeitstheorie und auf die Möglichkeit der

Modellierung mit Hilfe der Evidenztheorie nach Dempster und Shafer (DST) eingegangen. Abschnitt 3.7 beschreibt die notwendigen Operationen um die Datenstruktur zu modifizieren und Messungen zu integrieren. In Abschnitt 3.8 werden unterschiedliche Funktionen präsentiert, anhand derer der Detaillierungsgrad der Karte gesteuert werden kann. In Abschnitt 3.9 erfolgt eine Evaluierung der Effizienz der Datenstruktur und Algorithmen. Schließlich folgt in Abschnitt 3.10 eine Zusammenfassung und Möglichkeiten zur Erweiterung der vorgestellten Ansätze.

3.2 Stand der Technik

Zweidimensionale Karten

Im Jahr 1986 begründete Alberto Elfes Belegungskarten (engl. Occupancy Grids) für Anwendungen in der Robotik [35]. Seine Arbeiten basieren auf Ultraschallsensorik und haben die Kartierung und Navigation im Innenraum von Gebäuden zum Ziel. Da Messungen von Ultraschallsensoren mit hohen Unsicherheiten behaftet sind, wird eine probabilistische Darstellung gewählt. Eine solche Karte teilt einen mehrdimensionalen Raum in diskrete Zellen, deren Zustand (frei oder belegt) als Belegungswahrscheinlichkeit angegeben wird. Sensordaten werden in dieser Belegungskarte akkumuliert. Initial beträgt die Belegungswahrscheinlichkeit einer Zelle typischerweise 50% und steigt oder sinkt dann je nachdem, welche Sensormessungen in dieser Zelle anfallen. In Anlehnung an Landkarten verwendet Elfes eine flache Modellierung der Umgebung mit einer zweidimensionalen Karte. Dies ist zudem nützlich um mit den eingeschränkten Kapazitäten der damaligen Rechner auszukommen. Elfes schlägt 1986 bereits Karten mit unterschiedlich hohem, jedoch konstantem Detaillierungsgrad vor, um je nach Problemstellung die passende Auflösung zur Verfügung zu stellen. Der Begriff „Occupancy Grid“ wird durch Elfes 1989 eingeführt [34].

Konolige [73] beschreibt ein Sensormodell für Ultraschallsensoren, um in Belegungskarten besser mit Totalreflexionen der Sensoren umgehen zu können.

Thrun [128] führt die Verwendung eines Vorwärtssensormodells ein. Typischerweise wird bei Belegungskarten ein inverses Sensormodell genutzt: Von einer Sensormessung wird auf die wahrscheinlichste zugehörige Umgebung geschlossen, Abhängigkeiten zwischen benachbarten Sensormessungen werden dabei in der Regel ignoriert. Beim Vorwärtssensormodell wird diejenige Karte gesucht, die am wahrscheinlichsten zum gegebenen Satz von Sensormessungen geführt hat. Hierbei handelt es sich um ein aufwändiges Optimierungsproblem, bei dem theoretisch alle möglichen Karten betrachtet werden müssen. Im Gegensatz zum inversen Sensormodell werden dabei Abhängigkeiten zwischen benachbarten Zellen berücksichtigt und das Problem der Kartenerstellung im hochdimensionalen Raum gelöst.

In [130] gibt Thrun eine umfassende Einführung in Belegungskarten und nutzt diese um mit Laserscannern 2D-Karten zu erstellen sowie Navigations- und Lokalisierungsprobleme zu lösen.

C.-C. Wang [136] stellt einen Ansatz vor um die Selbstlokalisierung und die Detektion bewegter Objekte integriert zu lösen. Dazu nutzt er eine 2D-Belegungskarte für innerstädtische Szenerien.

M. Wang [137] beschreibt einen Ansatz zur Segmentierung und Merkmalsextraktion aus Laserscannerdaten. Die Landschaftsdaten werden vom Flugzeug aus aufgenommen und in einer Octree-Datenstruktur verarbeitet.

Linzmeier [80] erstellt aus den Messdaten mehrerer Thermopiles eine 2D-Karte, um Fußgänger zu detektieren.

Weiss beschreibt zweidimensionale Belegungskarten für Anwendungen im Automobilbereich [140]. Als Sensor kommt ein Alasca XT (Abschnitt 2.4.2) Laserscanner zum Einsatz. Die Positions- und Orientierungsunsicherheit des Fahrzeuges wird mit der Messunsicherheit des Laserscanners überlagert. Als Anwendungen behandelt er die Detektion bewegter Objekte, die Detektion des Fahrbahnrandes und Navigation mit Punktlandmarken. Damit keine unnötigen Diskretisierungsartefakte entstehen wird das Fahrzeug über die Karte bewegt. Bevor das Fahrzeug den Rand der Karte erreicht, wird diese im Speicher dynamisch verschoben.

Hundelshausen et al. [132] beschreiben eine auf einem 3D-Laserscanner basierende fahrzeuglokale Belegungskarte, die in jedem Zeitschritt neu erstellt wird. Diese wird benutzt, um eine Freiraumanalyse für autonome Fahrfunktionen durchzuführen. Dieser Ansatz wird mit „Fahren mit Tentakeln“ bezeichnet.

Richter et al. [102] motivieren die Notwendigkeit der Bestimmung von Objektgeschwindigkeiten aus Belegungskarten über eine Funktion zur Kollisionsvermeidung. Dazu nutzen sie eine 2D-Karte und berechnen Geschwindigkeitsvektoren für jede Zelle. Um den Berechnungsaufwand zu reduzieren, stellen sie einen Ansatz namens Pyramiden-Gitter vor. In Anlehnung zur Beschreibung von Elfes [35] werden mehrere Karten in unterschiedlichen Auflösungen betrachtet. Zuerst werden Messungen in ein grobes Gitter mit konstanter Auflösung integriert. Belegte Gitterzellen werden anschließend weiter betrachtet: Die zugehörige Messung wird erneut in ein weiteres Gitter mit höherer Auflösung integriert. Nicht belegte Zellen werden von der groben Karte in die Karte mit höherer Auflösung kopiert.

Alternativ zur klassischen Wahrscheinlichkeitstheorie beschreiben unter anderem Mullane [89], Effertz [30] und Konrad [74] die DST als Grundlage für die Schätzung des Belegungszustandes einer Zelle. Durch die Evidenztheorie lässt sich neben dem Belegungszustand auch eine Art Unsicherheit über den Belegungszustand schätzen.

Darms stellt die Sensorfusion eines Fernbereichsradars mit Monokamera in einer Belegungskarte vor [21]. Dabei kommen sowohl UND- als auch ODER-Karten zum Einsatz. Als Anwendung für die Belegungskarte wird eine Fahrbahnschätzung vorgestellt, die unabhängig von Spurmarkierungen arbeitet.

Konrad [77] beschreibt eine Belegungskarte zur Fahrbahnverlaufsschätzung. Als Sensoren kommen Radar- und Lidarsensoren sowie eine Monokamera zum Einsatz. Bilder der Monokamera werden über die Methode „Inverse Perspective Mapping“ (IPM) in die Belegungskarte integriert. Für IPM wird eine flache Bodenebene angenommen. Die Grauwerte im Kamerabild werden über eine Koordinatentransformation auf die Bodenebene im dreidimensionalen Raum abgebildet. Somit kann eine Zuordnung zu Zellen in der Belegungskarte erfolgen. Zur Fahrbahnverlaufsschätzung wird eine digitale Karte mit der Belegungskarte registriert, um eine höhere Vorausschau zu erreichen.

Zweidimensionale Karten mit Höheninformationen

Evans et al. [41] stellen eine 2D-Karte mit Höheninformationen für den Innenraum vor. Die Höheninformationen werden durch einen einzeiligen Laserscanner gewonnen, der um 12° nach unten geneigt ist.

Montemerlo und Thrun [85] beschreiben eine $2\frac{1}{2}$ D-Karte mit Höheninformation. Dabei gehen sie davon aus, dass bei einem Distanzsensor wie bei einem Laserscanner mit steigendem Abstand auch die Messunsicherheiten steigen. Sie benutzen eine Reihe von Belegungskarten, die je nach Abstand einer Messung aktualisiert werden, damit durch die geringe Auflösung in großen Distanzen keine Hindernisse übersehen werden.

Kleiner [71] verwendet eine Belegungskarte für Rettungsroboter. Er schätzt die Höhe von Objekten in einem 2D-Gitter, in dem neben der Belegungswahrscheinlichkeit Höhenwerte abgelegt werden. Er geht dabei von einer Welt ohne dynamische Objekte aus.

Himmelsbach et al. [60] benutzen eine nicht-akkumulierende Belegungskarte, um die dreidimensionalen Punktwolken eines Velodyne Laserscanners zu segmentieren. Für die Segmentierung werden die absoluten Höhendifferenzen in den Zellen der Karte genutzt. Weitere Verarbeitungsschritte wie eine Objektklassifikation beruhen wieder auf den zugehörigen 3D-Punktwolken.

Badino [4] repräsentiert Verkehrssituationen mit Höheninformationen durch den Stixel-Ansatz. Die Stixel werden aus den Daten einer Stereokamera abgeleitet. Ein Stixel beschreibt einen auf einer hypothetischen Fahrbahnebene senkrecht stehenden Bereich durch einen Abstandswert zur Kamera, einen Azimut-Winkel und eine Höhe. Die Stixelwelt stellt in dieser Form eine kompakte Umgebungsrepräsentation dar, die jedoch dieselben Nachteile hat wie

andere $2\frac{1}{2}$ D-Ansätze: Brücken und andere überhängende Strukturen können nicht korrekt dargestellt werden und sorgen dafür, dass der befahrbare Freiraum möglicherweise unnötig eingeschränkt wird. Pfeiffer [98] erweitert den Ansatz, indem er Stixel in mehreren Ebenen betrachtet, um diese Probleme zu beseitigen.

Dreidimensionale Karten

Erstmals in der Literatur treten bei Meagher 1982 Octrees zur Modellierung dreidimensionaler geometrischer Objekte auf [83].

Moravec [88] erweitert die Idee der Belegungskarte auf drei Dimensionen im Jahre 1996. Er beschreibt eine dichte dreidimensionale Belegungskarte konstanter Auflösung für den Innenraum. Als Sensor benutzt er eine Stereokamera.

Payeur et al. [96] stellen 1997 die Modellierung der Umgebung durch Belegungskarten in 3D durch Octrees vor. Dabei wird für eine Sensormessung zunächst eine polare Karte angelegt, welche anschließend in die Belegungskarte integriert wird. Der hierarchische Aufbau des Octrees wird genutzt: Lediglich in Knoten, in denen sich eine Objektgrenze befindet, wird die Rekursionstiefe erhöht und die Belegungswahrscheinlichkeit detailliert übertragen. Die Detaillierung der Karte in belegten Zellen ist jedoch konstant. Der Einsatz der Karte dient der Überprüfung von Stromleitungen durch einen Spezialroboter. Das Verfahren wird auf simulierten Daten einer Stromleitung evaluiert. Die Berechnung einer Karte mit 3 Metern Größe benötigt bei einer maximalen Auflösung von 0,05 m auf den damaligen Rechnern noch bis zu 15 Minuten.

Thrun et al. [129] erstellen aus zwei 2D-Laserscannern eine dreidimensionale Karte. Die Karte wird durch Polygone repräsentiert, die aus den Laserscannerpunkten erstellt werden.

Huber et al. [62] erstellen mit unterschiedlichen Distanzsensoren eine dreidimensionale Karte, die ebenfalls durch Polygonnetze repräsentiert wird.

Foessel-Bunting et. al [43, 44] stellen ein dreidimensionales Sensormodell für ein frequenzmoduliertes Dauerstrichradar (FMCW) vor und erstellen damit eine dreidimensionale Belegungskarte fester Größe. Das Radar ist durch das notwendige wiederholte Aufnehmen der Situation in 15 Elevationswinkeln nicht in der Automobilindustrie anwendbar.

Teizer [123] benutzt eine 3D-Belegungskarte um Baustellen zu modellieren. Er benutzt eine dreidimensionale Belegungskarte mit konstanter Auflösung. Messunsicherheiten der Sensorik werden nicht betrachtet, das bedeutet, dass jede Messung jeweils genau eine Gitterzelle beeinflusst.

Scheunert et al. [105] beschreiben ein Grid Framework zur Akkumulierung von Sensordaten. Dazu werden polare oder kartesische Belegungskarten verwendet. Die Daten eines mehrzeiligen Laserscanners und eines PMD-Sensors werden in einer Karte mit konstanter Auflösung akkumuliert. Es werden drei Modi zur Kartierung beschrieben: Eine globale Kartierung mit einer ortsfesten Karte, eine fahrzeuglokale Kartierung und eine Kartierung mit objektlokalen Karten.

Pathak [95] beschreibt ein dreidimensionales Vorwärts-Sensormodell und wendet es auf eine 3D-Karte an. Als Sensoren kommen ein PMD-Sensor und eine Stereokamera zum Einsatz, die in der Karte fusioniert werden.

Wurm et al. [146] stellen die „OctoMap“ vor, eine Belegungskarte mit Octrees. In der „OctoMap“ werden Datensätze unterschiedlicher Laserscanner integriert. Das Hauptaugenmerk der Arbeit liegt auf einem geringen Speicherbedarf der Karte. So wird zur Komprimierung der Karte der Belegungszustand durch eine Schwellwertoperation in den ternären Zustandsraum { frei, belegt, unbekannt } überführt und der entstehende Bitdatenstrom komprimiert. Die Auflösung der Karte ist konstant zu beispielsweise 0,2 Meter gewählt, wobei jedoch Abfragen auf gröberen Stufen möglich sind. Ebenso werden homogene Bereiche automatisch zusammengefasst. Die Veröffentlichung enthält eine gute Übersicht über bisherige Arbeiten zur Umgebungsmodellierung mit Octrees. Die Implementierung der „OctoMap“ ist frei verfügbar.

Einhorn et al. [31, 32] führen einen N^d -Baum für Belegungskarten ein. Ein Quadtree ist dabei der Spezialfall eines 2^2 -Baumes und ein Octree der eines 2^3 -Baumes. Die Karte wird vergrößert, indem neue Wurzelknoten hinzugefügt werden. Der Detaillierungsgrad wird direkt durch anfallende Messungen gesteuert: Treten in einer Zelle Messungen auf, die zum vorherigen Belegungszustand inkonsistent sind, wird die Hierarchietiefe erhöht bis zu einer maximalen vorgegebenen Auflösung von beispielsweise 0,05 Metern Kantenlänge. Dazu wird für jede Zelle ein Histogramm über die angefallenen Messungen geführt.

Dryanovski et al. [28] beschreiben „Multi-Volume Occupancy Grids“ als effiziente Alternative zur Modellierung dreidimensionaler Belegungskarten. Hierbei handelt es sich um ein 2D-Gitter, bei dem jede Zelle zwei Listen enthält: Jeweils eine Liste mit belegten und eine mit freien Quadern, die sich senkrecht entlang der z-Achse über der Zelle befinden. Als Sensor kommt ein PMD-Sensor mit einer Reichweite von bis zu 8 Metern im Innenraum zum Einsatz.

Bouzouraa und Hofmann [10, 11] benutzen einen Laserscanner zum Aufbau einer Belegungskarte für Fahrerassistenzsysteme. Dynamische Objekte werden über Inkonsistenzen zwischen der bestehenden Belegungskarte und den Messungen des Laserscanners detektiert. Der Zustand dynamischer Objekte wird geschätzt. Bei der Prädiktion eines Objektes werden die zugehörigen Zellen aus der Belegungskarte an neue Positionen verschoben. Dabei wird eine Subzellularberechnung eingesetzt, um entstehende Diskretisierungsartefakte zu minimieren.

Übersicht

Tabelle 3.1 gibt abschließend eine Übersicht über den Stand der Technik zu Belegungskarten. Zusammenfassend lässt sich sagen, dass Belegungskarten intensiv in der Robotik zum Einsatz kommen, sowohl in zwei als auch in drei Dimensionen. Der Einsatz von zweidimensionalen Belegungskarten im automobilen Kontext für Fahrerassistenzsysteme ist in den letzten Jahren ebenfalls sehr beliebt geworden.

Dreidimensionale Belegungskarten für den Einsatz in Fahrerassistenzsystemen sind jedoch noch relativ unbekannt. Im Rahmen dieser Arbeit wurde hierzu 2010 ein Paper veröffentlicht [153]. Eine weitere Arbeit, ebenfalls aus 2010, stammt von Bouzouraa [10]. Einige Arbeiten, vor allem im Robotik-Bereich, beschreiben Karten unterschiedlicher Auflösung für unterschiedliche Anwendungsbereiche. Es existiert jedoch noch kein Verfahren, bei dem die Auflösung der Karte dynamisch und von der Anwendung gesteuert für unterschiedliche Bereiche der Belegungskarte angepasst werden kann. Bestehende Verfahren sind demnach dazu in der Lage, die Bereiche der Karte fein aufzulösen, in denen Messungen gehäuft auftreten. Es ist jedoch nicht möglich, anwendungsrelevante Bereiche definieren, in denen die Auflösung angepasst wird. Somit ist es nicht möglich, und somit eine genauere Kartierung erzielt wird.

Tabelle 3.1 – Übersicht zum Stand der Technik zu Belegungskarten

Anwendungen	2D	2,5D	3D
Robotik			
Kartierung	[35]	[41, 85]	[88, 146, 31]
Navigation	[35, 130]	[71]	[96]
Fahrerassistenzsysteme			
Kartierung	[140]		[44]
Spurschätzung	[21, 140, 77, 109]		
Selbstlokalisierung	[136, 140]		
Detektion bewegter Objekte	[136, 141, 87]		[11]
Fußgängerdetektion	[80]		
Autonomes Fahren	[132, 30, 125]	[60]	
PreCrash	[102]		
Freiraumanalyse		[4]	

3.3 Koordinatensysteme

Abbildung 3.1 zeigt eine Übersicht über die in der Arbeit verwendeten Koordinatensysteme. Diese werden im Folgenden näher erläutert.

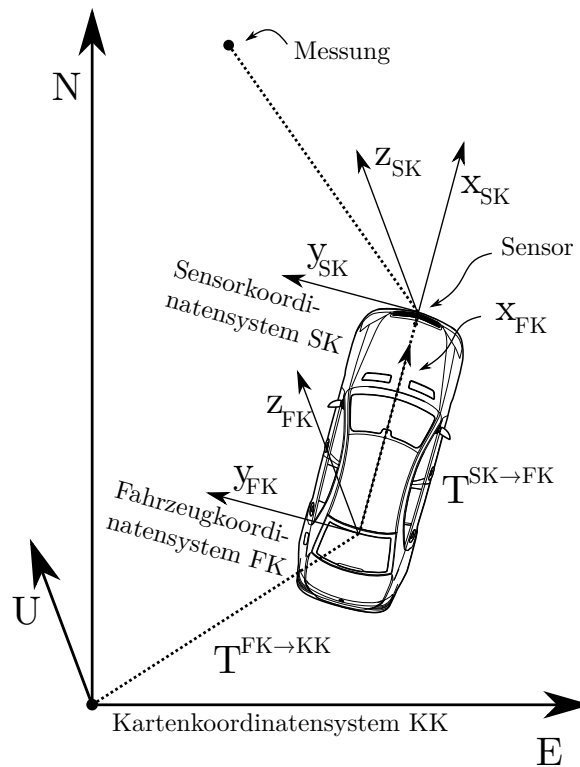


Abbildung 3.1 – Die in der Arbeit verwendeten Koordinatensysteme: Globales Kartenkoordinatensystem KK, fahrzeuglokales Koordinatensystem FK und Sensorkoordinatensystem SK.

3.3.1 Kartenkoordinatensystem

Es gibt zwei Möglichkeiten das Kartenkoordinatensystem zu wählen: Entweder man wählt das Koordinatensystem ortsfest und das Sensorfahrzeug bewegt sich über die Karte, oder man wählt ein fahrzeuglokales Koordinatensystem, das sich mit dem Fahrzeug durch die Welt bewegt.

Weiss [140] beschreibt das Problem, das entsteht wenn die Kartenkoordinaten fest zum Sensorfahrzeug gewählt werden: Bei jeder Bewegung des Fahrzeuges müssen bereits diskretisierte Belegungswerte neu diskretisiert werden, was zu einem erhöhten Rechenaufwand und einer zunehmenden Unschärfe in einzelnen Gitterzellen führt. Abbildung 3.2 verdeutlicht dieses Problem.

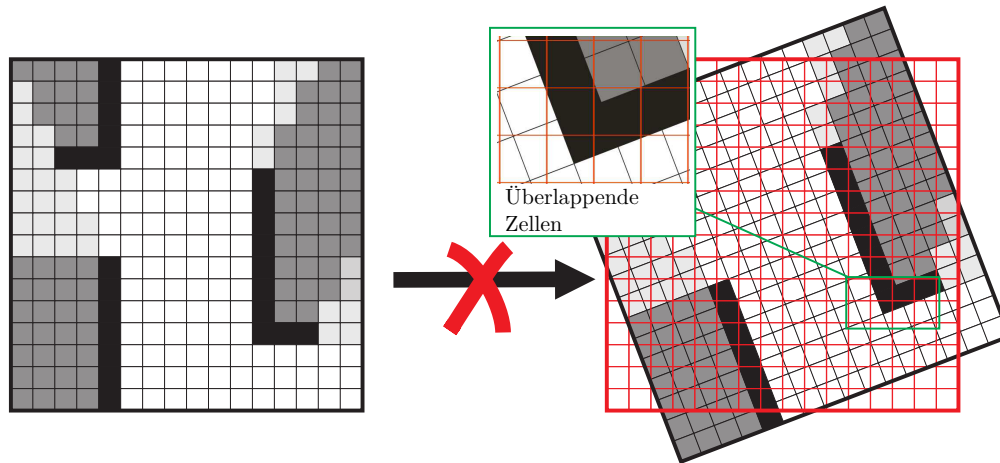


Abbildung 3.2 – Eine Neudiskretisierung in jedem Zeitschritt macht eine fahrzeuglokale Karte unpraktikabel. Quelle: Weiss [140].

Bei einem ortsfesten Kartenkoordinatensystem KK sind alle Gitterzellen inhärent parallel zu den Koordinatenachsen ausgerichtet. Die Pose des Fahrzeuges relativ zur Karte wird mitgeführt. Abbildung 3.3 zeigt zwei Möglichkeiten, ein kartesisches ortsfestes Koordinatensystem zu definieren: Das „Earth Centered Earth Fixed“ (ECEF) Koordinatensystem und das „East-North-Up“ (ENU) Koordinatensystem. Beim ECEF liegt der Ursprung im Erdmittelpunkt, die x -Achse geht durch den Schnittpunkt des Äquators mit dem Nullmeridian und die z -Achse geht durch den Nordpol. Beim ENU-Koordinatensystem liegt der Ursprung hingegen in einem definierten Punkt auf der Erdoberfläche. Die E-N-Ebene stellt eine lokale Tangentialebene dar. Die U-Achse zeigt im Ursprung stets senkrecht von der Erde weg nach außen. Die geographische Länge wird hier mit λ und die geographische Breite mit φ bezeichnet.

Aus praktischen Gründen wird in dieser Arbeit das ENU-System gewählt. Der Ursprung des Kartenkoordinatensystems wird auf den Startpunkt der Messfahrt festgelegt.

3.3.2 Fahrzeugkoordinatensystem

Das Fahrzeugkoordinatensystem FK wird wie in [12] definiert festgelegt. Die x -Achse entspricht der Fahrzeuglängsachse, in Fahrtrichtung positiv; die y -Achse der Fahrzeugquerachse, positiv nach links und die z -Achse der Fahrzeughochachse, positiv nach oben. Der Koordinatenursprung liegt im Mittelpunkt der Hinterachse.

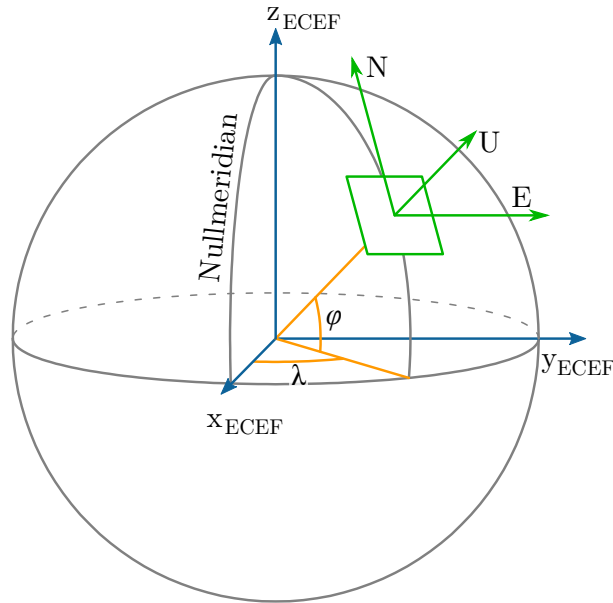


Abbildung 3.3 – Bezug zwischen ECEF-Koordinatensystem und ENU-Koordinatensystem. Das ENU-Koordinatensystem stellt eine lokale Tangentialebene dar.

3.3.3 Sensorkoordinatensysteme

Der Ursprung des Sensorkoordinatensystems SK wird in den Ursprung des jeweiligen Sensors gelegt. Die Messungen eines Sensors, welche ein Distanzprofil in der Ebene darstellen, liegen in der Regel in Polarkoordinaten vor. Dabei wird der Abstand mit r und der Azimutwinkel mit φ bezeichnet. Bei Sensoren, die ein dreidimensionales Distanzprofil messen, kommt der Elevationswinkel θ hinzu. Um die Sensormessungen einheitlich darstellen zu können wird für Sensoren, die keine Elevation messen, der Elevationswinkel auf 0° festgelegt. Damit liegt eine Sensormessung generell in sphärischen Koordinaten vor:

$$\mathbf{y} = \begin{bmatrix} r & \varphi & \theta \end{bmatrix}^T \quad (3.1)$$

Um die Position einer Sensormessung in das kartesische Kartenkoordinatensystem transformieren zu können, wird diese zunächst in kartesische Sensorkoordinaten transformiert. Dazu wird folgende Transformationsvorschrift verwendet:

$$\begin{aligned} \mathbf{p}^{\text{SK}} &= \begin{bmatrix} x & y & z \end{bmatrix}^T \\ &= \begin{bmatrix} r \cos \theta \cos \varphi & r \cos \theta \sin \varphi & r \sin \theta \end{bmatrix}^T \end{aligned} \quad (3.2)$$

3.3.4 Koordinatentransformationen

Die Position des Fahrzeuges bezüglich der Karte wird in kartesischen Koordinaten mit dem Translationsvektor $[x_F \ y_F \ z_F]^T$ angegeben. Die Lage des Fahrzeuges im Raum wird mit den drei Eulerwinkeln Rollwinkel (Φ), Nickwinkel (Θ) und Gierwinkel (Ψ) beschrieben. Es gibt also sechs Freiheitsgrade, die in dem Vektor \mathbf{x}_F zusammengefasst werden:

$$\mathbf{x}_F = \begin{bmatrix} x_F & y_F & z_F & \Phi & \Theta & \Psi \end{bmatrix}^T \quad (3.3)$$

Eine aus diesen Komponenten aufgestellte homogene 4x4-Transformationsmatrix dient der Transformation von FK in KK und wird mit $\mathbf{T}^{\text{FK} \rightarrow \text{KK}}$ bezeichnet:

$$\mathbf{T}^{\text{FK} \rightarrow \text{KK}} = \begin{pmatrix} & x \\ R_{xyz} & y \\ & z \\ 0 & 1 \end{pmatrix} \quad (3.4)$$

Wobei R_{xyz} eine 3x3-Rotationsmatrix ist, welche aus den Drehmatrizen um die einzelnen Koordinatenachsen zusammengesetzt ist:

$$R_{xyz} = R_x(\Phi) R_y(\Theta) R_z(\Psi) \quad (3.5)$$

$\mathbf{T}^{\text{FK} \rightarrow \text{KK}}$ wird in diesem Kapitel als bekannt vorausgesetzt. Näheres zur Bestimmung der Lage und Orientierung des Fahrzeuges findet sich in Kapitel 4. Die Sensoranbauposition und Orientierung wird mit $\mathbf{T}^{\text{SK} \rightarrow \text{FK}}$ angegeben und wird aus der Anbauposition und -orientierung des jeweiligen Sensors bestimmt.

Die Position einer Sensormessung relativ zum Fahrzeug \mathbf{p}^{FK} erhält man dann mit (3.6), die Position über der Karte \mathbf{p}^{KK} mit (3.7):

$$\mathbf{p}^{\text{FK}} = \mathbf{T}^{\text{SK} \rightarrow \text{FK}} \cdot \mathbf{p}^{\text{SK}} \quad (3.6)$$

$$\mathbf{p}^{\text{KK}} = \mathbf{T}^{\text{FK} \rightarrow \text{KK}} \cdot \mathbf{T}^{\text{SK} \rightarrow \text{FK}} \cdot \mathbf{p}^{\text{SK}} \quad (3.7)$$

Bei Laserscannern tritt durch die abtastende Arbeitsweise ein Effekt auf, der zu einer Verzerrung in den Messdaten führt: Durch eine Bewegung des Sensorfahrzeuges erscheinen Messpunkte teilweise an der falschen Position. Die erforderliche Korrektur für \mathbf{p}^{KK} wird in Ab-

schnitt 3.4.3 beschrieben.

3.4 Inverses Sensormodell

Ein Sensormodell beschreibt, welchen Messwert ein Sensor in einer realen Umgebung liefert. Das inverse Sensormodell hingegen lässt, basierend auf dem Messwert eines Sensors, Rückschlüsse darauf ziehen, wie die reale Welt dazu ausgesehen haben muss. Um nicht für jeden Sensor ein spezielles inverses Sensormodell angeben zu müssen, wird ein einheitliches generisches Modell eingeführt. Das inverse Sensormodell soll zur Integration von sowohl Lidar- als auch Radarmessungen geeignet sein.

3.4.1 Messrauschen

Jede Sensormessung ist mit einer Messunsicherheit behaftet. Wenn man davon ausgeht, dass die Messunsicherheit durch die Überlagerung einer großen Anzahl von voneinander unabhängigen Einflüssen entsteht, kann mit dem zentralen Grenzwertsatz angenommen werden, dass die Messunsicherheit normalverteilt ist. In der Realität stellt die multivariate Normalverteilung (2.5) eine gute Näherung für die Messunsicherheiten von aktiven Sensoren dar.

Ein Vorteil der Normalverteilung ist, dass diese einfach parametrisch beschrieben werden kann: Durch eine Kovarianzmatrix \mathbf{C} und einen Mittelwert μ . Für jeden Sensor, der ein Distanzprofil in Kugelkoordinaten liefert, kann eine Kovarianzmatrix \mathbf{C}_m basierend auf der Sensorspezifikation aufgestellt werden. Diese nähert im Sinne der Normalverteilung das Messrauschen an. Diese Kovarianzmatrizen werden in den folgenden Abschnitten für unterschiedliche Sensoren eingeführt.

Lidar-Sensoren

Bei einem Lidar-Sensor kann das Messrauschen durch eine Diagonalmatrix \mathbf{C}_m dargestellt werden. Die Elemente auf der Hauptdiagonalen werden durch die Genauigkeit der Distanzmessung σ_r , die horizontale Strahlaufweitung α_h und die vertikale Strahlaufweitung α_v bestimmt.

$$\mathbf{C}_m = \text{diag} \left(\sigma_r^2, (\alpha_h \cdot 6^{-1})^2, (\alpha_v \cdot 6^{-1})^2 \right) \quad (3.8)$$

Der Faktor 6^{-1} resultiert aus der Annahme, dass durch den Strahl 99,7% aller Messwerte entstehen und dieser damit dem $\pm 3\sigma$ -Intervall entspricht.

Laserscanner mit Spiegeloptik

Für Laserscanner wird für die Beschreibung der Strahlaufweitung dieselbe Matrix wie für Lidar-Sensoren genutzt (3.8). Die Strahlkeulen bestimmter Laserscanner neigen sich jedoch, bedingt durch das Messprinzip, nach außen hin. Unter anderem ist dies beim Ibeo Alasca XT und beim Hella IDIS Scanner der Fall [9, 51], siehe Abschnitt 2.4.2. Aus diesem Grund werden die Strahlkeulen auch im Modell in der Längsachse um den Azimutwinkel φ der Messung rotiert, der dem negativen Neigungswinkel der Strahlkeulen entspricht. Dafür wird die Rotationsmatrix R_x genutzt. Für den Hella IDIS Scanner ist diese gegeben durch:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-\varphi) & \sin(-\varphi) \\ 0 & -\sin(-\varphi) & \cos(-\varphi) \end{bmatrix} \quad (3.9)$$

Der Velodyne Laserscanner (siehe Abschnitt 2.4.2) hat diesen Effekt nicht, so dass hier $R_x = \mathbf{I}$ gesetzt wird.

Für Sensoren mit Spiegeloptik gilt also:

$$\mathbf{C}_m = R_x \cdot \text{diag} \left(\sigma^2, (\alpha_h \cdot 6^{-1})^2, (\alpha_v \cdot 6^{-1})^2 \right) \cdot R_x^T \quad (3.10)$$

Besonderheiten des Hella IDIS mit variabler Auflösung

Der Hella IDIS Scanner mit einem Öffnungswinkel von 161° besitzt unterschiedliche Auflösungen je nach Azimutwinkel. In den Randbereichen wird die Auflösung reduziert: Dies dient der Reduktion der Auslastung des Datenübertragungskanal. Im Sensor intern wird dies folgendermaßen umgesetzt: Es wird zunächst mit konstanter Auflösung gemessen, anschließend werden aber mehrere Messungen zusammengefasst. Im Intervall $\varphi \in [0^\circ, 20,5^\circ]$ wird mit 1° Auflösung gemessen; im Intervall $\varphi \in [20,5^\circ, 60,5^\circ]$ wird mit 2° gemessen, damit werden jeweils $N = 2$ Einzelmessungen zusammengefasst. Im Bereich $\varphi \in [60,5^\circ, 80,5^\circ]$ wird mit 4° Auflösung gemessen, damit wird jeweils über $N = 4$ Einzelmessungen gemittelt. Um die Messunsicherheiten korrekt zu modellieren, werden zunächst virtuelle Punkte \mathbf{p}_i mit deren Messunsicherheiten \mathbf{C}_{m_i} in konstantem Abstand von 1° berechnet. Die Messpunkte werden anschließend über (3.11) zu $\bar{\mathbf{p}}$ gemittelt und die Messunsicherheiten über (3.12) zu $\bar{\mathbf{C}}_m$ zusammengefasst. Abbildung 3.4 zeigt die modellierten Messunsicherheiten für eine reale Messung.

$$\bar{\mathbf{p}} = \frac{1}{N} \sum_i^N \mathbf{p}_i \quad (3.11)$$

$$\bar{\mathbf{C}}_m = \sum_i^N \left[\mathbf{C}_{m_i} + (\mathbf{p}_i - \bar{\mathbf{p}})(\mathbf{p}_i - \bar{\mathbf{p}})^t \right] \quad (3.12)$$

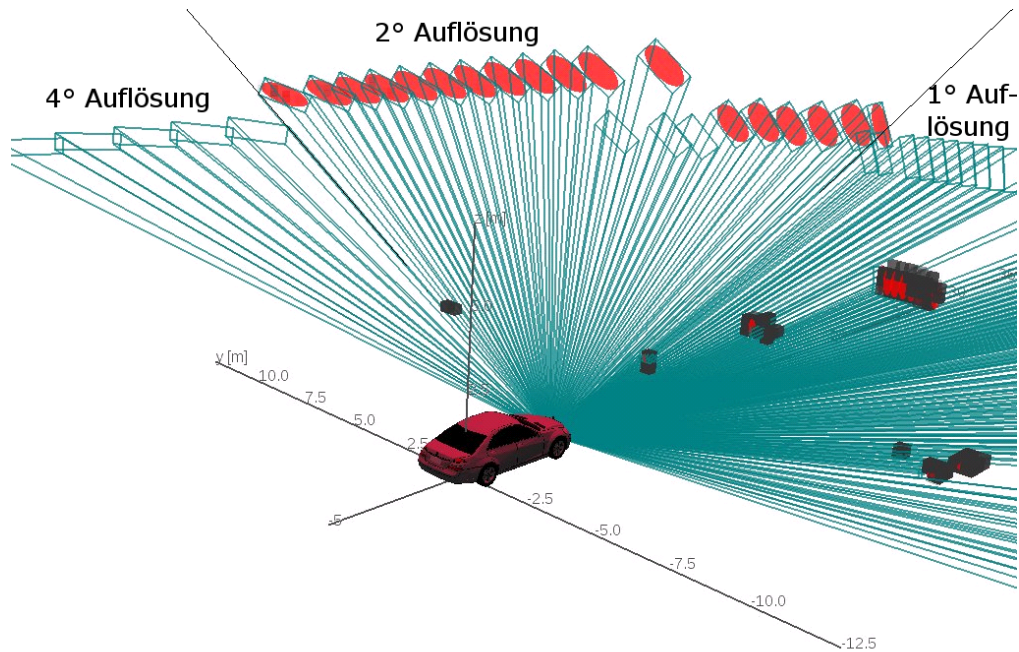


Abbildung 3.4 – Reale Messung mit modellierten Messunsicherheiten des Hella IDIS Laserscanners mit variabler Auflösung.

Radarsensoren

Radarsensoren liefern in der Regel sowohl nicht gefilterte Rohdatenziele als auch gefilterte Ziele (Objekte). Für die Integration in die Karte werden die ungefilterten Rohdatenziele gewählt. Unter Vernachlässigung des Mehrwegeempfangs und der Nebenkeulen kann deren Messunsicherheit \mathbf{C}_m wie bei Lidar-Sensoren mit einer Kovarianzmatrix dargestellt werden:

$$\mathbf{C}_m = \text{diag}(\sigma_r^2, \sigma_\varphi^2, \sigma_\theta^2) \quad (3.13)$$

Transformation in kartesische Koordinaten

Die Transformation einer Kovarianzmatrix von Kugelkoordinaten in das kartesische Koordinatensystem kann über eine Jacobimatrix \mathbf{J} approximiert werden. \mathbf{J} ist gegeben durch:

$$\mathbf{J} = \begin{bmatrix} \cos \theta \cos \varphi & -r \cos \theta \sin \varphi & -r \sin \theta \cos \varphi \\ \cos \theta \sin \varphi & r \cos \theta \cos \varphi & -r \sin \theta \sin \varphi \\ \sin \theta & 0 & r \cos \theta \end{bmatrix} \quad (3.14)$$

Die kartesische Kovarianzmatrix \mathbf{C}_{mc} kann dann approximiert werden durch:

$$\mathbf{C}_{mc} = \mathbf{J} \mathbf{C}_m \mathbf{J}^T \quad (3.15)$$

Diese Transformation verursacht einen Fehler, der mit steigender Winkelunsicherheit wächst. Für große Winkelunsicherheiten sollte demnach die „Unbiased Transformation“ verwendet werden, hierfür sei auf Longbin [81] verwiesen.

\mathbf{C}_{mc} kann durch den assoziierten $\pm 3\sigma$ -Ellipsoiden dargestellt werden. Dieser wird exemplarisch für einen Datensatz eines Hella IDIS Laserscanners in Abbildung 3.5 (oben) gezeigt. Unten sind die realen Messkeulen des Sensors abgebildet. Im direkten Vergleich sieht man, dass die Modellierung durch die Normalverteilung eine gute Annäherung bietet.

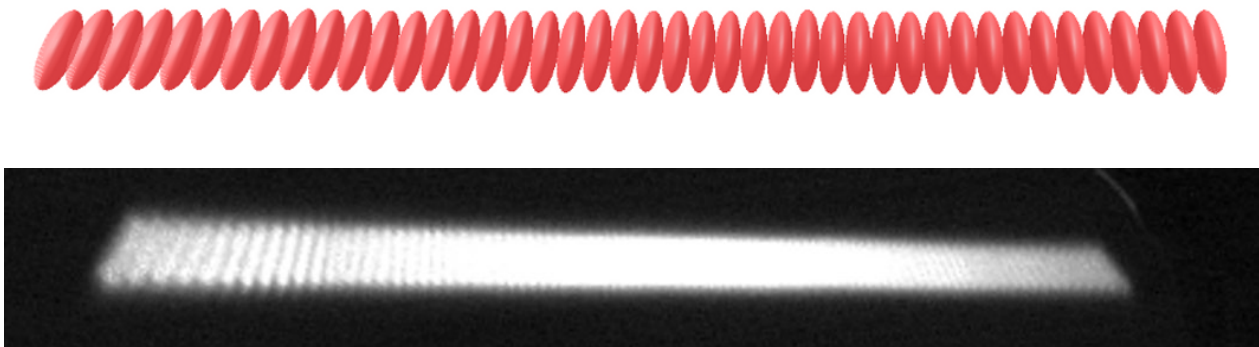


Abbildung 3.5 – Messunsicherheiten des Hella IDIS Laserscanners. Oben: Modellierung durch eine dreidimensionale Normalverteilung und Visualisierung der $\pm 3\sigma$ -Ellipsoide. Unten: Aufnahme der realen Strahlkeulen, die auf eine Wand treffen. Durch die perspektivische Sicht der Kamera werden die Strahlkeulen nach rechts schmaler.

Zum Vergleich unterschiedlicher Sensoren und deren Messunsicherheiten wird exemplarisch eine Momentaufnahme aus einer Sequenz betrachtet, die in einem städtischen Szenario aufgezeichnet wurde. Abbildung 3.6a zeigt das Bild der Dokumentationskamera. Rechts am Straßenrand befinden sich parkende Fahrzeuge, das Sensorfahrzeug fährt einem PKW hinterher.

Abbildung 3.6b zeigt die zugehörigen Messungen der Nahbereichsradare, wie in Abschnitt 2.4.1 beschrieben. Links ist die Situation mit den Sensormessungen in der Draufsicht darge-

stellt, rechts in einer perspektivischen Darstellung aus der Sicht des Fahrers. Auf dem vorausfahrenden Fahrzeug liegt eine Messung, ebenso liegen zwei Messungen direkt hintereinander auf dem vordersten der parkenden Fahrzeuge.

In Abbildung 3.6c wird die Messung des Ibeo LUX Laserscanners, wie in Abschnitt 2.4.2 beschrieben, für dieselbe Momentaufnahme gezeigt. Wie bei den Radarsensoren ist die Situation links in der Draufsicht dargestellt, rechts in einer perspektivischen Darstellung. Das vorausfahrende Fahrzeug bildet sich deutlich durch einige Messungen ab, in der unteren Ebene sind sogar die Räder des Fahrzeuges zu erkennen. Die Kontur der parkenden Fahrzeuge wird abgebildet, ebenso der Bordstein auf der linken und auf der rechten Seite.

Abbildung 3.6d zeigt eine stixelbasierte Messung der Stereokamera, wie in Abschnitt 2.4.3 beschrieben. Zum Vergleich ist links wiederum die Situation in der Draufsicht, rechts in der perspektivischen Ansicht dargestellt. Das vorausfahrende Fahrzeug bildet sich hier ebenfalls durch mehrere Messungen ab. Die Kontur der parkenden Fahrzeuge ist auch deutlich zu erkennen. Rechts neben dem vorderen parkenden Fahrzeug wird der Baum mit deutlichem Höhenunterschied zu den Fahrzeugen gemessen.

3.4.2 Gemessener Freiraum

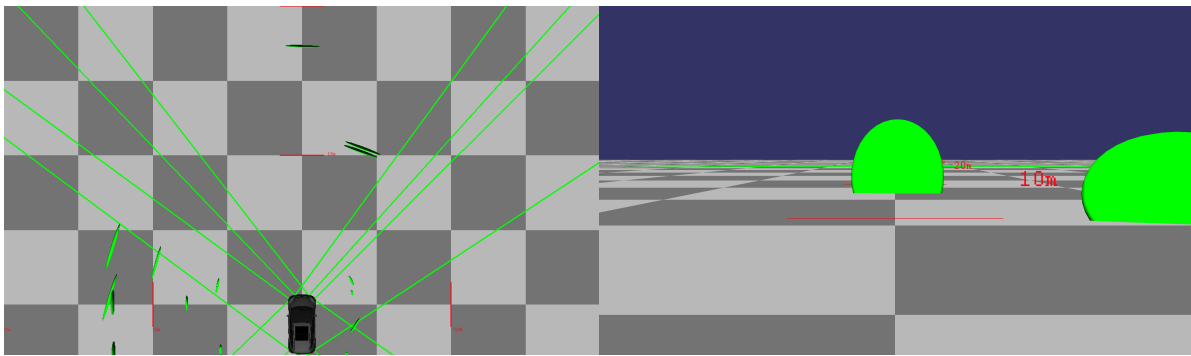
Lidar-Sensoren liefern neben dem gemessenen Abstandswert aufgrund des stark gebündelten Laserstrahls implizit die Information, dass der Weg von der Laserdiode zum gemessenen Objekt frei ist. Dieser Freiraum wird in dieser Arbeit im dreidimensionalen über eine Pyramide approximiert, um geometrische Berechnungen einfach zu gestalten. Die Pyramide wird über fünf Ebenen beschrieben, vier Ebenen die den Strahl längs begrenzen und eine Ebene, die das Ende des Strahles markiert. Die Ebenen werden durch einen Normalen- und einen Stützvektor repräsentiert. Der Normalenvektor zeigt stets in das Innere des Strahles. Mit dem Kreuzprodukt kann für einen Punkt effizient bestimmt werden, ob dieser innerhalb oder außerhalb der Pyramide liegt.

Um die Ellipsenform des Strahles anzunähern können die vier Ebenen rhombenförmig oder rechteckig angeordnet werden. Abbildung 3.7 zeigt die zwei unterschiedlichen Möglichkeiten.

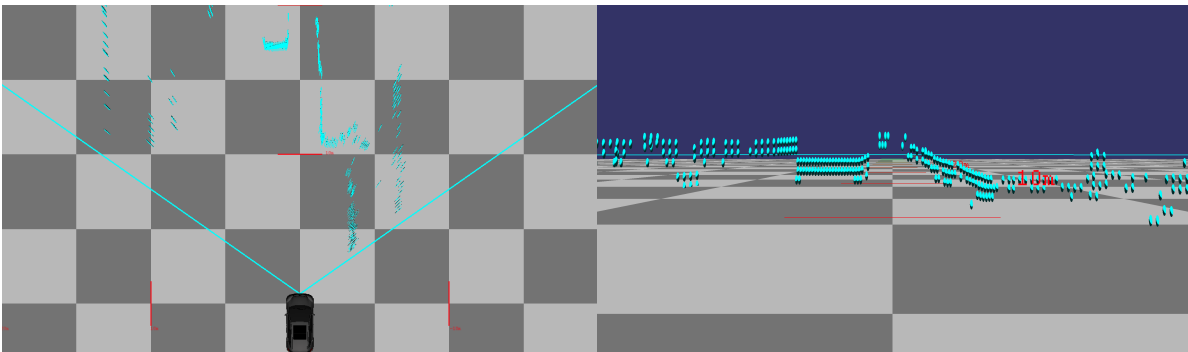
Abbildung 3.8 zeigt ein Beispiel der modellierten pyramidenförmigen Freiräume anhand realer Messdaten. In Abbildung 3.9 ist das zugehörige Bild der Dokumentationskamera zu sehen.



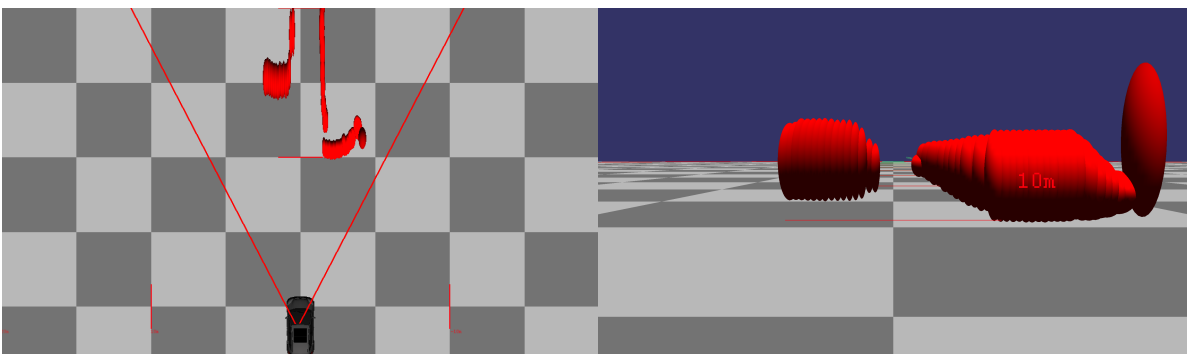
(a) Bild der Dokumentationskamera einer Testfahrt in Böblingen.



(b) Messungen von sechs Nahbereichsradaren. In Grün sind die Sichtbereiche dargestellt.



(c) Messungen des Ibeo LUX Laserscanners. Der Sichtbereich ist in Cyan dargestellt.



(d) Messungen einer Stereokamera, der Sichtbereich ist in Rot visualisiert.

Abbildung 3.6 – Es sind jeweils Messungen zu der in (a) gezeigten Sequenz dargestellt. Links in der Draufsicht mit den Sichtbereichen der Sensoren, rechts aus der Perspektive des Fahrers. Die roten Linien markieren jeweils 10 m-Schritte im Abstand zum Ursprung.

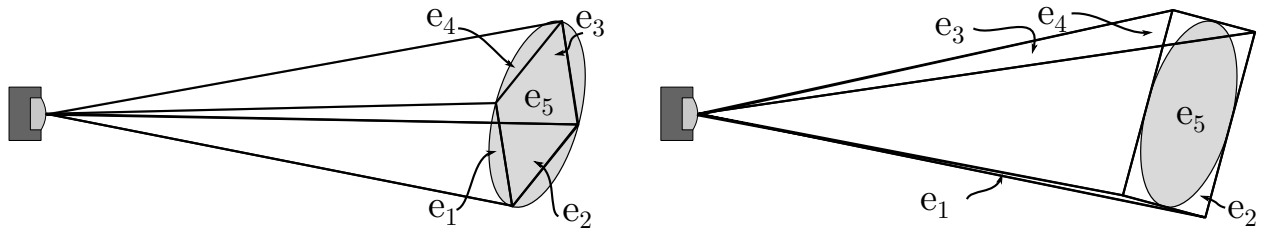


Abbildung 3.7 – Modellierung eines Laserstrahles als Pyramide. $e_1 - e_5$ bezeichnen jeweils die fünf Ebenen, die zur Begrenzung des Freiraums dienen. Es gibt zwei unterschiedliche Methoden. Links: Approximierung durch einen Rhombus. Rechts: Approximierung durch ein Rechteck.

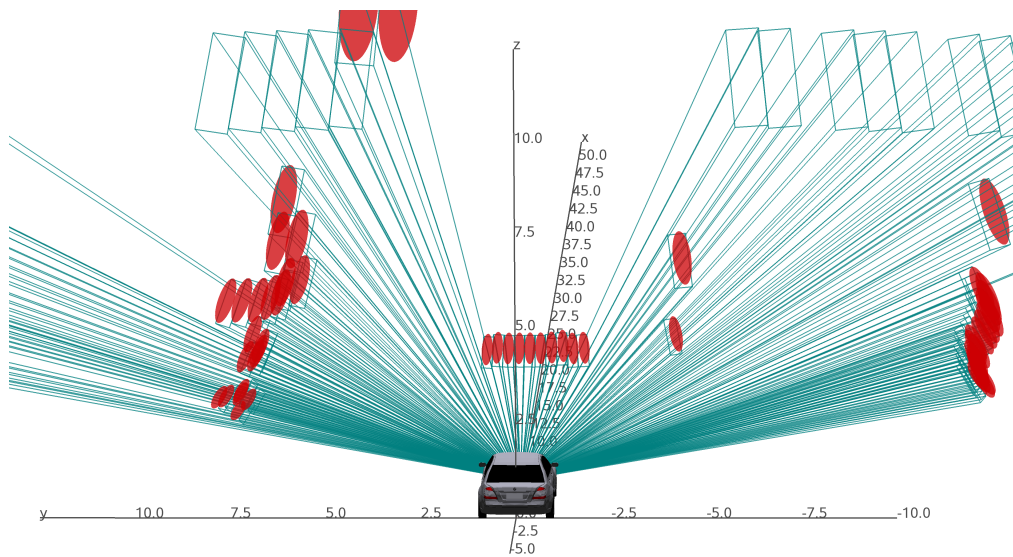


Abbildung 3.8 – Modellierung der gemessenen Freiräume durch Pyramiden (cyan) mit dem Hella IDIS Laserscanner. Ein vorausfahrendes Fahrzeug bildet sich im Distanzprofil des Laserscanners ab. Man erkennt deutlich die Modellierung der Messunsicherheiten über Ellipsoide.



Abbildung 3.9 – Referenzbild der Dokumentationskamera zu der in Abbildung 3.8 gezeigten Modellierung.

3.4.3 Motion-Scan-Effekt

Motivation

Laserscanner tasten die Umgebung des Fahrzeuges schrittweise ab. Daraus resultiert, dass Messpunkte aus benachbarten Winkelschritten nicht zur gleichen Zeit aufgenommen werden können, sondern einer nach dem anderen. Das Distanzprofil, das der Laserscanner auf der Datenschnittstelle ausgibt, spiegelt diese schrittweise Aufnahme jedoch nicht wider. Bei einem stillstehenden Fahrzeug und einer statischen Szene wirkt sich dies nicht aus. Bei einem bewegten Fahrzeug oder einer Bewegung in der Umgebung treten allerdings Fehler auf.

Kapp [70] liefert eine genaue Beschreibung des Motion-Scan-Effekts und zeigt die Auswirkungen durch eine Simulation. Der Fehler, der durch die Relativbewegung zwischen dem eigenen Fahrzeug und einem angemessenen Fahrzeug auftritt, wird durch ein Bewegungsmodell kompensiert. Zusätzlich müssen Annahmen zur Form und Bewegungsrichtung der Verkehrsteilnehmer getroffen werden. In einer Simulation kann dies berücksichtigt werden.

Für den Aufbau einer Belegungskarte, die hauptsächlich statische Elemente der Umgebung widerspiegeln soll, genügt eine Kompensation der Eigenbewegung. Auf eine Kompensation des Fehlers von anderen Verkehrsteilnehmern, deren Bewegungsrichtung nicht bekannt ist, und die sich aus dem Distanzprofil des Sensors nicht direkt ermitteln lässt, wird verzichtet.

Beispiel 3.1 *Ein hypothetischer Laserscanner ist mit 180° Sichtbereich nach vorne auf einem Fahrzeug befestigt, rotiert mit 20 Hz und tastet in 1° -Schritten die Umgebung ab. Für das Distanzprofil, das er auf der Datenschnittstelle ausgibt, wirkt sich eine Geschwindigkeit von 180 km/h (=50 m/s) bei der Geradeausfahrt wie folgt aus:*

Eine volle Rotation erfolgt alle 50 ms. Der Sichtbereich wird demnach in 25 ms überstrichen. Daraus ergibt sich zwischen zwei benachbarten Winkelschritten eine Zeitdifferenz von 0,139 ms. Diese wirkt sich bei 50 m/s Eigengeschwindigkeit als Fehler von 0,7 cm in x-Richtung aus. Über den kompletten Sichtbereich von 180° summiert sich der Fehler zwischen den äußersten Winkelbereichen auf 1,25 m auf.

Inertialkorrektur

Durch die Inertialkorrektur werden Sensormessungen korrigiert, die durch die Inertialbewegung des Fahrzeuges verfälscht wurden. Dafür spielt die Einbaulage des Sensors im Fahrzeug eine wesentliche Rolle. Ein vom Sensor gemessener Punkt \mathbf{p}^{SK} wird deshalb zunächst ins Fahrzeugkoordinatensystem transformiert, siehe Gleichung (3.6).

Durch das Bewegungsmodell des Fahrzeuges, das in Abschnitt 4.3 eingeführt wird, kann der Fehler verringert werden. Die Änderung der Position des Fahrzeuges im Raum wird dabei von der Funktion $s(v, \dot{\Theta}, \dot{\Psi}, \Delta t)$ beschrieben, die von der Geschwindigkeit v , der Nickrate $\dot{\Theta}$, der Gierrate $\dot{\Psi}$ und der Zeitdifferenz Δt abhängt, wie in Gleichung (4.7) definiert. Dabei beschreibt Δt die Zeitspanne, die zwischen dem realen Aufnahmezeitpunkt eines einzelnen Messpunktes und dem Zeitstempel t_0 des Distanzprofils liegt.

Ein Messpunkt in Fahrzeugkoordinaten wird korrigiert durch:

$$\mathbf{p}_{\text{corr}}^{\text{FK}} = \mathbf{p}^{\text{FK}} - s(v, \dot{\Theta}, \dot{\Psi}, \Delta t) \quad (3.16)$$

$$\mathbf{p}_{\text{corr}}^{\text{KK}} = \mathbf{T}^{\text{FK} \rightarrow \text{KK}} \cdot \mathbf{p}_{\text{corr}}^{\text{FK}} \quad (3.17)$$

Durch (3.17) wird der Scanpunkt relativ zum Kartenkoordinatensystem berechnet.

3.5 Datenstruktur

Für zweidimensionale Belegungskarten wird häufig ein Datenfeld (engl. Array) als Datenstruktur verwendet. Bei Datenfeldern ist ein schneller und einfacher Zugriff möglich. Bei dreidimensionalen Belegungskarten ist ein Datenfeld, das in diesem Fall ein Voxelgitter darstellt, jedoch ungeeignet, da der Speicherbedarf kubisch mit der Kartengröße wächst.

Beispiel 3.2 *Bei einer gewünschten Auflösung von 0,05 m und einer Größe der Karte von 100 m · 100 m · 25 m ergeben sich 2 Milliarden Zellen. Dies entspricht bei 16 Bit Belegungswahrscheinlichkeit einem Speicheraufwand von 3,73 GB.*

3.5.1 Dünn besetzte Voxelgitter

Ein Voxelgitter ist ein Raster zur räumlichen Unterteilung in drei Dimensionen. Eine einzelne Zelle des Gitters wird Voxel genannt. Werden alle Einträge gespeichert, so nennt man das Gitter dicht besetzt, andernfalls dünn besetzt.

Ein dünn besetztes Voxelgitter wird in dieser Arbeit durch ein assoziatives Datenfeld realisiert. Zur Indizierung wird als Schlüssel das Tripel (i, j, k) verwendet, wobei i den Index auf der x-Achse, j den Index auf der y-Achse und k den Index auf der z-Achse angibt. Die Implementierung wurde aus der C++-Standard Bibliothek gewählt. In Abbildung 3.10 wird ein exemplarisches Voxelgitter dargestellt, an dem die Indizierung dargestellt wird.

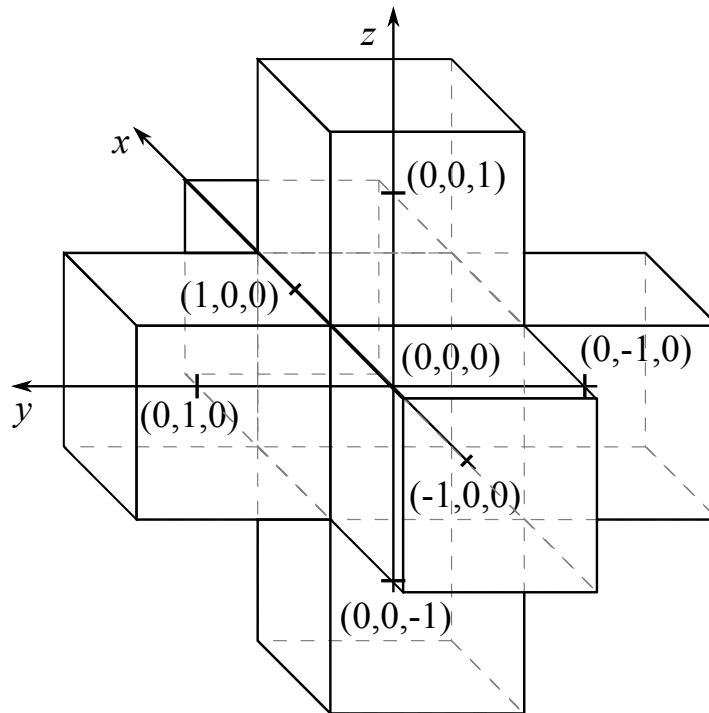


Abbildung 3.10 – Überblick über ein dünn besetztes Voxelgitter. Dargestellt sind beispielhaft sieben Zellen, die in einem globalen Koordinatensystem liegen. Indiziert wird eine Zelle über ein Tripel (i,j,k) . Der Ursprung des Koordinatensystems entspricht dem Zentrum der Zelle an Position $(0,0,0)$. Diese Vorgehensweise ist unabhängig von der konkreten Seitenlänge einer Zelle, diese muss allerdings bekannt sein um den Zellen Messungen zuzuordnen zu können.

3.5.2 Hierarchische Datenstrukturen

Meist ist eine hohe Auflösung nur in funktionsrelevanten Bereichen notwendig, beispielsweise an Objekträndern oder bei Landmarken. In homogenen Bereichen genügt ein deutlich geringerer Detaillierungsgrad. Eine hierarchische Datenstruktur erlaubt das Zusammenfassen homogener Bereiche zu größeren Blöcken. Zudem ist neben der Erhöhung der Kartenauflösung in inhomogenen Bereichen auch das explizite Erhöhen der Auflösung für relevante Bereiche und die explizite Verringerung für nicht relevante Bereiche möglich. Auf den dynamischen Detaillierungsgrad wird weiter in Abschnitt 3.8 eingegangen.

Folgende hierarchische Datenstrukturen zur räumlichen Unterteilung in drei Dimensionen sind aus der Literatur bekannt [45]:

BSP-Baum

Bei der binären Raumpartitionierung BSP (engl. Binary Space Partitioning) [48, 49] werden multidimensionale Daten rekursiv durch Hyperebenen unterteilt. Die Hyperebenen sind frei

wählbar, im Besonderen müssen sie nicht senkrecht aufeinander stehen. Dadurch ergeben sich Unterräume mit unterschiedlichsten Formen. Dies macht die Anwendung von BSP-Bäumen auf Belegungskarten in der Praxis kompliziert.

k-d-Baum

k-d-Bäume wurden von Bentley [5] eingeführt. Ein k-d-Baum ist ein Spezialfall eines BSP-Baumes, bei dem die Unterteilungsebenen stets achsenparallel sind. Dabei werden die einzelnen Raumdimensionen nacheinander betrachtet. Bei einem k-d-Baum mit drei Raumdimensionen wird der Raum zunächst mit einer Ebene senkrecht zur x-Achse unterteilt. Beide Teilgebiete werden nun jeweils mit einer senkrechten Ebene zur y-Achse unterteilt, anschließend alle vier Teilgebiete jeweils mit einer senkrechten Ebene zur z-Achse. Daraus resultieren acht Gebiete. Genügt diese Unterteilung nicht, wird der Raum weiter nacheinander mit senkrechten Ebenen zur x-, y- und z-Achse unterteilt. Da die Koordinatenachsen separat betrachtet werden, entstehen im dreidimensionalen in der Regel unterschiedlich große Quader. Abbildung 3.11 zeigt einen würfelförmigen dreidimensionalen k-d-Baum mit jeweils einer Unterteilung pro Dimension. Die unterschiedlich großen Quader würden die Berechnung der Belegungswahrscheinlichkeiten äußerst aufwändig gestalten.

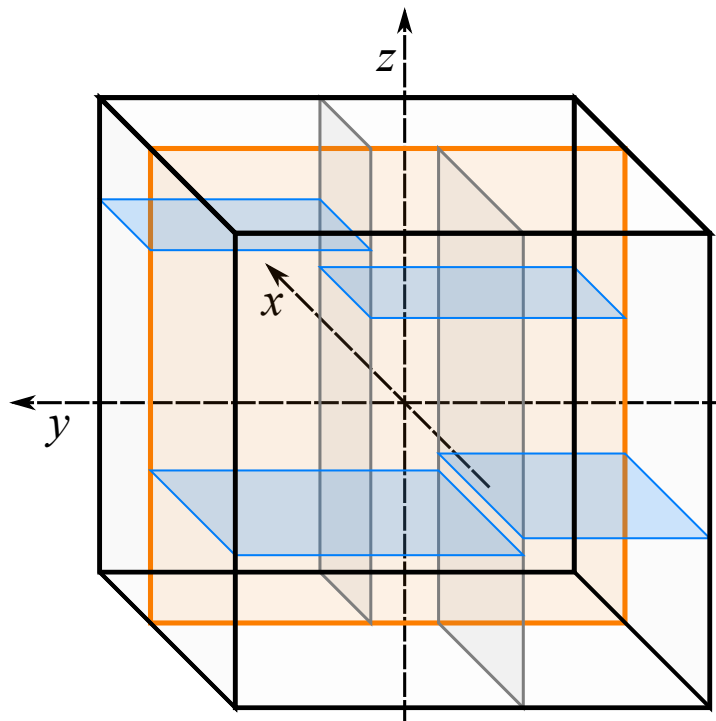


Abbildung 3.11 – k-d-Baum mit drei Dimensionen und jeweils einer Unterteilung pro Koordinatenachse. Zunächst wird senkrecht zur x-Achse aufgeteilt (Magenta), dann senkrecht zur y-Achse (Grün) und schließlich senkrecht zur z-Achse (Blau).

N^d -Baum

Einhorn et al. [31] beschreiben einen N^d -Baum. Abbildung 3.12 zeigt einen N^d -Baum mit $N = 3$ und $d = 2$. Ein d -dimensionaler Wurzelknoten mit gleich langen Seiten wird dabei rekursiv in jeweils N^d gleich große Kindknoten unterteilt.

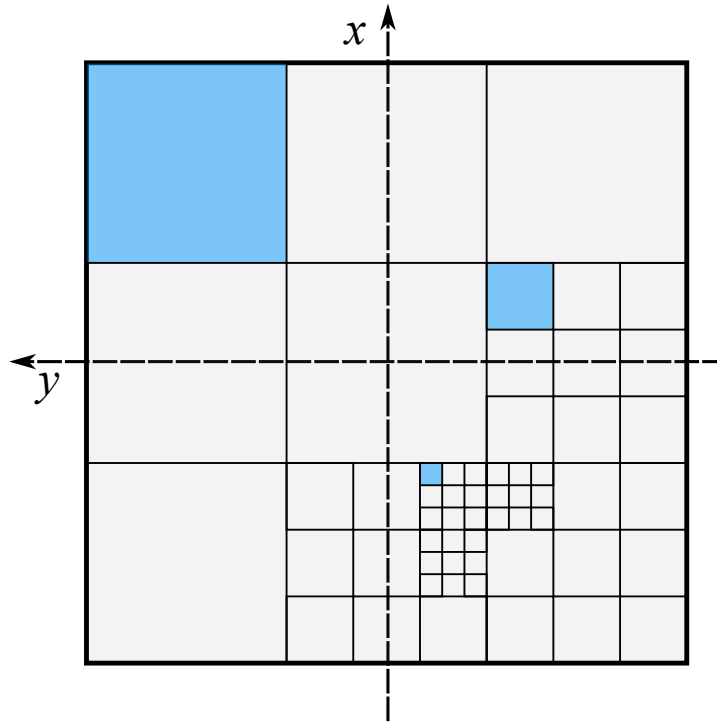


Abbildung 3.12 – 3^2 -Baum aus [31]. In diesem Beispiel wird ein zweidimensionales Quadrat rekursiv in jeweils $3^2 = 9$ Quadrate unterteilt.

Octree

Ein Octree ist der Spezialfall eines N^d -Baumes mit $N = 2$ und $d = 3$. Ein würfelförmiger Wurzelknoten wird so lange rekursiv in acht Kindknoten mit halber Seitenlänge des Elternknotens unterteilt, bis die gewünschte Auflösung erreicht ist. Alle Knoten auf einer Hierarchiestufe sind demnach würfelförmig und haben dieselbe Größe. Der Octree und viele Operationen auf Octrees, wie zum Beispiel der effiziente Zugriff auf einzelne Knoten, Mengenoperationen auf ganzen Bäumen oder Raycasting durch die Datenstruktur sind aus der Computergrafik bekannt [45].

Für eine dreidimensionale räumliche Aufteilung in Belegungskarten bietet sich der Octree als Datenstruktur an, da die Knoten jeweils würfelförmige Bereiche abdecken. Die Vorteile eines Octree gegenüber einem N^3 -Baum mit $N > 2$ ist die höhere Speichereffizienz und die Performanzvorteile bei großen Detaillierungsunterschieden in der Karte.

In Abbildung 3.13 ist exemplarisch ein Octree dargestellt. Die Kindknoten auf der ersten Ebene sind gestrichelt eingezeichnet. Die Indizierung der Kindknoten ist über die in der Abbildung angegebenen Ziffern definiert. Einen Knoten, der selbst keine Kinder hat, nennt man Blattknoten. Das Octree-lokale Koordinatensystem hat seinen Ursprung im Zentrum des Wurzelknotens. Die Seitenlänge des Octree-Wurzelknotens sei mit l bezeichnet. Die Seitenlänge eines Kindknotens ist damit $\frac{l}{2}$ und die Positionen der Teilbäume sind dann entsprechend:

$$\begin{aligned}
 1 &: \left(\frac{+l}{4}, \frac{+l}{4}, \frac{+l}{4} \right) & 5 &: \left(\frac{+l}{4}, \frac{+l}{4}, \frac{-l}{4} \right) \\
 2 &: \left(\frac{+l}{4}, \frac{-l}{4}, \frac{+l}{4} \right) & 6 &: \left(\frac{+l}{4}, \frac{-l}{4}, \frac{-l}{4} \right) \\
 3 &: \left(\frac{-l}{4}, \frac{+l}{4}, \frac{+l}{4} \right) & 7 &: \left(\frac{-l}{4}, \frac{+l}{4}, \frac{-l}{4} \right) \\
 4 &: \left(\frac{-l}{4}, \frac{-l}{4}, \frac{+l}{4} \right) & 8 &: \left(\frac{-l}{4}, \frac{-l}{4}, \frac{-l}{4} \right)
 \end{aligned} \tag{3.18}$$

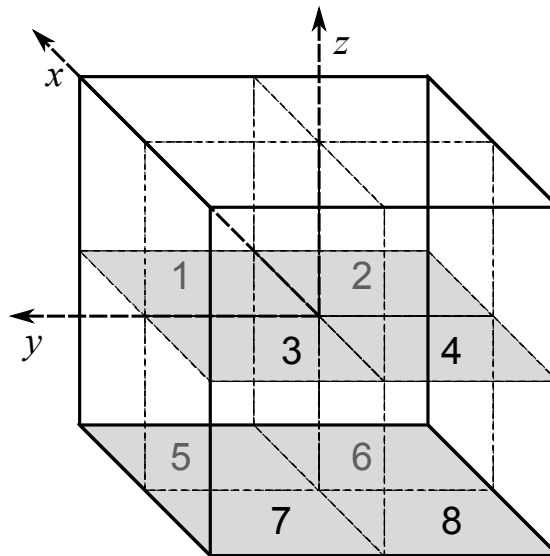


Abbildung 3.13 – Indizierung der Kindknoten eines Octreeknotens. Das octree-lokale Koordinatensystem hat seinen Ursprung im Zentrum des Wurzelknotens.

Um eine Zelle in einer Belegungskarte zu repräsentieren muss ein Octree-Knoten lediglich die Information über den Belegungswert enthalten und Zeiger auf sowohl den Vaterknoten als auch die acht Kindknoten, sofern vorhanden. Es sind keine Informationen über die Position des Knotens abgelegt; diese ergeben sich implizit bei der Traversierung des Baumes. Bei der Implementierung wird auf die speichereffiziente Methode aus [146] zurückgegriffen, bei der Zeiger auf Kindknoten erst angelegt werden, wenn diese benötigt werden. Alternativ zu

der mit Zeigern realisierten Variante lassen sich Octrees auch über eine Streuwert-Funktion implementieren, siehe Choi [14].

Analog zur dreidimensionalen Raumunterteilung durch Octrees lässt sich der zweidimensionale Raum hierarchisch durch Quadrees aufteilen. Hierdurch können im Vergleich zu einem zweidimensionalen Datenfeld bei gleichem Speicheraufwand punktuell höhere Auflösungen erreicht werden.

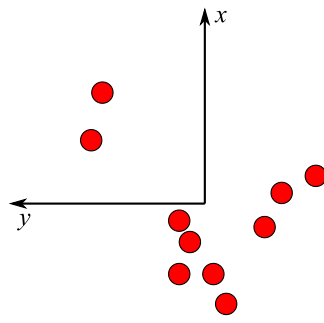
3.5.3 Kombination von Voxelgittern und hierarchischen Datenstrukturen

Der Wurzelknoten einer hierarchischen Baumstruktur, wie der eines Octrees, hat eine feste Kantenlänge, die sich im Nachhinein nicht mehr ändern lässt ohne die Datenstruktur neu zu organisieren. Da das Kartenkoordinatensystem ortsfest ist und sich das Fahrzeug darüber fortbewegt, würde das Fahrzeug früher oder später an die Grenzen einer durch einen einzelnen Baum repräsentierten Karte geraten. Ein dünn besetztes Voxelgitter hat diesen Nachteil nicht, es bietet allerdings auch nicht den Vorteil, dass die Auflösung in einem durch eine Applikation definierten Bereich variiert werden kann.

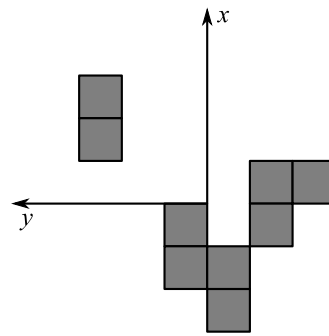
Um eine Belegungskarte zu erhalten, die sowohl im Laufe der Beobachtungszeit wachsen kann als auch eine Adaption der Auflösung ermöglicht, werden Octree-Wurzelknoten gleicher Größe als Zellen eines dünn besetzten Voxelgitter eingesetzt. Die Beschreibung der in dieser Arbeit verwendeten Datenstruktur für Fahrerassistenzsysteme wurde vorab in [153] veröffentlicht.

Für die zweidimensionale Variante der Implementierung werden Quadrees eingesetzt, analog zu den Octrees im dreidimensionalen Fall. Da der zweidimensionale Fall eine Vereinfachung des dreidimensionalen Falls darstellt, wird fortan nur noch der dreidimensionale Fall beschrieben.

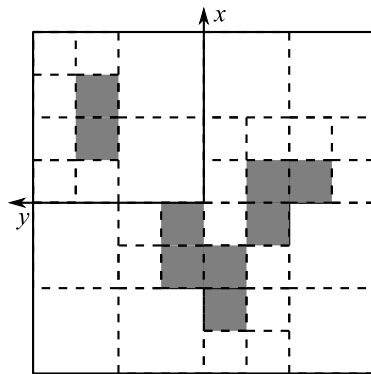
Die vorgeschlagene Datenstruktur lässt sich flexibel konfigurieren. In den Extremfällen lässt sich die Umgebung entweder als ein Octree mit großer Kantenlänge und großer Hierarchietiefe darstellen oder als Voxelgitter, das lediglich Wurzelknoten ohne Kindknoten enthält. Abbildung 3.14 verdeutlicht diese Varianten für einen exemplarischen Messdatensatz, der integriert wird.



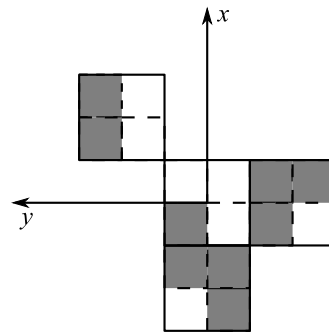
(a) Exemplarischer Messdatensatz im Kartenkoordinatensystem, der durch die Datenstruktur dargestellt werden soll.



(b) Datenstruktur als dünn besetztes Voxelgitter. Die enthaltenen Octrees bestehen lediglich aus Wurzelknoten, die Hierarchietiefe ist 0. Die Kantenlänge der Octrees entspricht der Zellgröße. Belegte Zellen sind grau markiert.



(c) Datenstruktur als einzelner Octree. Die Kantenlänge ist achtmal so groß wie in (b). Die Hierarchietiefe ist 3.



(d) Kombination aus (b) und (c). Die Kantenlänge ist doppelt so groß wie in (b), die Hierarchietiefe ist 1.

Abbildung 3.14 – Demonstration der Wandlungsfähigkeit der Datenstruktur. Zur Vereinfachung wird lediglich die Draufsicht dargestellt.

3.5.4 Zugriff auf die Datenstruktur

Zugriff auf eine einzelne Zelle

Für einen Zugriff auf eine Zelle der Belegungskarte wird zunächst der zugehörige Octree O_j im Voxelgitter \mathcal{V} ermittelt. Die Funktion idx bildet einen Punkt $\mathbf{p} \in \mathbb{R}^3$ auf den Index j des zugehörigen Octree O_j ab:

$$\text{idx} : \mathbf{p} \rightarrow j, \quad \mathbf{p} \in \mathbb{R}^3, j \in \mathbb{N} \quad (3.19)$$

In dem ermittelten Octree wird anschließend die betroffene Zelle ermittelt. Dies erfolgt mittels „Lokalisierungscodes“, wie von Frisken und Perry vorgestellt [47]. Lokalisierungscodes ermöglichen eine effiziente Vorausberechnung der Indizes der Kindknoten bis zum gewünschten Blattknoten. Dazu werden die x-, y- und z-Komponenten von \mathbf{p} jeweils separat mit $2^{c_{\max}}$ multipliziert. Die Konstante c_{\max} entspricht der maximalen Hierarchietiefe des Octrees. Die drei ermittelten Werte werden nun wie folgt interpretiert: Zunächst werden diese ins Dualsystem konvertiert. Anschließend wird für jede Stelle aus den drei Dualzahlen eine Zahl im Oktalsystem zusammengesetzt. Diese entspricht exakt dem Index des Kindknoten, in dem der gesuchte Punkt \mathbf{p} liegt, wie in Abbildung 3.13 festgelegt.

Abbildung 3.15 zeigt beispielhaft Lokalisierungscodes für ein eindimensionales Beispiel eines Binärbaums. Frisken und Perry beschreiben in ihrer Veröffentlichung eine zweidimensionale Implementierung für Quadrees, die für diese Arbeit auf drei Dimensionen erweitert wurde.

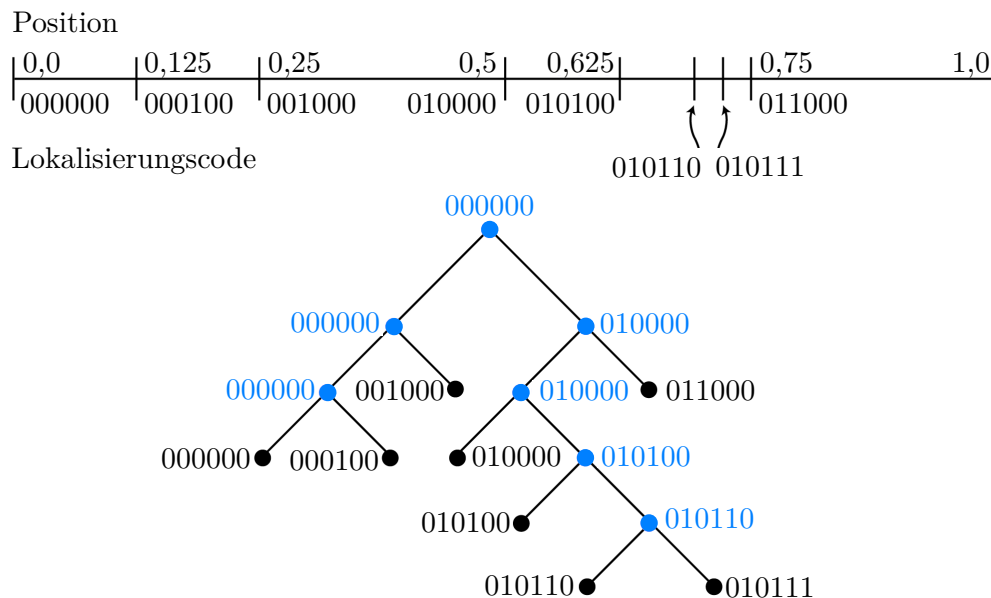


Abbildung 3.15 – Eindimensionales Beispiel eines Binärbaums mit „Lokalisierungscodes“ nach Frisken und Perry [47]. Oben: Intervall mit räumlicher Unterteilung. Unten: Zugehöriger Binärbaum mit Lokalisierungscodes für die entsprechenden Knoten.

Zugriff auf einen Bereich der Belegungskarte

Betrifft eine Operation einen bestimmten Bereich der Karte, so sind im Allgemeinen mehrere Blattknoten aus unterschiedlichen Octrees \mathcal{O} betroffen. Die Ermittlung der Liste aller Blattknoten $n_i \in \mathbb{N}$ eines Kartenausschnittes Q wird im Folgenden beschrieben. Der Kartenausschnitt wird durch einen achsenparallelen Quader Q definiert, der durch zwei Punkte \mathbf{p}_a und \mathbf{p}_b begrenzt wird. In diesem Bereich werden alle Octrees $O_{j_k} \in \mathcal{O}$ durch (3.20) bestimmt.

$$\text{idx} : \mathbf{p}_a, \mathbf{p}_b \rightarrow \{j_k\}, \quad \mathbf{p}_a, \mathbf{p}_b \in \mathbb{R}^3, j_k \in \mathbb{N} \quad (3.20)$$

Anschließend erfolgt die Ermittlung der betroffenen Knoten \mathbf{N}_j für jeden Octree in \mathcal{O} . Die Lokalisierungs-codes lassen sich auch für Bereichsabfragen nutzen, in dem für den Quader Q der kleinste gemeinsame Vaterknoten T ermittelt wird. Dies erfolgt durch eine bitweise Oder-Operation der jeweiligen Lokalisierungs-codes der Punkte \mathbf{p}_a und \mathbf{p}_b . Der resultierende Lokalisierungscode führt direkt zu T .

In dem ermittelten Teilbaum T wird die Menge \mathbf{N}_j aller Blattknoten bestimmt, die ganz oder teilweise in Q enthalten sind. Die Menge aller betroffenen Blattknoten \mathbf{N} wird über die Vereinigung der Blattknoten pro Octree gebildet:

$$\mathbf{N} = \bigcup \mathbf{N}_j \quad (3.21)$$

Paralleler Zugriff

Für eine Bearbeitung der Datenstruktur in einer Laufzeitumgebung, die Nebenläufigkeit unterstützt, muss die Sperrung einzelner Bereiche der Karte zur exklusiven Bearbeitung mittels wechselseitigem Ausschluss vorgenommen werden. Sowohl durch das dünn besetzte Voxelgitter als auch durch die Octrees als Elemente des Voxelgitters ergibt sich eine Unterteilung der Karte auf unterschiedlichen Hierarchiestufen. Damit lässt sich der Zugriff auf die Datenstruktur auf unterschiedlichen Ebenen sperren.

Das Integrieren einer Messung betrifft in der Regel nur einen begrenzten Bereich der Datenstruktur. Im Rahmen dieser Arbeit wird der wechselseitige Ausschluss auf Ebene der Wurzelknoten der Octrees vorgenommen. So können Messungen, die unterschiedliche Octrees betreffen, parallel integriert werden.

Projektion der Karte in eine Ebene

Gegebenenfalls ist für Algorithmen lediglich eine zweidimensionale Projektion der dreidimensionalen Karte notwendig. Insbesondere für Bildverarbeitungs-algorithmen wird ein zweidimensionales Bild benötigt. Zur Visualisierung der dreidimensionalen Karte in einer zweidimensionalen Ansicht bietet sich eine zweidimensionale Textur ebenso an.

Eine Draufsicht erhält man dadurch, dass die dreidimensionale Karte auf die x-y-Ebene des Kartenkoordinatensystems $\mathbf{K}\mathbf{K}$ projiziert wird. Unterschiedliche Zellgrößen werden hierbei aufgelöst, so dass alle Zellen mit konstant maximaler Auflösung abgebildet werden. Die Größe der Zellen der zweidimensionalen Textur entspricht der Zellgröße der Octrees auf maximaler

Hierarchietiefe c_{\max} . Für die Projektion ermittelt ein rekursiver Algorithmus alle Blattknoten, deren Wahrscheinlichkeiten dann in einem zweidimensionalen Raster geeignet akkumuliert werden.

Analog zu diesem Verfahren lassen sich auch Projektionen auf die x-z- und auf die y-z-Ebene realisieren.

3.6 Schätzung des Belegungsstatus

Jeder Octree-Knoten n in der Datenstruktur wird als Bereich der realen Welt interpretiert, für den der Belegungsstatus (belegt oder frei) geschätzt wird. Die Belegungswahrscheinlichkeiten mehrerer Messungen werden über die Zeit akkumuliert. Im Folgenden werden zwei Möglichkeiten zur Schätzung des Belegungsstatus erläutert.

3.6.1 Binäres Bayes-Filter

Das binäre Bayes-Filter mit statischem Zustand wird in der Literatur häufig für die Parameterschätzung in Belegungskarten verwendet, siehe Thrun [130]. Der Belegungsstatus der Karte \mathbf{K} zum Zeitpunkt t wird durch die Belegungswahrscheinlichkeit $p_t(\mathbf{K})$ ausgedrückt, siehe Gleichung (3.22). Sie ist abhängig von allen bisherigen Messungen. Sie ist jedoch nicht abhängig von den Aktionen des Fahrzeuges, da sich der Zustand der Karte durch Bewegungen des Fahrzeuges nicht ändert. Es gibt in diesem Fall also keinen Steuervektor \mathbf{u} , wie sonst bei Bayes-Filtern üblich, siehe Abschnitt 2.3.

$$p_t(\mathbf{K}) = p(\mathbf{K}|\mathbf{y}_{1:t}) \quad (3.22)$$

Zur Vereinfachung wird in diesem Kapitel angenommen, dass die Umgebung des Fahrzeuges sich nicht ändert, also statisch ist. Deshalb erhält die Belegungskarte \mathbf{K} keinen zeitlichen Index. In Kapitel 5 werden Methoden beschrieben, die diese Einschränkung aufheben.

Weiterhin wird angenommen, dass die einzelnen Zellen n_i der Karte voneinander unabhängig sind:

$$p(\mathbf{K}|\mathbf{y}_{1:t}) = \prod_i p(n_i = \text{OCC}|\mathbf{y}_{1:t}), \quad n_i \in \mathbf{K} \quad (3.23)$$

Formel (3.23) gibt den Belegungszustand für die gesamte Karte an, für den Belegungszustand eines einzelnen Knoten n_i gilt zum Zeitpunkt t :

$$p_t(n_i = \text{OCC}) = p(n_i = \text{OCC} | \mathbf{y}_{1:t}) \quad (3.24)$$

Ein Knoten n_i kann dabei entweder den Wert belegt (OCC) oder den Wert frei (EMP) annehmen, wobei gilt: $p(\overline{\text{OCC}}) = 1 - p(\text{OCC})$.

In Abschnitt 2.3.1 wurde das binäre Bayes-Filter (2.42) hergeleitet. Angewandt auf die Zustandsschätzung einer Zelle der Belegungskarte lautet die Formulierung über das logarithmische Quotenverhältnis:

$$\text{logit}_t(n_i) = \log \frac{p(n_i | \mathbf{y}_t)}{1 - p(n_i | \mathbf{y}_t)} + \text{logit}_{t-1}(n_i) - \text{logit}_0(n_i) \quad (3.25)$$

Jede Zelle der Karte wird mit logit_0 (2.41) initialisiert. Für den initialen Zustand wird die maximale Entropie angenommen; das heißt, dass als Zustand für eine Zelle frei und belegt als gleich wahrscheinlich angenommen wird. Somit gilt $p(n) = p(\bar{n}) = \frac{1}{2}$, und logit_0 ergibt sich zu 0, wie (2.41) zeigt. Die rekursive Gleichung (3.25) vereinfacht sich damit zu:

$$\text{logit}_t(n_i) = \log \frac{p(n_i | \mathbf{y}_t)}{1 - p(n_i | \mathbf{y}_t)} + \text{logit}_{t-1}(n_i) \quad (3.26)$$

Diese rekursive Formulierung erleichtert die Integration aufeinanderfolgender Messdaten. Sie wird in dieser Form für die Implementierung in diesem und in den folgenden Kapiteln eingesetzt.

3.6.2 Evidenztheorie

In der Regel wird die klassische Wahrscheinlichkeitstheorie verwendet, um den Zustand einer Zelle einer Belegungskarte zu modellieren. Die Wahrscheinlichkeit, dass eine Zelle belegt ist $p(n)$, entspricht dabei der Wahrscheinlichkeit, dass die Zelle nicht frei ist:

$$p(n) = \overline{p(\bar{n})} \quad (3.27)$$

$$= 1 - p(\bar{n}) \quad (3.28)$$

Dies hat einen entscheidenden Nachteil bei der Modellierung der Zustände von Zellen einer Belegungskarte:

Beispiel 3.3 *Ein Sensor misst, dass eine Zelle belegt ist. Dem Sensor kann zu 60% vertraut werden. Diese Messung besagt nun nicht automatisch, dass dieselbe Zelle zu 40% frei ist, sondern dass der Zustand der Zelle zu 40% unbekannt ist.*

Mit der Evidenztheorie nach Dempster und Shafer DST (engl. Dempster-Shafer Theory) [22, 110] lässt sich dieses Nicht-Wissen über den Belegungszustand korrekt modellieren.

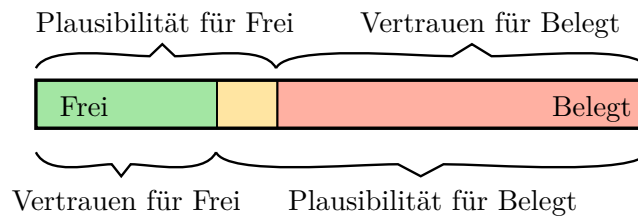


Abbildung 3.16 – Beispiel des modellierten Belegungszustandes einer Zelle durch die DST. Zusätzlich zum Vertrauen in die Zustände „frei“ und „belegt“ wird eine Unsicherheit modelliert.

Abbildung 3.16 zeigt exemplarisch den Belegungszustand einer Zelle, der mit Hilfe der DST modelliert wird. Das Vertrauen für „frei“ ist 30%, das Vertrauen für „belegt“ ist 60%. Dadurch wird nicht das ganze Intervall abgedeckt. Die restlichen 10% stellen gewissermaßen eine Unsicherheit dar. Diese fließt jeweils in die Plausibilitäten ein: Die Plausibilität für „frei“ ist 40%, wogegen die Plausibilität für „belegt“ 70% beträgt. Der Unterschied zwischen Bayes- und Dempster-Shafer-Methoden wird von Dietmayer [23] erläutert.

In der Literatur wurde bereits mehrfach die Modellierung eines Belegungsgitters mit der DST beschrieben [89, 30]. In Moras [87] findet sich erstmals ein Hinweis auf einen praktischen Nutzen der Modellierung mit der DST gegenüber der Wahrscheinlichkeitstheorie: Durch bewegte Objekte entstehen in den überquerten Zellen kurzfristig Inkonsistenzen. Diese können genutzt werden, um bewegte Objekte zu detektieren.

Mit der in Abschnitt 3.8 beschriebenen dynamischen Zellverfeinerung ergibt sich ebenfalls eine interessante Möglichkeit für den praktischen Nutzen der DST: Zellen mit kleiner Unsicherheit (das heißt, es liegen bereits Messungen vor) und einem ausgewogenen Verhältnis von Evidenz für einen freien Bereich und Evidenz für einen belegten Bereich lassen auf widersprüchliche Messungen schließen. In der Realität ist ausgeschlossen, dass ein Punkt gleichzeitig belegt und frei ist. Wenn der inkonsistente Zustand nicht durch ein bewegtes Objekt entstanden ist, lässt sich also darauf schließen, dass die Diskretisierung der Belegungskarte an dieser Stelle zu grob ist. Gerade an Randbereichen von Objekten tritt dies regelmäßig auf. So ergibt sich durch die DST ein sehr guter Indikator dafür, ob die Seitenlänge einer Zelle zu groß

gewählt ist und lässt sich automatisch durch die dynamische Zellverfeinerung verkleinern, bis der Rand des Objektes genau erfasst wird und sich keine widersprüchlichen Messungen mehr ergeben. Umgekehrt lassen sich homogene Bereiche mit dieser Methode vergrößern, womit der Speicherplatzbedarf und Rechenzeit sinkt. Dieses Verfahren wurde in [152] patentiert.

3.6.3 Zeitliches Abklingen

Durch Unsicherheiten in der Schätzung der Pose nimmt die Qualität der Karte mit der Zeit ab. Durch das regelmäßige Anwenden eines Verfallsfaktors auf die komplette Karte, analog zu [28], kann dem entgegengewirkt werden. Hierdurch wird der Belegungszustand von Zellen, die in der nahen Vergangenheit keine Messungen erhalten haben, gewissermaßen zurückgesetzt. Ein Abklingen wird durch die Integration einer Belegungswahrscheinlichkeit von $\frac{1}{2} - \epsilon$ für jede Zelle n_i erreicht:

$$p_t(n_i) = p_{t-1}(n_i) \left(\frac{1}{2} - \epsilon \right) \quad (3.29)$$

Als logarithmisches Quotenverhältnis ausgedrückt gilt:

$$\text{logit}_t(n_i) = \text{logit}_{t-1}(n_i) + \log \left\{ \frac{\frac{1}{2} - \epsilon}{\frac{1}{2} + \epsilon} \right\} \quad (3.30)$$

ϵ ist abhängig vom Messrauschen der eingesetzten Sensoren sowie von der Anwendung und wird empirisch ermittelt.

3.7 Operationen auf der Datenstruktur

Zu Beginn ist der Zustand der Belegungskarte unbekannt. Allen Bereichen wird eine Belegungswahrscheinlichkeit von $p(n) = \frac{1}{2}$ zugeordnet. Für jede Messung, die eingetragen werden soll, werden zunächst an den betroffenen Stellen Octrees mit einem Wurzelknoten in das Voxelgitter eingesetzt. Um eine Sensormessung in erhöhter Auflösung in die Karte einzutragen, müssen die entsprechenden Wurzelknoten der Octrees anschließend rekursiv durch das Anlegen von Kindknoten verfeinert werden.

Im Gegensatz dazu können im späteren Verlauf hoch aufgelöste Bereiche, die nicht länger von Interesse sind, vergrößert werden, um Speicher- und Rechenaufwand zu reduzieren. Dazu sind Verfeinerungs- und Vergrößerungsoperation notwendig, die ein- oder mehrfach auf die entsprechenden Bereiche angewandt werden. Diese Operationen werden in den beiden

folgenden Abschnitten beschrieben.

3.7.1 Verfeinern eines Kartenausschnittes

Ein Kartenausschnitt wird verfeinert, indem alle betroffenen Octrees in einem Quader Q verfeinert werden. Dazu werden durch Gleichung (3.20) alle Octrees bestimmt, die durch den Quader ganz oder teilweise überdeckt werden. Die im Folgenden beschriebene Verfeinerungsoperation für Octree-Teilbäume wird anschließend auf alle so ermittelten Octrees angewandt.

Zur Verfeinerung eines Teilbaumes wird der Knoten, der den Teilbaum repräsentiert, rekursiv unterteilt. Dabei wird der Knoten in acht gleich große Knoten mit halber Seitenlänge unterteilt, die zusammen genommen wieder denselben Raum einnehmen wie der ursprüngliche Knoten. Die Belegungswerte für die Kindknoten n_i mit $i = 1 \dots 8$ werden nach Gleichung (3.31) direkt vom Vaterknoten n_p übernommen. In Bezug auf das Volumen V der Knoten ergibt sich so für die Belegungswahrscheinlichkeit in diesem Bereich keine Änderung, wie in Gleichung (3.32) dargestellt. Entsprechen die neu entstandenen Kindknoten noch nicht dem gewünschten Detaillierungsgrad, werden diese ebenfalls rekursiv verfeinert.

$$p(n_i) = p(n_p), \quad i \in 1 \dots 8 \quad (3.31)$$

$$p(n_p) \cdot V_{n_p} = \sum_{i=1}^8 p(n_i) \cdot V_{n_i} \quad (3.32)$$

3.7.2 Reduktion der Auflösung eines Kartenausschnittes

Analog zur Verfeinerung wird bei einem Kartenausschnitt die Auflösung reduziert, indem die Hierarchiestufe aller betroffenen Octrees in einem Quader Q verkleinert wird. Dazu werden durch Gleichung (3.20) alle Octrees bestimmt, die durch den Quader ganz oder teilweise überdeckt werden. Die im Folgenden beschriebene Operation wird anschließend auf alle so ermittelten Octrees angewandt.

Ein Teilbaum wird vergrößert, indem die Kindknoten des Knotens, der den Teilbaum repräsentiert, entfernt werden. Eine maximale Vergrößerung liegt vor, wenn ein Octree lediglich aus einem Wurzelknoten besteht. Bevor die Kinder eines Knotens entfernt werden, wird der Mittelwert über die Belegungswahrscheinlichkeiten der Kindknoten berechnet. Diese Wahrscheinlichkeit wird dann als gemittelte Wahrscheinlichkeit des neuen Blattknotens gesetzt. Anschließend werden alle Kindknoten entfernt, so dass der Knoten wieder zum Blattknoten wird. Handelt es sich bei den Kindknoten nicht um Blattknoten, werden diese zuvor rekursiv vergrößert.

Die Reduktion der Auflösung von Bereichen der Karte ist sinnvoll, um Ressourcen in den Bereichen der Karte zu sparen, in denen keine hohe Auflösung mehr benötigt wird. Dies kann abstandsabhängig zum Sensorfahrzeug erfolgen: Entfernt sich ein Octree zu weit vom Sensorfahrzeug, so wird eine Vergrößerung durchgeführt. Für eine Fahrbahnschätzung oder eine grobe Lokalisierung genügt eine grob aufgelöste Karte. Bereiche, die gänzlich unwichtig geworden sind, werden komplett gelöscht, indem die zugehörigen Octrees mit allen Knoten gelöscht werden.

3.7.3 Integration von Messungen

Alle Komponenten, die zur Integration von Messungen in die Belegungskarte notwendig sind, sind in den Abschnitten Sensormodell (Abschnitt 3.4), Datenstruktur (Abschnitt 3.5) und Schätzung des Belegungsstatus (Abschnitt 3.6) beschrieben. Liefert ein Sensor mehrere Messungen gleichzeitig, so werden diese sequentiell eingetragen. Im Folgenden wird demnach lediglich von der Integration einer Messung gesprochen.

Messungen eines aktiven Sensors liefern in der Regel eine Indikation über einen belegten Bereich, wie den Abstand zu einem Fahrzeug oder einem Gebäude. Für die Integration einer Messung in die Karte werden folgende Schritte durchgeführt:

Betroffenen Kartenausschnitt ermitteln Die Position der Messung in der Karte ist durch \mathbf{p}_{KK} gegeben, siehe Gleichungen (3.7) und (3.17).

Aufgrund der Messunsicherheit betrifft eine Messung in der Regel nicht nur einen Blattknoten. Um den von der Integration einer Messung betroffenen Kartenausschnitt zu ermitteln, wird die Kovarianzmatrix \mathbf{C}_{mc} herangezogen, die ein Modell für die Unsicherheit darstellt (Abschnitt 3.4.1).

Geometrisch kann das $\pm 3\sigma$ -Intervall der Kovarianzmatrix \mathbf{C}_{mc} als Ellipsoid betrachtet werden. Der umschreibende Quader Q um dieses Ellipsoid an der Position \mathbf{p}_{KK} legt den Bereich fest, in dem die Belegungskarte modifiziert wird.

Nun muss der Fall betrachtet werden, dass die Region Q mehrere Octrees \mathcal{O} in \mathcal{V} berührt. Die Menge der Octrees \mathcal{O} wird daher zunächst, wie in Abschnitt 3.5.4 beschrieben, bestimmt. Die folgenden Operationen erfolgen demnach pro Octree $O_j \in \mathcal{O}$. Die in Abschnitt 3.5.4 beschriebenen Lokalisierungscode werden eingesetzt, um den Teilbaum T zu bestimmen, den es zu modifizieren gilt.

Kartenausschnitt verfeinern Eine oder mehrere Anwendungen, die auf Basis der Belegungskarte arbeiten, definieren die Auflösung der Karte, wie in Abschnitt 3.8 eingeführt wird. Vor der Integration der Messung wird die Auflösung im ermittelten Kar-

tenausschnitt Q deshalb gegebenenfalls wie von der Anwendung gefordert erhöht. Die Verfeinerung wird in Abschnitt 3.7.1 beschrieben. Nach einer Verfeinerung muss der zu modifizierende Teilbaum T gegebenenfalls erneut bestimmt werden.

Betroffene Zellen ermitteln Um die Belegungskarte zu modifizieren, ist eine Liste aller betroffenen Blattknoten n_i notwendig. In dem ermittelten Teilbaum T wird die Menge \mathbf{N} aller Blattknoten bestimmt, die ganz oder teilweise in Q enthalten sind, siehe Abschnitt 3.5.4.

Belegungswahrscheinlichkeiten berechnen Für jeden Knoten $n_i \in \mathbf{N}$ wird die Belegungswahrscheinlichkeit $r(n_i)$ nach Gleichung (3.33) berechnet. Die Belegungswahrscheinlichkeit wird für jeden Knoten näherungsweise lediglich an dessen Mittelpunkt $m(n)$ bestimmt. Infolgedessen ist als weiterer Schritt eine Normierung notwendig.

$$r(n_i) = \exp \left\{ -\frac{1}{2} \left(m(n_i) - \mathbf{p}^{\text{KK}} \right)^T \cdot \mathbf{C}_{m_c}^{-1} \cdot \left(m(n_i) - \mathbf{p}^{\text{KK}} \right) \right\} \quad (3.33)$$

Wahrscheinlichkeiten normieren Nach der Berechnung der Wahrscheinlichkeiten muss aufgrund der Annäherung in den Zellen eine Normierung durchgeführt werden. Dafür wird der Kehrwert der Summe aller betroffenen Knoten in allen Octrees \mathcal{O} als Normierungsfaktor eingesetzt. Für einen Knoten n_i ist die zu integrierende Belegungswahrscheinlichkeit $p(n_i)$ dann definiert durch:

$$p(n_i) := \frac{1}{2} + \frac{r(n_i)}{2 V(n_i) \sum_j^{\mathcal{O}} \sum_k^{\mathbf{N}} r(n_{j,k})} \quad (3.34)$$

$V(n_i)$ beschreibt das Volumen des Knotens n_i .

3.7.4 Löschen freier Bereiche

Ein Lidar-Sensor oder Laserscanner besitzt stark gebündelte Strahlkeulen und misst ein Objekt in der Regel direkt an, so dass hier auch Aussagen über den Freiraum getroffen werden können: Der zurückgelegte Weg vom Sensor zum angemessenen Objekt muss frei sein. Diese Information kann ebenfalls in die Karte integriert werden.

Bei anderen aktiven Sensoren wie einem Radarsensor kann der Freiraum nicht in der Art und Weise modelliert werden, da die Antennenkeule deutlich weniger stark fokussiert ist und Objekte unter Umständen auch über Mehrwegeempfang detektiert werden.

Freiraumbestimmung durch Lidar-Sensorik

Die Messdaten eines Lidar-Sensors oder Laserscanners enthalten neben den Abstandswerten noch die implizite Information über den Freiraum zwischen Messung und Sensor. Erhält ein Scanner in einem Winkelbereich eine Messung in 50 m Entfernung, so hat der Laserstrahl den Weg dorthin ungehindert zurückgelegt. Das lässt darauf schließen, dass der Bereich zwischen Sensor und angemessenem Objekt frei ist [139]. In diesem Abschnitt wird beschrieben, wie diese Information in die Karte einfließen kann. Bei einem mehrzielfähigen Laserscanner wird dabei lediglich der Freiraum zwischen dem Sensor und des ersten Ziels integriert. Der Raum zwischen zwei Zielen kann im Allgemeinen nicht als frei angenommen werden, wie Abbildung 3.17 verdeutlicht. Wenn der Scanner in einem Winkelschritt keine gültige Messung liefert, wird ein Bereich von einem Viertel der maximalen Reichweite als frei angenommen.

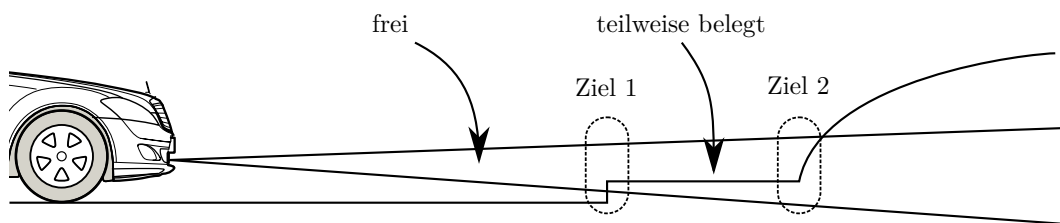


Abbildung 3.17 – Ein mehrzielfähiger Laserscanner ist in der Lage, pro ausgesendetem Laserpuls mehrere Abstandsmessungen auszuwerten. Hier werden durch eine hohe vertikale Strahlaufweitung zwei Ziele detektiert: Der Bordstein und die dahinterliegende Böschung. Der Bereich zwischen Sensor und Ziel 1 kann als frei angenommen werden, der Bereich zwischen Ziel 1 und 2 dagegen jedoch nicht.

Für die Reduzierung der Belegungswahrscheinlichkeit in einem als frei gemessenen Raum werden folgende Schritte durchgeführt:

Betroffenen Kartenausschnitt ermitteln Die Bestimmung des betroffenen Kartenausschnittes erfolgt analog zur Vorgehensweise bei belegten Bereichen. Lediglich das grundlegende geometrische Modell unterscheidet sich: Der gemessene Freiraum wird durch die in Abschnitt 3.4.2 beschriebene Pyramide modelliert. Der umschreibende Quader Q um diese Pyramide legt den Bereich fest, in dem die Belegungskarte modifiziert wird.

Kartenausschnitt verfeinern Um Freiraum explizit zu modellieren, müssen im Bereich Q gegebenenfalls Zellen angelegt und auf die gewünschte Auflösung gebracht werden. Dies erfolgt analog zur Verfeinerung.

Betroffene Zellen ermitteln Um die Belegungskarte zu modifizieren, ist eine Liste aller betroffenen Blattknoten n_i notwendig. Dazu wird für jede Zelle in Q bestimmt, ob sie innerhalb des modellierten Freiraumes liegt oder diesen schneidet. Die fünf Ebenen der

Pyramide werden dabei durch jeweils einen Normalen- und einen Stützvektor repräsentiert. Die Normalenvektoren zeigen dabei stets in das Innere der Pyramide. Mit dem Skalarprodukt kann für einen Punkt effizient bestimmt werden, ob dieser innerhalb oder außerhalb der Pyramide liegt.

Für alle Knoten n_i in Q werden jeweils alle acht Eckpunkte getestet. Für jeden Knoten n_i trifft einer der folgenden drei Fälle zu:

- (a) Alle acht Eckpunkte liegen oberhalb aller fünf Ebenen: Der Knoten liegt vollständig in der Pyramide.
- (b) Alle acht Eckpunkte liegen unterhalb mindestens einer der fünf Ebenen: Der Knoten liegt außerhalb des Strahles und schneidet ihn auch nicht.
- (c) Sonst: Die Zelle berührt die Pyramide.

Im Fall (c) wird der Knoten so lange verfeinert, bis die gewünschte Auflösung erreicht ist. Der Algorithmus wird dann rekursiv auf die Kindknoten angewandt, wobei bereits berechnete Ecken wieder verwendet werden um eine hohe Performanz zu gewährleisten.

Belegungswahrscheinlichkeiten reduzieren Im Fall (a) liegt der Knoten vollständig innerhalb des freien Raumes. Deshalb wird die Belegungswahrscheinlichkeit hier deutlich reduziert, es wird der empirisch ermittelte Wert $p(n) = 0,3$ integriert. Im Fall (b) liegt der Knoten außerhalb des Strahls und wird nicht weiter betrachtet. Wenn auf der höchsten Detaillierungsstufe jedoch erneut Fall (c) eintritt, so wird nicht weiter verfeinert. Diskretisierungsfehler werden auf dieser Stufe in Kauf genommen, die Belegungswahrscheinlichkeit wird jedoch nicht mehr so stark reduziert wie in Fall (a). Es wird die ebenfalls empirisch ermittelte Belegungswahrscheinlichkeit von $p(n) = 0,4$ integriert.

Abbildung 3.18 zeigt die Integration einer Freiraummessung an einem Beispiel. Es sind drei Octrees betroffen. Knoten, die den Rand der Pyramide berühren, werden rekursiv verfeinert. Auf der höchsten Detaillierungsstufe wird dann eine Belegungswahrscheinlichkeit von $p(n) = 0,4$ integriert. Alle Knoten, die vollständig im Strahl liegen, müssen nicht rekursiv verfeinert werden. Hier wird die Belegungswahrscheinlichkeit direkt durch eine Integration von $p(n) = 0,3$ reduziert.

Vom Sensorfahrzeug überquerte Bereiche

Belegte Zellen, die vom eigenen Fahrzeug durchquert werden, stellen für sich genommen eine Kollision dar. Resultiert solch eine Zelle jedoch aus Falschmessungen und wird korrekterweise durch eine Schwellwertbildung oder durch eine weitergehende Signalverarbeitung ignoriert, so kann diese in der Karte als frei markiert werden. Dazu werden die Belegungswahrschein-

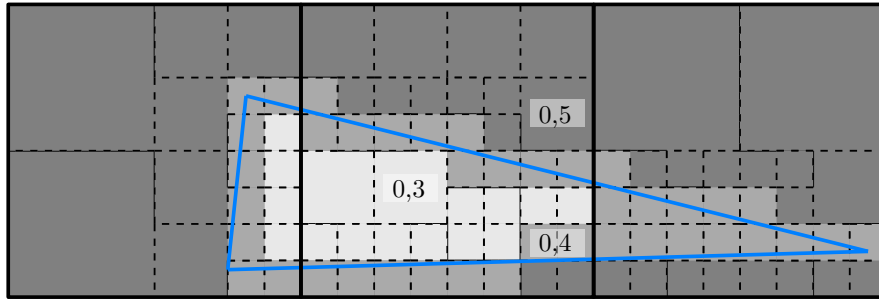


Abbildung 3.18 – Beispiel einer Freiraummessung, die drei Octrees überdeckt. Zur Integration der Belegungswahrscheinlichkeiten werden zunächst die vom Sensormodell berührten Bereiche verfeinert. Die integrierten Belegungswahrscheinlichkeiten sind in Graustufen kodiert. Für Zellen, die komplett in der Pyramide liegen, wird die Wahrscheinlichkeit mit $p(n)=0,3$ stärker reduziert als für Zellen, welche die Pyramide lediglich berühren: $p(n)=0,4$.

lichkeiten der Zellen, die vom Fahrzeug überquert werden, zurückgesetzt. Als Modell des Fahrzeuges kommt dabei ein Quader, der den Abmessungen des Fahrzeuges entspricht, zum Einsatz.

Beispielesequenz

Die Integration von Messungen wird an einem Beispiel verdeutlicht. Abbildung 3.19 zeigt eine Beispielsituation, in der ein einzeliger Laserscanner eingesetzt wird. Der Laserscanner ist links in der Abbildung zu sehen, zwei statische Objekte O_1 und O_2 befinden sich auf der rechten Seite in grün. Die Freiräume werden in den Abbildungen der Übersichtlichkeit halber nicht explizit modelliert, so dass alle Belegungswahrscheinlichkeiten $p(n) \leq 0,5$ weiß dargestellt werden.

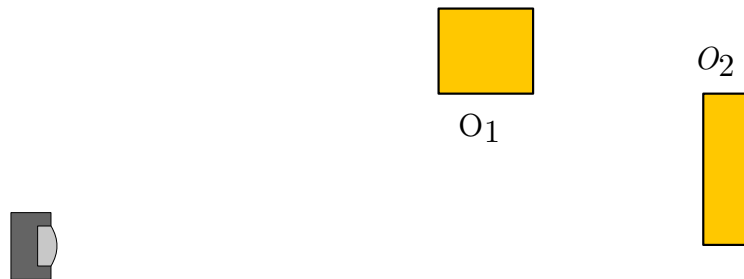


Abbildung 3.19 – Beispielsituation mit einem einzeligen Laserscanner auf der linken Seite und zwei statischen Objekten O_1 und O_2 auf der rechten Seite.

In Abbildung 3.20 wird eine Messung von Objekt O_1 in die Karte integriert. Sensormessungen sind in rot durch das $\pm 3\sigma$ -Ellipsoid dargestellt. Die Summe der Wahrscheinlichkeiten aller

betroffenen Zellen wird auf 1 normiert. Objekt O_2 aus Abbildung 3.19 wird vom Laserscanner noch nicht gesehen.

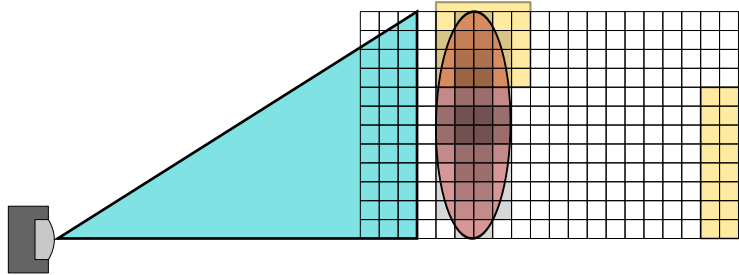


Abbildung 3.20 – Sensormessungen sind in rot durch das $\pm 3\sigma$ -Ellipsoid dargestellt, welches die Messunsicherheit darstellt. Eine Messung wird in die Belegungskarte integriert. Die Summe der Belegungswahrscheinlichkeiten aller Knoten wird auf 1 normiert.

Nach einer Änderung der Sensorposition nach rechts verschwindet Objekt O_1 aus dem Sichtbereich des Sensors (Abbildung 3.21). Stattdessen wird Objekt O_2 erstmalig vom Sensor detektiert. Die neue Messung wird integriert. Der vom Objekt O_2 resultierende gemessene Freiraum des Sensors wird genutzt, um die zuvor integrierten Belegungswahrscheinlichkeiten von Objekt O_1 zu reduzieren.

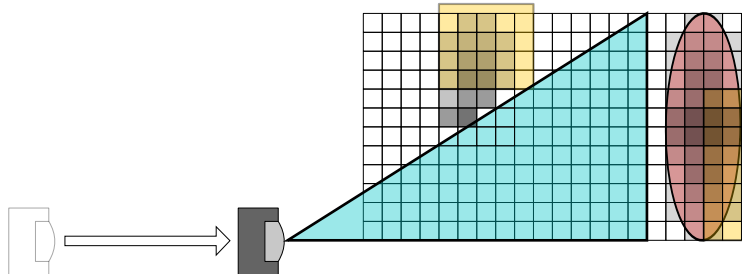


Abbildung 3.21 – Nachdem sich die Position des Sensors geändert hat, wird ein weiteres Objekt vom Sensor detektiert. Es wird in die Karte integriert. Über den gemessenen Freiraum werden Belegungswahrscheinlichkeiten des zuerst angemessenen Objektes O_1 wieder reduziert.

3.7.5 Mengenoperationen auf Karten

Als Grundlage für weiterführende Algorithmen sind Mengenoperationen auf Belegungskarten notwendig. Insbesondere bei der Detektion bewegter Objekte in Kapitel 5 kommen Mengenoperationen zum Einsatz. Wird dabei die Messkarte separat von der statischen Karte betrachtet, ist die Vereinigung zweier Karten notwendig. Weiterhin ist die Differenz und die Schnittmenge zweier Karten von Interesse.

Vereinigung zweier Karten

In Foley [45] wird die Vereinigung zweier Quadrees exemplarisch beschrieben. Basierend darauf kann auch die Vereinigung zweier Belegungskarten A und B zu U durchgeführt werden:

$$U = A \cup B \quad (3.35)$$

Für jeden Octree-Index j werden die korrespondierenden Octrees aus den Karten A und B betrachtet. Enthält entweder A oder B einen Octree an Index j , so wird dieser direkt in U übernommen.

Enthalten sowohl A als auch B einen Octree an Index j , so wird ein rekursiver Algorithmus auf die Wurzelknoten angewandt, um die Octrees zu vereinigen. Dabei müssen Bereiche aus A oder B gegebenenfalls verfeinert werden, damit die Knoten aus A und B jeweils auf derselben Hierarchiestufe sind. Für die Belegungswahrscheinlichkeiten des Ergebnisknotens wird jeweils das Maximum der Belegungswahrscheinlichkeiten der zugehörigen Knoten aus A und B gewählt.

Differenz zweier Karten

Die Subtraktion läuft analog zur Vereinigung und wird folgendermaßen notiert:

$$D = A - B \quad (3.36)$$

Für jeden Octree-Index j in A oder B werden folgende Betrachtungen angestellt: Tritt O_j lediglich in Karte A auf, wird der Octree direkt in die Ergebniskarte übernommen. Tritt er lediglich in Karte B auf wird er nicht übernommen. Enthält A und B einen Octree an Index j , so wird ein rekursiver Algorithmus zur Subtraktion angewandt.

Dabei müssen Bereiche aus A oder B gegebenenfalls verfeinert werden, damit die Knoten aus A und B jeweils auf derselben Hierarchiestufe sind. Die Belegungswahrscheinlichkeit eines Ergebnisknotens $n_{D,i}$ auf maximaler Hierarchietiefe entsteht vereinfachend durch eine Schwellwertbildung: Wenn $p(n_{B,i}) \leq \frac{1}{2}$ wird $p(n_{D,i})$ auf $p(n_{A,i})$ gesetzt, andernfalls auf 0.

Schnittmenge zweier Karten

Die Schnittbildung läuft ebenso analog zur Vereinigung. Für jeden Octree-Index j in A oder B werden nur Octrees betrachtet, die sowohl in A als auch in B enthalten sind. Ein rekursiver

Algorithmus berechnet die Schnittmenge. Dabei müssen Bereiche aus A oder B gegebenenfalls verfeinert werden, damit die Knoten aus A und B jeweils auf der gleichen Hierarchiestufe sind. Die Belegungswahrscheinlichkeit eines Knotens in C entsteht dann durch Minimumbildung der Belegungswahrscheinlichkeiten der zugehörigen Knoten aus A und B .

$$C = A \cap B \tag{3.37}$$

3.8 Bestimmung des Detaillierungsgrades der Belegungskarte

Bei existierenden Verfahren zur Kartierung der Umgebung mit Belegungskarten ist es notwendig, die gesamte Karte mit derselben hohen Auflösung darzustellen. Die hierarchische Datenstruktur aus Abschnitt 3.5 erlaubt es, Bereiche mit hoher Auflösung und Bereiche mit niedriger Auflösung zu definieren.

Ein wichtiger Punkt bei der Wahl des dynamischen Detaillierungsgrades einer Region ist, dass nicht alle Knoten automatisch bis zur gewünschten Auflösung verfeinert werden. Lediglich an Stellen, an denen neue Messungen auftreten, wird der Bereich vor dem Integrieren der Messung auf die gewünschte Auflösung verfeinert. Leere Bereiche oder Bereiche, in denen keine Beobachtungen mehr auftreten, bleiben also unberührt. Dies würde auch keinen Vorteil bieten, da durch die nachträgliche Verfeinerung die ursprüngliche Messung nicht mehr zur Verfügung steht. Lediglich durch ein Speichern vorangegangener Messungen bringt eine nachträgliche Verfeinerung einen Vorteil, was in dieser Arbeit jedoch nicht verfolgt wird.

3.8.1 Detaillierung abhängig vom Einsatzbereich der Umgebungskarte

Eine Belegungskarte lässt sich für vielerlei Anwendungen einsetzen. Dadurch ergeben sich für die Detaillierungssteuerung unterschiedliche Anforderungen. Für eine PreCrash Anwendung ist beispielsweise der Bereich direkt vor dem Fahrzeug wichtiger als weiter entfernte Objekte. Für eine merkmalsbasierte Navigation auf Punktlandmarken muss die Position der Landmarken möglichst genau bestimmt werden, wohingegen Objekte wie Büsche ignoriert werden können. Für eine Parkfunktion soll die Kontur der Parklücke möglichst exakt dargestellt werden können, wogegen der gegenüberliegende Fahrbahnrand nicht so interessant ist.

Abhängig von dem Anwendungsfall oder den Anwendungsfällen ist es von Vorteil, die Auflösung der Karte gezielt steuern zu können. Dies wird über Funktionen realisiert, die den Detaillierungsgrad für jeden Bereich der Karte definieren. Bei mehreren Anwendungen wird der maximale Detaillierungsgrad gewählt, um alle Anwendungen optimal bedienen zu können.

Der vorgegebene Detaillierungsgrad LOD (engl. Level of Detail) für einen Punkt $\mathbf{p} \in \mathbb{R}^3$ wird als Funktion $\text{LOD}(\mathbf{p})$ definiert, welche die entsprechende Hierarchietiefe der Octrees angibt:

$$\text{LOD}(\mathbf{p}) := \max_{i=1 \dots n} \text{LOD}_i(x) \quad \forall \mathbf{p} \in \mathbb{R}^3, \text{LOD}_i \in \text{LOD} \quad (3.38)$$

Die Menge der Funktionen $\text{LOD} = \text{LOD}_i, i \in 1 \dots n$ ist anwendungsabhängig. Im Folgenden werden mehrere Funktionen angegeben. Für eine konkrete Anwendung müssen diese in der Regel angepasst werden. Das Verfahren zur Anpassung des Detaillierungsgrades in hierarchischen Karten wurde in [148] zum Patent angemeldet.

3.8.2 PreCrash-Funktionen

Für eine PreCrash-Anwendung wird eine hohe Auflösung in der Nähe des eigenen Fahrzeuges benötigt, wohingegen weniger relevante Objekte in großer Distanz nicht so hoch aufgelöst werden müssen. Deshalb wird hier LOD zu LOD_{PC} gewählt. LOD_{PC} ist eine Detaillierungsfunktion, welche die Auflösung mit steigender Entfernung zum Sensorfahrzeug in drei diskreten Schritten abstuft:

$$\text{LOD}_{\text{PC}}(\mathbf{p}) = \begin{cases} c_{\max}, & |\text{dist}(\mathbf{x}_F, \mathbf{p})| < c_{\text{PC}_a} \\ c_{\max} - 1, & c_{\text{PC}_a} \leq |\text{dist}(\mathbf{x}_F, \mathbf{p})| < c_{\text{PC}_b} \\ c_{\max} - 2, & \text{sonst} \end{cases} \quad \forall \mathbf{p} \in \mathbb{R}^3 \quad (3.39)$$

$\text{dist}(a, b)$ entspricht dabei dem euklidischen Abstand zwischen den Punkten a and b . Weiterhin gilt $0 < c_{\text{PC}_a} < c_{\text{PC}_b}$. Diese Konstanten werden abhängig von der konkreten Funktion gewählt. Die Position des Sensorfahrzeuges relativ zur Karte lässt sich direkt aus der Transformationsmatrix $\mathbf{T}^{\text{FK} \rightarrow \text{KK}}$ ermitteln. Die Konstante c_{\max} entspricht der maximal gewünschten Octree-Hierarchietiefe. Diese gibt direkt an, wie hoch die dargestellte Auflösung der Datenstruktur sein soll.

Abbildung 3.22 zeigt ein Beispiel, bei dem die Funktion LOD_{PC} (3.39) verwendet wird. Die Auflösung ist somit nahe dem Fahrzeug am höchsten und sinkt mit steigendem Abstand zum Fahrzeug.

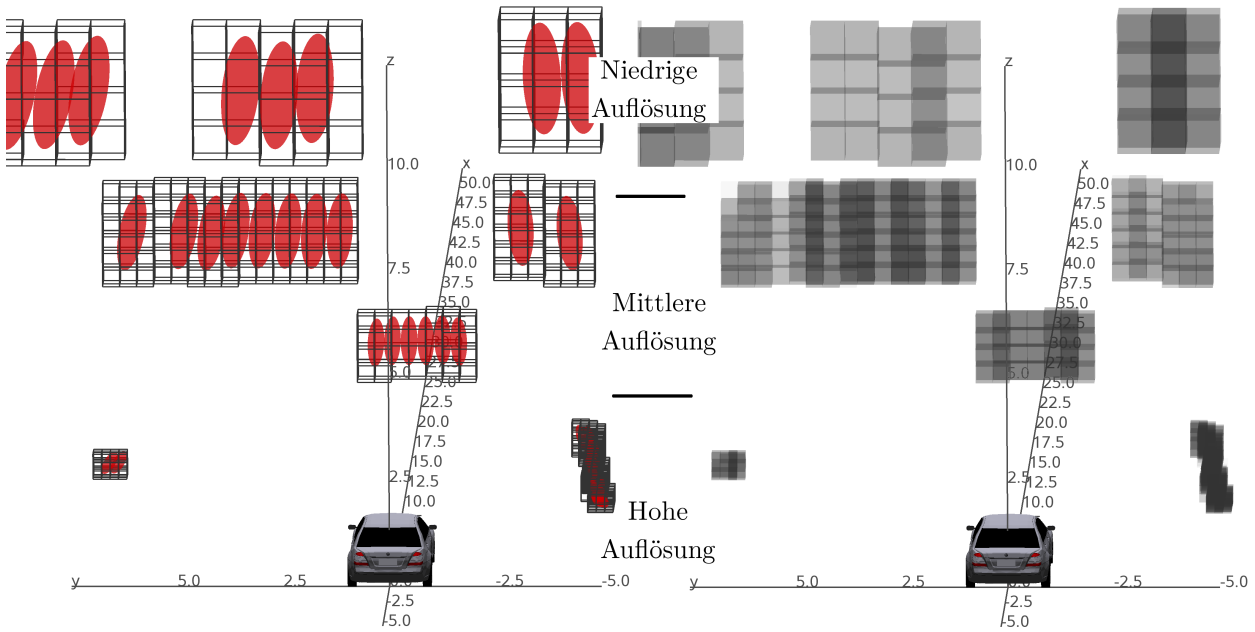


Abbildung 3.22 – Messungen des Hella IDIS Scanners werden unter Berücksichtigung der Messunsicherheiten in die Belegungskarte integriert. Zur Detaillierungssteuerung wird eine PreCrash-Anwendung angenommen. Die Auflösung ist somit nahe dem eigenen Fahrzeug am höchsten und sinkt mit steigendem Abstand. Die unterschiedlichen Auflösungen werden durch die unterschiedlich großen Zellen in der Belegungskarte deutlich. Links: Messunsicherheiten sind durch Ellipsoide visualisiert. Pro Ellipsoid werden alle Zellen in einem umschreibenden Quader ermittelt. Rechts: Die Zellen werden durch eine normalisierte Belegungswahrscheinlichkeit aktualisiert.

Zur Ermittlung der Hierarchietiefe, welche die beste Performanz bietet, sei auf Abschnitt 3.9 verwiesen.

3.8.3 Navigation mit Punktlandmarken

Für die Navigation mit Punktlandmarken muss die Position der Landmarken so genau wie möglich ermittelt werden, siehe beispielsweise Weiss [139]. Eine zu grobe Diskretisierung durch die Belegungskarte ist hier von Nachteil. Eine lokale Erhöhung des Detaillierungsgrades für Punktlandmarken löst dieses Problem, ohne den Speicherbedarf der gesamten Karte zu erhöhen. Für eine Menge S von Punktlandmarken mit $|S| = l$ wird LOD zu $\{\text{LOD}_{\text{LM}_i}\}, i \in 1 \dots l$ gewählt. Die Konstante c_{LM} beschreibt den Radius um eine Landmarke, in dem die Belegungskarte verfeinert wird.

$$\text{LOD}_{\text{LM}_i}(\mathbf{p}) = \begin{cases} c_{\text{max}}, & |\text{dist}(s_i, \mathbf{p})| < c_{\text{LM}} \\ c_{\text{max}} - 2, & \text{sonst} \end{cases} \quad \forall \mathbf{p} \in \mathbb{R}^3, s_i \in S \quad (3.40)$$

3.8.4 Parkfunktionen

Um eine Parklücke zu detektieren, muss die Kontur der Parklücke sehr genau bestimmt werden. Dazu ist eine hohe Zellaufösung notwendig. Es ist jedoch nicht effizient, die gesamte Szene in hoher Auflösung darzustellen. Die dynamische Detaillierung der Karte bietet die Möglichkeit lediglich entlang der Kontur der Parklücke eine hohe Auflösung zu verwenden. Die Detaillierungssteuerung für Parkfunktionen wird ausführlich in Abschnitt 6.2.4 behandelt.

3.9 Evaluierung der Effizienz der Algorithmen

Zur Ermittlung der Effizienz der in diesem Kapitel beschriebenen Belegungskarte werden die benötigte Rechenzeit und der Speicherplatzbedarf beim Aufbau der Karte anhand realer Messdaten betrachtet. Dazu werden unterschiedliche Konfigurationen evaluiert und miteinander verglichen. Die Ergebnisse der folgenden Auswertung werden als Box-Whisker-Plots dargestellt.

3.9.1 Mögliche Konfigurationen der Belegungskarte

Die beschriebene Datenstruktur ist bei konstanter minimaler Zellgröße in Bezug auf die Hierarchietiefe flexibel, wie in Abbildung 3.14 verdeutlicht wird. Um die effizienteste Variante zu ermitteln, werden unterschiedliche Kombinationen von maximaler Octree-Hierarchietiefe c_{max} und Kantenlänge der Octrees miteinander verglichen. Die maximale Zellgröße wird dabei auf beispielsweise $(0,1\text{ m})^3$ festgehalten.

Beispiel 3.4 *Eine Zellgröße von $(0,1\text{ m})^3$ kann durch Octrees der Kantenlänge $0,1\text{ m}$ und einer maximalen Hierarchietiefe von $c_{max} = 0$ erreicht werden. Dieselbe Zellgröße wird erreicht durch $0,2\text{ m}$ Kantenlänge und einer Hierarchietiefe von $c_{max} = 1$. Analog lassen sich beliebig große Octrees verwenden, wenn die Hierarchietiefe entsprechend erhöht wird.*

Im ungünstigsten Fall WC (engl. Worst Case) sind belegte oder freie Zellen räumlich so verteilt, dass die dünn besetzte hierarchische Datenstruktur in allen Teilbäumen maximal aufgeteilt ist. Dieser Fall wird zur Auswertung simuliert, indem für jeden Octree sofort alle Knoten bis zur maximale Hierarchiestufe angelegt werden.

3.9.2 Randbedingungen

System zur Auswertung

Die Auswertung erfolgt auf Standardhardware. Dazu wird eine Intel Core i7-2620M CPU mit 2,7 GHz und 4 GB Speicher eingesetzt. Als Betriebssystem wird Ubuntu 10.4 LTS 32 Bit verwendet; das Framework zur Entwicklung von Fahrerassistenzsystemen ist ADTF [33] in der Version 2.6.2.

Als Sensor kommt der Ibeo LUX (siehe Kapitel 2.4.2) zum Einsatz. Es werden alle gültigen Messpunkte in die Karte integriert, die vom Sensor nicht als Bodenmessung klassifiziert wurden. Im Mittel werden 307 Messungen pro Zeitschritt integriert. Der Sensor wird mit 25 Hz betrieben; das bedeutet, dass alle 40 ms ein neuer Datensatz vorliegt.

Evaluiertes Szenario

Die ausgewertete Sequenz wurde während einer Fahrt auf einer innerstädtischen Straße in Böblingen aufgezeichnet. Die Sequenz hat eine Länge von 01:08 Minuten. Aus den Ergebnissen der Evaluierung durch eine Sequenz lässt sich nicht auf Allgemeingültigkeit schließen, es lassen sich jedoch Tendenzen im Vergleich zwischen den Konfigurationen untereinander erkennen. Insbesondere in der Hinsicht auf Parkhilfefunktionen stellt diese Sequenz mit mehreren geparkten Fahrzeugen ein Szenario dar, das in der Realität regelmäßig vorkommt. Abbildung 3.23 zeigt zur Verdeutlichung ein Standbild der Dokumentationskamera aus der Sequenz.



Abbildung 3.23 – Ausgewähltes Standbild der Dokumentationskamera aus der evaluierten Sequenz.

Abbildung 3.24 zeigt die zugehörige erstellte Belegungskarte in der Draufsicht. Die Belegungskarte ist für diese Evaluierung auf einen Radius von 60 m um das Sensorfahrzeug begrenzt, es wird also eine Strecke von einer Länge bis zu 120 m abgebildet.

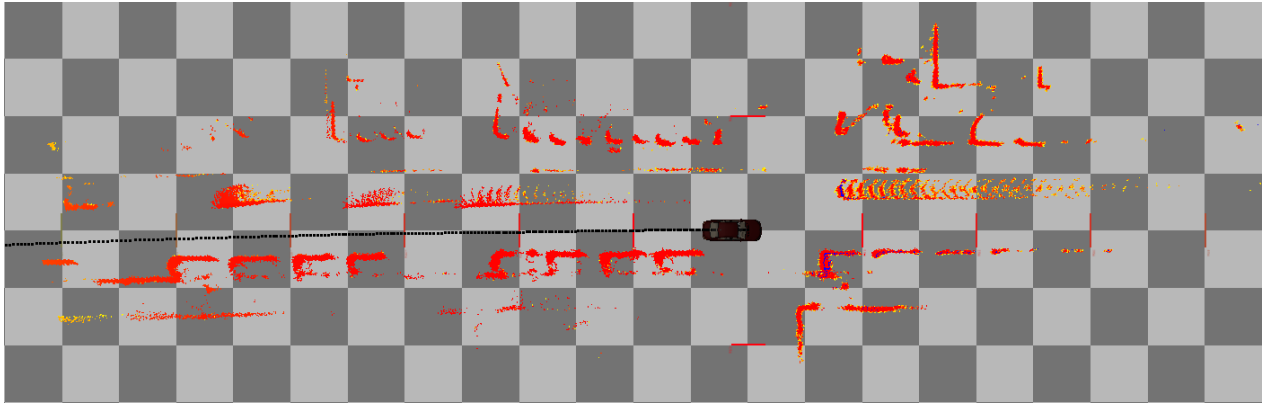


Abbildung 3.24 – Ausschnitt aus der evaluierten Sequenz in der Draufsicht. Etwa in der Mitte ist das Sensorfahrzeug zu sehen. Die Zellen der Belegungskarte sind von gelb nach rot farblich kodiert: Gelb entspricht einer Belegungswahrscheinlichkeit von knapp über 50%, rot bedeutet eine Belegungswahrscheinlichkeit von 100%. Am rechten Fahrbahnrand sind parkenden Fahrzeuge zu sehen. Auf der linken Fahrbahn sind entgegenkommende Fahrzeuge sichtbar, die ohne Sonderbehandlung Artefakte in der Karte hinterlassen. Des Weiteren sind Gebäude zu sehen. Die Zellen des quadratischen Gitters im Hintergrund haben eine Kantenlänge von 5 m.

3.9.3 Berechnungsaufwand

Der Berechnungsaufwand wird separat für zwei- und dreidimensionale Karten ermittelt. Zur Evaluierung der Effizienz der Belegungskarte wird die Berechnungsdauer unterschiedlicher Konfigurationen bei Operationen auf der Datenstruktur verglichen.

Zu diesen Operationen zählen:

- Das Integrieren aller in einem Zeitschritt angefallenen Messungen mit den modellierten Messunsicherheiten,
- Verwaltungsoperationen auf der Karte, wie das Anlegen, Verfeinern, Vergrößern und Löschen von Bäumen und
- das Anwenden eines konstanten Verfallfaktors auf die gesamte Karte.

Zweidimensionale Karten

Abbildung 3.25 zeigt die Rechenzeit der Belegungskarte mit einer konstanten maximalen Zellgröße von $(0,1\text{ m})^2$ mit unterschiedlichen Hierarchietiefen der Quadtree-Implementierung. Die Box-Whisker-Plots über die Rechenzeit zeigen, dass die höchste Effizienz durch Quadtrees mit $c_{\max} = 5$ erreicht wird, was in dem Fall einer Kantenlänge von 3,2 m entspricht. Für alle folgenden Auswertungen mit hierarchischen Konfigurationen in zwei Dimensionen wird daher die Hierarchietiefe 5 eingesetzt.

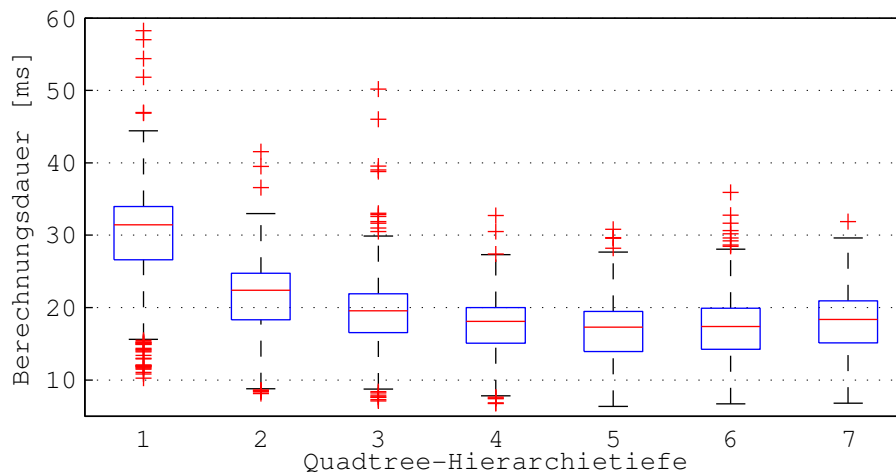


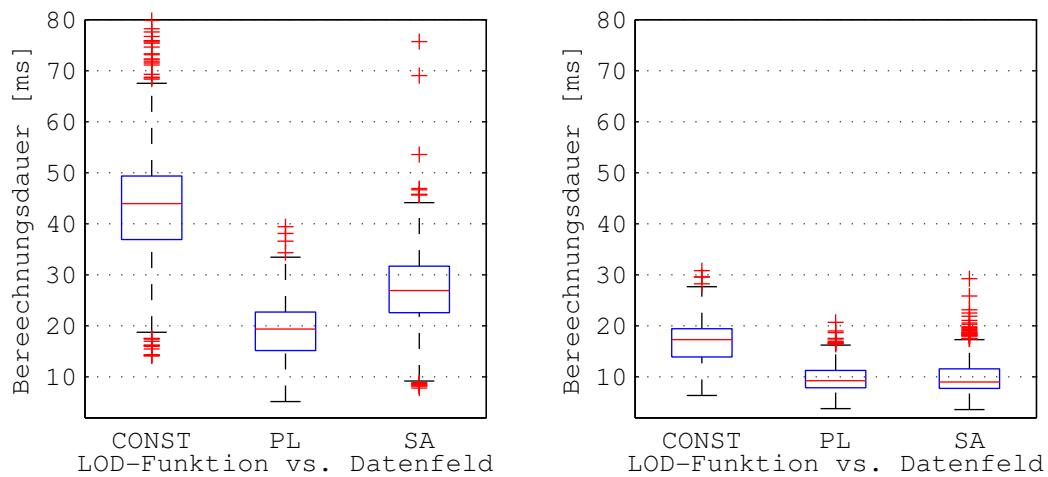
Abbildung 3.25 – Rechenzeit für Operationen auf der zweidimensionalen hierarchische Karte mit $(0,1\text{ m})^2$ großen Zellen. Auf der y-Achse sind die Rechenzeiten über die Integration der Messungen als Box-Plots aufgetragen. Es werden unterschiedliche Quadtree-Hierarchietiefen verglichen. Das Minimum zeigt sich bei Tiefe 5.

Durch die Anwendung einer Funktion zur dynamischen Detaillierungssteuerung für Parklücken, siehe Abbildung 3.26, kann der maximale Zeitaufwand im Vergleich zu einer konstanten Auflösung in den Quadtrees deutlich reduziert werden.

Die Referenz besteht in der Implementierung „SA“ (engl. Sparse Array), bei der die Quadtrees jeweils durch ein dünn besetztes zweidimensionales Datenfeld ersetzt wurden. Bei einer konstanten Auflösung „CONST“ werden alle Zellen maximal bis zur vorgegebenen Hierarchietiefe aufgelöst. Die „SA“ zeigt einen Geschwindigkeitsvorteil zur konstanten Auflösung „CONST“, die dargestellte Belegungskarte ist jedoch identisch.

„PL“ entspricht der Funktion zur Parklückenerkennung LOD_{PL} , siehe Gleichung (6.4). Die Konstante c_{PL} wird hierbei auf 1 m gesetzt. Das bedeutet, dass alle Zellen in einem Radius von einem Meter um definierte Konturpunkte in maximaler Auflösung dargestellt werden, währenddessen andere Zellen in einer reduzierten Auflösung angelegt werden. Dazu wird die hierarchische Struktur der Quadtrees genutzt. Bei einer Auflösung von 0,1 m (Abbildung 3.26b) zeigt sich der Geschwindigkeitsvorteil durch die LOD_{PL} -Funktion nicht deutlich, bei einer höheren Auflösung (Abbildung 3.26a) von 0,05 m wird jedoch im Mittel 28% der benötigten Rechenzeit eingespart.

Abbildung 3.27 zeigt die Rechenzeit für unterschiedliche Zellgrößen bei einer dynamischen Hierarchietiefe von 5 mit der LOD_{PL} -Funktion. Die zweidimensionale Karte mit 0,025 m Auflösung ist mit einem maximalen Zeitaufwand von 179,0 ms nicht echtzeitfähig. Eine Halbierung der Auflösung auf 0,05 m reduziert den maximalen Zeitaufwand auf 39,5 ms und ist damit echtzeitfähig. Eine weitere Reduktion der Auflösung führt erwartungsgemäß zu einer



(a) 2D-Karte mit $(0,05\text{ m})^2$ großen Zellen. (b) 2D-Karte mit $(0,1\text{ m})^2$ großen Zellen.

Abbildung 3.26 – Vergleich der Funktion zur Detaillierungssteuerung mit einer Implementierung, die ein dünn besetztes Datenfeld nutzt. Links wird die Test-Sequenz mit einer maximalen Auflösung von 0,05 m ausgeführt, rechts mit einer maximalen Auflösung von 0,1 m. Die Funktion „CONST“, welche die Auflösung konstant hält, benötigt erwartungsgemäß am meisten Rechenzeit. Rechts ist jeweils die Rechenzeit für eine zweidimensionale Datenfeld-Implementierung „SA“ dargestellt. Bei der Steuerung des Detaillierungsgrades durch die PL-Funktion kann Rechenzeit eingespart werden, da nur selektiv Bereiche verfeinert werden.

weiteren Reduktion der Rechenzeit.

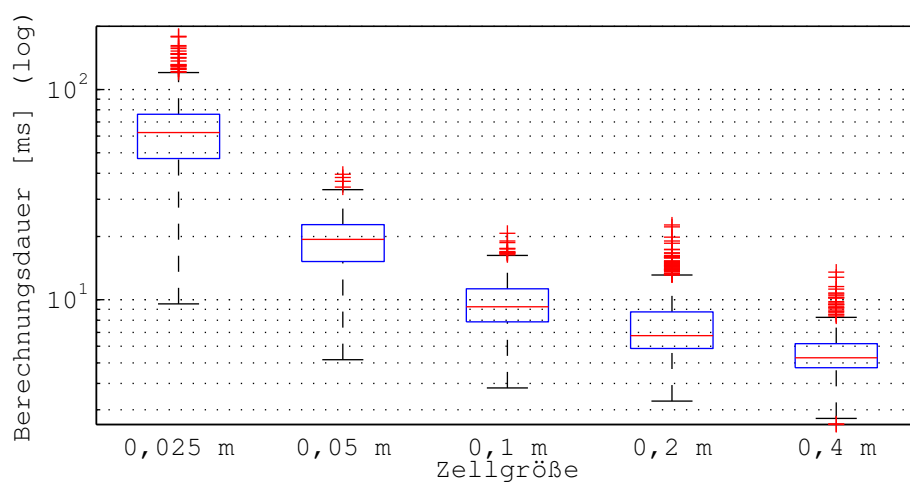


Abbildung 3.27 – Vergleich der Rechenzeit unterschiedlicher Zellgrößen der zweidimensionalen Karte bei dynamischer Hierarchietiefe in logarithmischer Skala.

Dreidimensionale Karten

Die Auswertung wird für dreidimensionale Belegungskarten analog durchgeführt. Zur Evaluierung der Effizienz der Belegungskarte wird die Berechnungsdauer mehrerer Konfigurationen bei Operationen auf der Datenstruktur verglichen. Abbildung 3.28 zeigt die Rechenzeit der dreidimensionalen Belegungskarte mit konstanter Zellgröße von $(0,2\text{m})^3$ mit unterschiedlichen Octree-Hierarchietiefen. Der Graph zeigt, dass die höchste Effizienz durch Octrees mit $c_{\max} = 4$ und einer Kantenlänge von 3,2m erreicht wird. Bei allen folgenden dreidimensionalen hierarchischen Kartenkonfigurationen wird demnach die Hierarchietiefe auf 4 festgelegt.

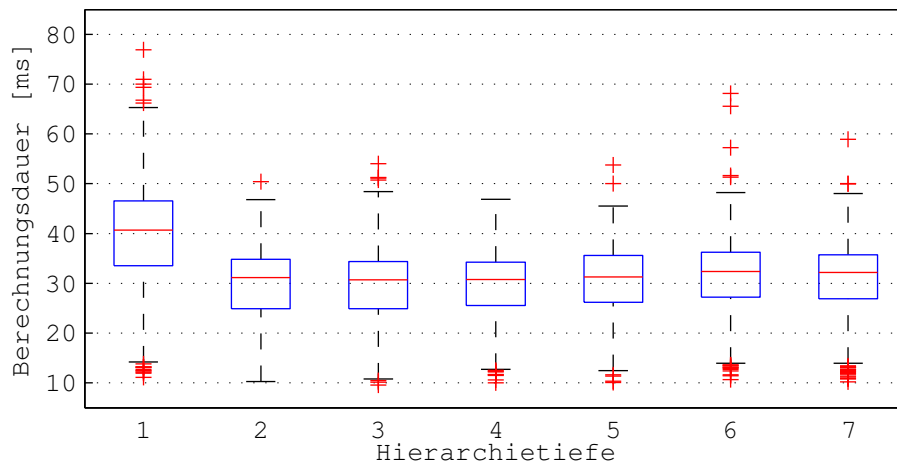


Abbildung 3.28 – Dreidimensionale Karte mit 0,2m großen Zellen. Auf der y-Achse ist die Rechenzeit über die Integration der Lux-Daten als Box-Plot aufgetragen. Es werden unterschiedliche Octree-Hierarchietiefen verglichen. Das Minimum zeigt sich bei Tiefe 4.

Durch die Anwendung der Funktion LOD_{PL} zur dynamischen Detaillierungssteuerung (siehe Abbildung 3.29) kann der maximale Zeitaufwand im Vergleich sowohl zu einer konstanten Auflösung „CONST“ als auch zur Referenzimplementierung „SA“ diesmal deutlich reduziert werden. Die Referenz besteht in der Implementierung „SA“ (engl. Sparse Array), bei der die Octrees jeweils durch ein dichtes dreidimensionales Datenfeld ersetzt wurden. Die dargestellte Belegungskarte ist somit identisch zu „CONST“. LOD_{PL} entspricht der Funktion zur Parklückenerkennung, siehe Gleichung (6.4). Hier wird die hierarchische Struktur der Octrees genutzt, indem weniger relevante Bereiche in geringerer Auflösung dargestellt werden. Bei einer Auflösung von 0,2m (Abbildung 3.29b) zeigt sich der Geschwindigkeitsvorteil durch die LOD_{PL} -Funktion bereits deutlich, im Mittel werden 50,4% der benötigten Rechenzeit eingespart. Bei einer höheren Auflösung von 0,1m (Abbildung 3.29a) wird im Mittel sogar 61,7% der benötigten Rechenzeit eingespart.

Abbildung 3.30 zeigt die Rechenzeit für unterschiedliche Zellgrößen bei einer dynamischen Hierarchietiefe von 4 in logarithmischer Skala. Die dreidimensionale Karte mit 0,05m Auflö-

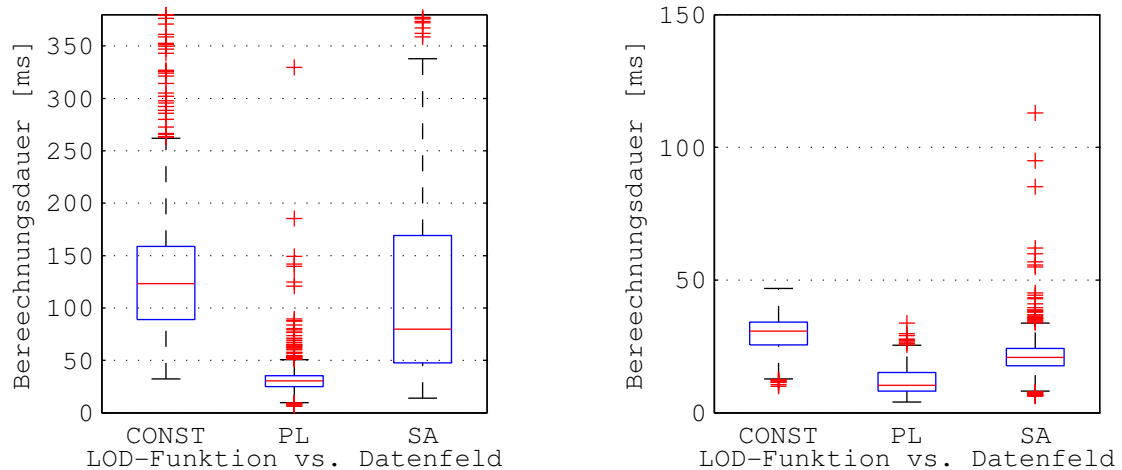
(a) 3D-Karte mit $(0,1\text{ m})^3$ großen Zellen.(b) 3D-Karte mit $(0,2\text{ m})^3$ großen Zellen.

Abbildung 3.29 – Vergleich von unterschiedlichen LOD-Funktionen für die dreidimensionale Implementierung. Links wird die Test-Sequenz mit einer maximalen Auflösung von 0,1 m ausgeführt, rechts mit einer maximalen Auflösung von 0,2 m. Die Funktion „CONST“, welche die Auflösung konstant hält, benötigt erwartungsgemäß am meisten Rechenzeit. Rechts ist jeweils die Rechenzeit für die dreidimensionale Datenfeld-Implementierung „SA“ dargestellt. Bei der Steuerung des Detaillierungsgrades durch die LOD_{PL} -Funktion wird deutlich Rechenzeit eingespart, da nur selektiv Bereiche verfeinert werden.

sung ist mit einem maximalen Zeitaufwand von 376,7 ms nicht echtzeitfähig. Eine Halbierung der Auflösung auf 0,1 m reduziert den maximalen Zeitaufwand auf 66,7 ms. Bei 0,2 m Auflösung ist der Ansatz echtzeitfähig, mit einer maximalen Rechenzeit von 33,8 ms.

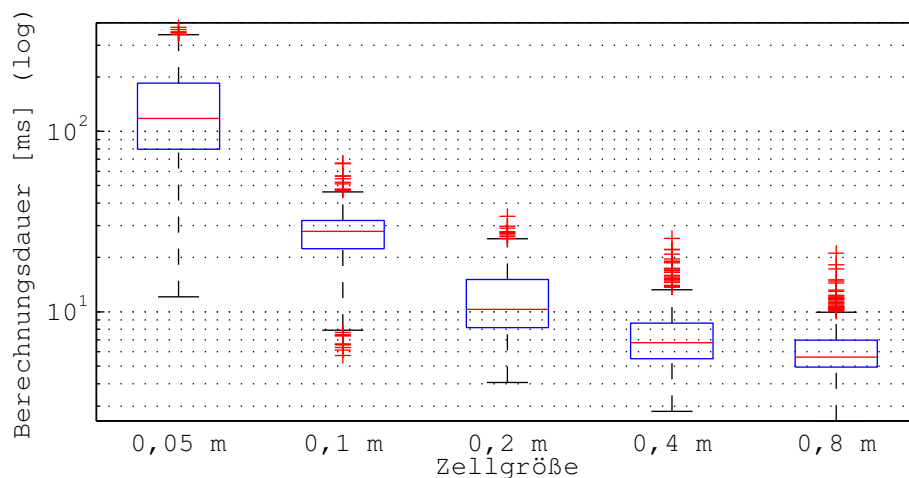


Abbildung 3.30 – Vergleich der Rechenzeit unterschiedlicher Zellgrößen der dreidimensionalen Karte bei einer dynamischen Hierarchietiefe von 4 in logarithmischer Skala.

3.9.4 Speicherbedarf

Der Speicherbedarf wird für zwei- und dreidimensionale Belegungskarten separat ermittelt. Für die Analyse des Speicherbedarfs wird das erste Fünftel der Sequenz, in der sich die Karte erst aufbaut, übersprungen.

Zweidimensionale Karten

In Abbildung 3.31 werden unterschiedliche Konfigurationen der hierarchischen Quadtree-Implementierung verglichen. Alle Konfigurationen stellen dieselbe Karte in konstant hoher Auflösung von 0,1m dar. Lediglich die übergeordnete Baumstruktur wird mit steigender Hierarchietiefe größer, was sich im Speicherbedarf zeigt. Bei der Hierarchietiefe 5 beträgt der maximale Speicherbedarf 1232,0 kB.

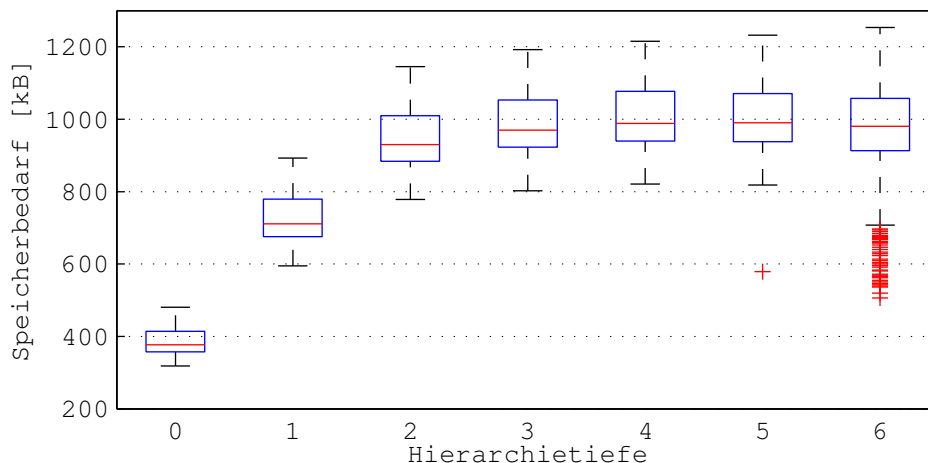


Abbildung 3.31 – Speicherbedarf der zweidimensionalen hierarchischen Karte. Die Auflösung beträgt konstant 0,1m. Mit steigender Hierarchietiefe wächst der Speicherbedarf.

Abbildung 3.32 zeigt die Auswertung für $(0,05\text{ m})^3$ große Zellen. Bei der Hierarchietiefe 5 liegt der maximale Speicherbedarf bei 3.999,4 kB und damit etwa um den Faktor 3,25 höher als bei der Variante mit 0,1 m Auflösung.

Abbildung 3.33 stellt unterschiedliche Konfigurationen gegenüber. Ausgangspunkt ist die maximale Auflösung von 0,1 m, die allen Konfigurationen gemein ist. Es werden die hierarchischen Implementierungen mit Detaillierungssteuerung betrachtet, wobei die maximale Hierarchietiefe zu 5 gewählt ist. Referenz ist der mit „CONST“ bezeichnete Plot, der die Zellgröße konstant hält und in Abbildung 3.31 der Hierarchietiefe 5 entspricht. LOD_{PL} entspricht der Funktion zur Parklückenerkennung, siehe Gleichung (6.4). Durch diese Funktion wird der Speicherbedarf noch einmal deutlich reduziert, um 77,5% auf 274 kB. Dennoch sind

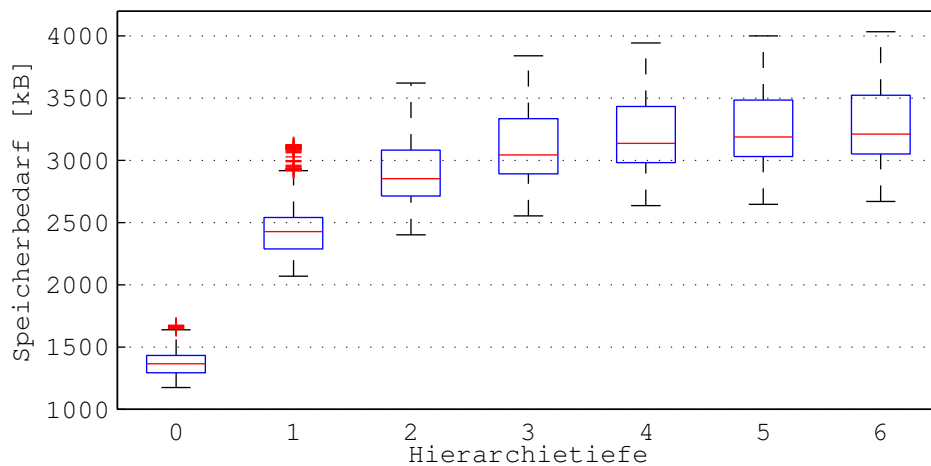


Abbildung 3.32 – Speicherbedarf der zweidimensionalen Karte bei einer Auflösung von 0,05 m. Es werden unterschiedliche Hierarchietiefen verglichen.

alle funktionsrelevanten Teile der Belegungskarte in maximaler Auflösung dargestellt. Der ungünstigste Fall „WC“ tritt dann ein, wenn für die Konfiguration „CONST“ durch eine ungünstige Verteilung der Messwerte alle Bäume maximal verfeinert werden.

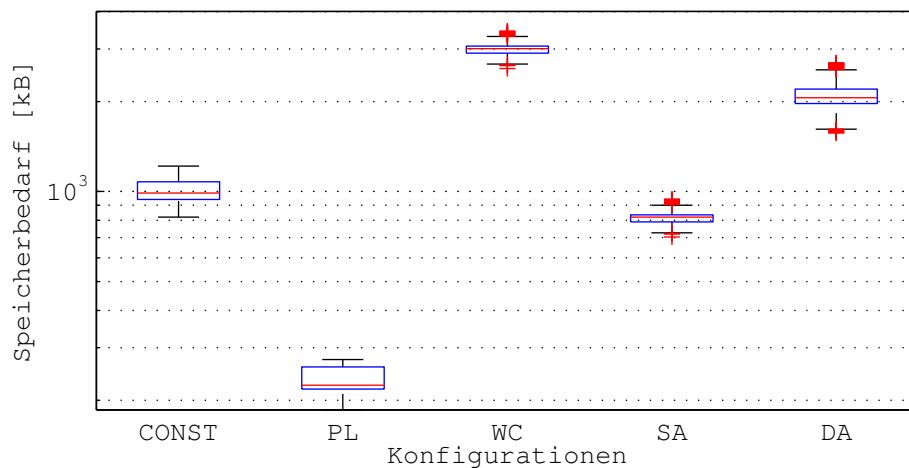


Abbildung 3.33 – Vergleich des Speicherbedarfs unterschiedlicher Konfigurationen der zweidimensionalen Karte mit 0,1 m Auflösung in logarithmischer Skala.

Des Weiteren werden zwei hypothetische Datenfeld-Konfigurationen zum Vergleich betrachtet. Die mit „SA“ bezeichnete Konfiguration repräsentiert den Bereich eines Voxels durch ein dünn besetztes Datenfeld (engl. Sparse Array). „DA“ hingegen repräsentiert die Karte in ihren gesamten Ausmaßen als dichtes Datenfeld (engl. Dense Array). Durch den zusätzlichen Speicheraufwand bei den hierarchischen Implementierungen benötigt „CONST“ 22,3% mehr Speicherplatz als „SA“ und „WC“ 21,8% mehr Speicherplatz als „DA“.

Es bleibt zu erwähnen, dass alle Konfigurationen bis auf LOD_{PL} die Karte in voller Auflösung darstellen.

Dreidimensionale Karten

In Abbildung 3.34 werden unterschiedliche Konfigurationen der hierarchischen Octree-Implementierung verglichen. Alle Konfigurationen stellen dieselbe Karte in konstant hoher Auflösung von 0,2 m dar. Lediglich die übergeordnete Baumstruktur wird mit steigender Hierarchietiefe wieder größer, was sich im Speicherbedarf zeigt. Bei der Hierarchietiefe 3 beträgt der maximale Speicherbedarf 2.095,7 kB.

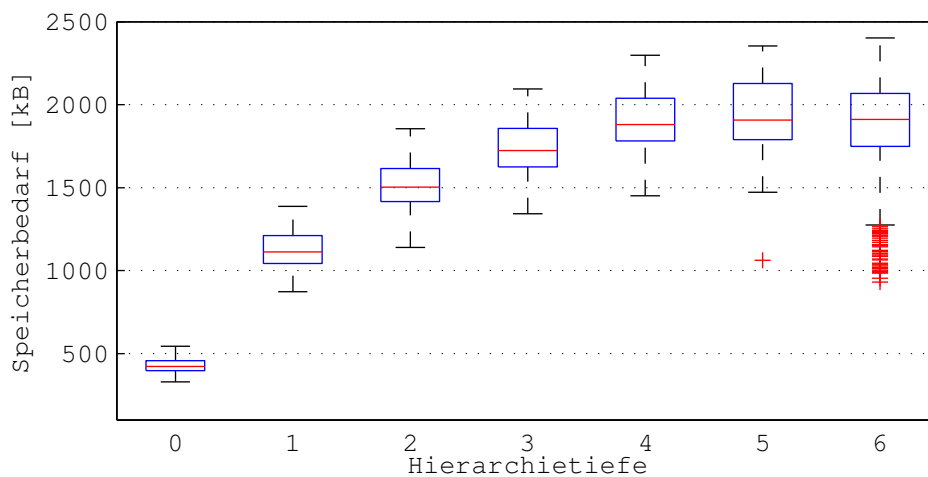


Abbildung 3.34 – Speicherbedarf der dreidimensionalen hierarchischen Karte. Die Auflösung beträgt konstant 0,2 m. Mit steigender Hierarchietiefe wächst der Speicherbedarf.

Abbildung 3.35 zeigt die Auswertung für $(0,1\text{ m})^3$ große Zellen. Bei der Hierarchietiefe 3 liegt der maximale Speicherbedarf bei 10.226,0 kB und damit etwa um den Faktor 4,9 höher als bei der Variante mit 0,2 m Auflösung.

Abbildung 3.36 stellt unterschiedliche Konfigurationen gegenüber. Ausgangspunkt ist die maximale Auflösung von 0,1 m, die allen Konfigurationen gemein ist. Es werden die hierarchischen Implementierungen mit Detaillierungssteuerung betrachtet, wobei die Hierarchietiefe konstant zu 3 gewählt ist. Referenz ist der mit „CONST“ bezeichnete Plot, der in Abbildung 3.35 Plot 3 entspricht. Durch die Funktion „PL“ lassen sich hier 81,7% gewinnen, der Speicherbedarf beträgt dann lediglich nur noch 1.823,3 kB im Vergleich zu 10.226,0 kB.

In der dreidimensionalen Variante kehrt sich der Speicheraufwand der Konfigurationen „SA“ und „DA“ um: Hier benötigen die Datenfeld-Varianten mehr Speicherplatz als die hierar-

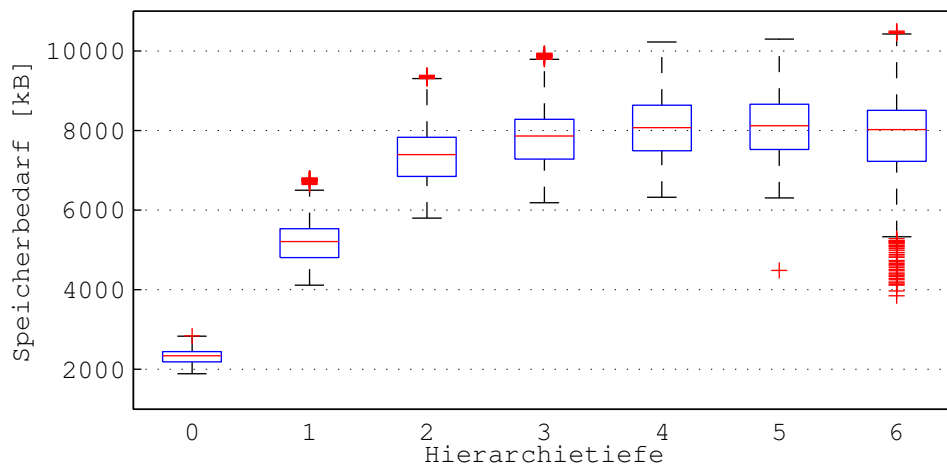


Abbildung 3.35 – Speicherbedarf der dreidimensionalen Karte bei einer Auflösung von 0,1 m. Es werden unterschiedliche Hierarchietiefen verglichen.

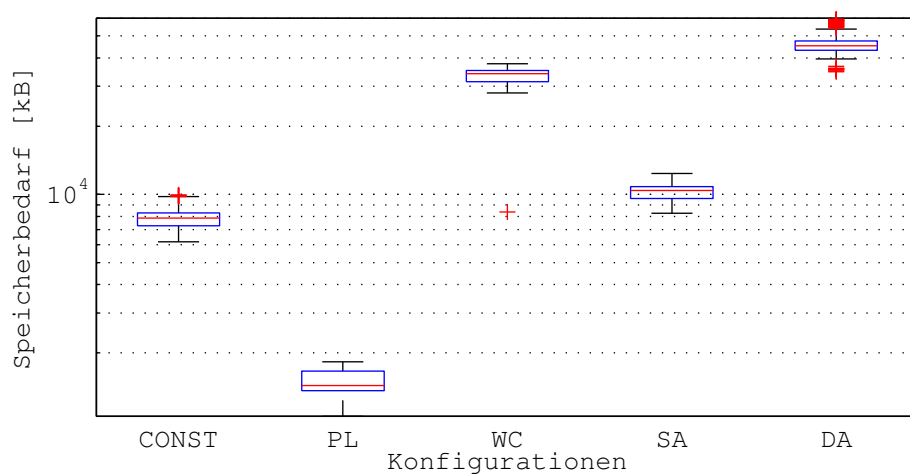


Abbildung 3.36 – Vergleich des Speicherbedarfs unterschiedlicher Konfigurationen der dreidimensionalen Karte mit Zellen von 0,1 m Kantenlänge in logarithmischer Skala.

chischen Implementierungen. „CONST“ benötigt 19,8% weniger Speicherplatz als „SA“ und „WC“ 36,5% weniger Speicherplatz als „DA“.

Bewertung des Speicherbedarfs

Zur Bewertung des Speicherbedarfs wird eine Auflösung von 0,1 m betrachtet. Gegenüber einer Implementierung mit einem dichten Datenfeld über die gesamten Kartenausmaße gewinnt man mit der zweidimensionalen hierarchischen Quadtree-Implementierung, in der dieselben

Informationen abgebildet werden, in der Testsequenz 54,9% Speicherplatz. Mit einer speziell für die Parklückendetektion ausgelegten Detaillierungssteuerung werden sogar 89,9% Speicherplatz weniger benötigt. Hier werden Bereiche, die für die Funktion weniger relevant sind, in geringerer Auflösung dargestellt.

Mit der dreidimensionalen Octree-Implementierung sind die Einsparungen noch deutlich größer: Gegenüber des dichten dreidimensionalen Datenfelds benötigt die hierarchische Variante 83,2% weniger Speicherplatz. Mit dem Fokus auf Parklücken können sogar 96,9% Speicherplatz eingespart werden.

3.10 Zusammenfassung und Ausblick

Zusammenfassung

In diesem Kapitel wurden Methoden beschrieben, um eine zwei- oder dreidimensionale hierarchische Belegungskarte aufzubauen. Der Fokus lag auf dreidimensionalen Karten, um möglichst viele Anwendungen durch eine Art der Umgebungsmodellierung abdecken zu können.

Erstmalig kann dabei der Detaillierungsgrad der Belegungskarte dynamisch von einer oder mehreren Anwendungen gesteuert werden. So können für eine Anwendung relevante Bereiche in hoher Auflösung dargestellt werden, wohingegen die Auflösung für nicht relevante Bereiche verringert wird. Dies dient einer Erhöhung der Genauigkeit der Datenbasis für eine sich anschließende Situationsanalyse bei gleichzeitig optimiertem Ressourcenverbrauch.

Zur Realisierung der Belegungskarte wurde eine dreidimensionale hierarchische Datenstruktur verwendet, welche die Vorteile der Octree-Datenstruktur mit denen eines dünn besetzten Voxelgitters kombiniert. Diese Kombination ermöglicht den effizienten Einsatz einer dreidimensionalen Karte, da homogene Bereiche zu größeren Knoten zusammengefasst werden.

Die Größe der Karte ist nicht konstant, wie häufig in anderen Ansätzen zu finden. Die Karte erweitert sich dynamisch je nach Anforderung und kann ebenso wieder reduziert werden. Damit eignet sich die Umgebungsmodellierung dazu, je nach Speicherkapazität beliebig große Karten darzustellen oder aber jeweils nur einen kleinen Bereich um das Sensorfahrzeug.

Als inverses Sensormodell wurde eine multivariate Normalverteilung eingesetzt. Dabei wurden erstmalig die Effekte, die durch Laserscanner mit Spiegeloptik entstehen, modelliert. Um Freiraummessungen zu integrieren wurde ein geometrisches Modell eingeführt, dass die Strahlkeulen von Laserscannern als Pyramiden modelliert.

In Abschnitt 3.6.2 wurde ein Verfahren beschrieben, um basierend auf der Unsicherheit des Zustandes einer Zelle die dynamische Detaillierung zu steuern. Die Unsicherheit im Belegungszustand wird mittels DST geschätzt. Das Verfahren wurde zum Patent angemeldet [152].

In Abschnitt 3.9 wurde die Performanz und der Speicherplatzbedarf der Implementierung ausgewertet. Die zweidimensionale Variante ist unter den beschriebenen Bedingungen mit einer Auflösung von 0,05 m echtzeitfähig. Für die dreidimensionale Implementierung ist eine Auflösung von 0,2 m echtzeitfähig. Gegenüber einer Implementierung mit einem dichten Datenfeld über die Kartenausmaße gewinnt man mit der zweidimensionalen hierarchischen Implementierung in der ausgewerteten Testsequenz 54,9% Speicherplatz. Mit einer speziell für die Parklückendetektion ausgelegten Detaillierungssteuerung werden sogar 89,9% Speicherplatz weniger benötigt. Für die dreidimensionale Karte sind die Einsparungen noch deutlich größer: Gegenüber dem dichten dreidimensionalen Datenfeld benötigt die hierarchische Variante 83,2% weniger Speicherplatz, mit Fokus auf Parklücken können sogar 96,9% Speicherplatz eingespart werden.

In diesem Kapitel wurden die Grundannahmen getroffen, dass die eigene Pose exakt bekannt ist und dass alle Objekte in der Welt statisch sind. In Kapitel 4 folgen Methoden zur Selbstlokalisierung auf der beschriebenen Belegungskarte und in Kapitel 5 werden Methoden zur Detektion bewegter Objekte basierend auf der Belegungskarte vorgestellt. In diesen beiden Kapiteln wird auch darauf eingegangen, wie die beiden oben genannten Grundannahmen erfüllt werden können.

In Kapitel 6 folgen praktische Experimente, um den Einsatz der Belegungskarte für Fahrerassistenz- und Sicherheitssysteme zu zeigen und die Genauigkeit festzustellen.

Ausblick

- Die Quad- und Octree-Datenstruktur wurde in dieser Arbeit klassisch über eine verzeigerte Baumstruktur beschrieben. Es bleibt zu untersuchen, welche Vorteile eine auf einer Streuwertfunktion basierende Implementierung bieten würde. Choi [14] beschreibt eine auf einer Streuwertfunktion basierende Octree-Datenstruktur. Eine Streuwertfunktion führt von einer absoluten Position direkt auf die Adresse des zugehörigen Knotens. Dieser Zugriff ist im Allgemeinen langsamer als eine auf Zeigern basierende Implementierung, da ein Zugriff in eine Streuwerttabelle nicht schneller erfolgt als eine Pointerindikation. Durch optimierte Zugriffsoperationen, die eine Traversierung der Baumstruktur unnötig machen, könnte jedoch ein Geschwindigkeitsvorteil entstehen, wie beispielsweise von Castro [13] beschrieben. Als Hash-Schlüssel eignen sich direkt die in dieser Arbeit verwendeten Lokalisierungs-codes.

- Messunsicherheiten werden in dieser Arbeit durch multivariate Normalverteilungen modelliert. Dies ist eine gute Approximation für Messungen mit geringen Winkelunsicherheiten. Bei hohen azimutalen Unsicherheiten, wie sie zum Beispiel bei einem Ultraschallsensor entstehen, kann mit anderen Modellen, wie zum Beispiel einer Gleichverteilung über den gesamten Winkelbereich, eine höhere Genauigkeit erzielt werden. Für bildgebende Radarsensoren bietet sich nach Shnidman [111] die nichtzentrale χ^2 -Verteilung als Modell für Rohdatenziele an. Um den Mehrwegeempfang und die Antennencharakteristik eines Radarsensors korrekt einfließen zu lassen, sind noch umfangreichere Sensormodelle notwendig. Zur Integration der Messdaten passiver Sensoren wie Mono- oder Stereokameras sind gegebenenfalls weitere Sensormodelle notwendig.
- Im Kontext der Fahrzeug-Fahrzeug (V2V)-Kommunikation beziehungsweise Fahrzeug-Infrastruktur (V2I)-Kommunikation sind fahrzeugübergreifende Belegungskarten denkbar. Dies würde einer umfassenderen Freiraum-Analyse für einzelne Verkehrsteilnehmer dienen, die ohne Kommunikation in einer Kreuzung nicht feststellen können, welchen Verlauf die Fahrbahn hinter einem Gebäude nimmt. Des Weiteren werden durch eine fahrzeugübergreifende Belegungskarte Freiräume frühzeitig allen Verkehrsteilnehmern bereitgestellt. Dies hat den Vorteil, dass mögliche Parklücken oder die Durchfahrtshöhe von Brücken von jedem Fahrzeug einzeln evaluiert werden können, da sie von der Fahrzeugaußenkontur abhängen. Die Fusion von Belegungskarten mehrerer Sensorfahrzeuge wirft jedoch mehrere Probleme auf: Zum einen ist eine kompakte Repräsentation der Karte zur Übertragung auf Datenbussen mit niedrigen Übertragungsraten notwendig. Hierzu bietet sich gegebenenfalls eine verlustfreie Kompression analog zu Wurm [146] an. Zum anderen resultieren aus einer Fusion mehrerer Belegungskarten mit unterschiedlichen Bezugssystemen zwangsweise Diskretisierungsfehler, die es zu beachten gilt. Weiterhin treffen Sensordaten von anderen Verkehrsteilnehmern in der Regel mit hoher Latenz ein, diese gilt es ebenso zu berücksichtigen.
- Der Belegungszustand wird in dieser Arbeit durch die Wahrscheinlichkeitstheorie geschätzt. In Abschnitt 3.6.2 wurde die DST beschrieben. Es bleibt zu untersuchen, welche Verbesserungen durch die Modellierung mit der DST erreicht werden. Die Unsicherheit im Belegungszustand kann, wie in Abschnitt 3.6.2 beschrieben, auch dazu genutzt werden, die dynamische Zellverfeinerung zu steuern. Die Vorteile einer Detaillierungssteuerung anhand der geschätzten Unsicherheit des Belegungszustandes einer Zelle bleiben mit einer Implementierung und Auswertung zu zeigen.
- Neben den Belegungszuständen belegt (OCC) und frei (EMP) könnten beispielsweise durch die DST auch mehrere Objektklassen unterschieden werden. Unter Zuhilfenahme eines kamerabasierten Klassifikators ließen sich beispielsweise die Zellklassen Fahrbahn, Fahrbahnmarkierung, Hindernis, PKW, LKW, Zweiradfahrer und Fußgänger unterscheiden.

- Viele aktive Sensoren wie Laserscanner und Radarsensoren liefern für jede Messung einen Amplitudenwert oder eine Echopulsbreite, die nähere Rückschlüsse über die reflektierte Energie erlauben. Es bleibt zu untersuchen, ob sich mit diesen Informationen die zu integrierenden Belegungswahrscheinlichkeiten gewinnbringend anpassen lassen.
- Alle Implementierungen wurden für Standard PC-Hardware vorgenommen. Die Übertragung der Software auf Steuergeräte, die im Fahrzeug eingesetzt werden können, steht noch aus. Durch die Umstrukturierung der Quad- oder Octree-Datenstruktur während der Kartenerstellung und -auswertung sind dynamische Speicherallokationen notwendig, die auf einem Steuergerät problematisch sein können. Die Datenstrukturen könnten auf den herausforderndsten Fall ausgelegt werden, wodurch jedoch der Gewinn durch die Reduktion der Auflösung verschwindet. Hierzu muss noch ein Konzept ausgearbeitet werden. Durch den hierarchischen Aufbau der Datenstruktur ist jedoch eine Parallelisierung der Operationen auf der Datenstruktur möglich, so dass bei einer Adaption für ein FPGA (engl. Field Programmable Gate Array) Geschwindigkeitsvorteile durch parallele Verarbeitungsmodul genutzt werden können.
- Die Parallelisierbarkeit der Operationen auf der Datenstruktur lässt sich auch in einem anderen Kontext nutzen: Ein Grafikprozessor GPU (engl. Graphics Processing Unit) kann über eine API oftmals direkt programmiert werden, wie das Beispiel von CUDA (engl. Compute Unified Device Architecture) zeigt. Da ein GPU über sehr schnelle parallele Recheneinheiten verfügt, würde eine Implementierung hierfür deutlich feinere Zellen wie bisher ermöglichen. Durch die vorgestellten Algorithmen sind lokal Zellauflösungen im Zentimeterbereich möglich. Durch eine GPU-Implementierung ließen sich möglicherweise Zellen im Millimeterbereich darstellen. Dies wäre insbesondere für Forschungs- und Evaluierungszwecke von Interesse.

4

Simultane Selbstlokalisierung und Kartierung

Für eine komplexe Situationsanalyse ist neben der Kenntnis über andere Verkehrsteilnehmer sowohl der dynamische Zustand des eigenen Fahrzeuges als auch dessen Pose relativ zu statischen Objekten der Umgebung erforderlich. Die Pose beinhaltet die Position und die Orientierung des Fahrzeuges im Raum. In diesem Kapitel wird die Schätzung des Zustandes des eigenen Fahrzeuges in Kombination mit der Erstellung einer Karte mit statischen Objekten betrachtet. In der Literatur wird dieses Verfahren als SLAM bezeichnet (engl. Simultaneous Localization and Mapping).

4.1 Motivation

Die Kenntnis über die Pose des eigenen Fahrzeuges und dessen Dynamik ist für moderne Fahrerassistenz- und Sicherheitssysteme essentiell. Navigationsgeräte sollen beispielsweise in Tunneln nicht ausfallen, sondern die eigene Position weiter aktualisieren. Radarsensoren als weiteres Beispiel benötigen Eigenbewegungsdaten für die Verfolgung von Verkehrsteilnehmern.

Eine grundlegende Posen- und Bewegungsschätzung kann durch Odometriedaten erfolgen. Aus der Anzahl der Umdrehungen der Räder und deren Umfang kann man auf Positions- und Lageänderungen schließen. Es wird jedoch keinerlei Höheninformation erfasst, so dass ei-

ne Positionierung ausschließlich in der zweidimensionalen Ebene erfolgt. Wesentlich genauere Daten können durch ein Inertialmesssystem (IMU, engl. Inertial Measurement Unit) ermittelt werden, das Gier-, Nick- und Wankrate sowie die Beschleunigungen entlang den drei Sensorkoordinatenachsen misst. Der dreidimensionale Fahrbahnverlauf ist relevant für die Fahrt auf Rampen oder in Steilkurven.

Doch selbst bei Inertialmesssystemen akkumulieren sich Messfehler über die Zeit auf, so dass Korrekturen durch absolute Stützmessungen notwendig sind, die beispielsweise durch GPS bestimmt werden können. Die Verfügbarkeit von GPS ist unter Umständen jedoch eingeschränkt, wie in Häuserschluchten oder Tunneln, so dass dies keine allgemeingültige Lösung darstellt. Aufgrund der Akkumulierung von Messfehlern entsteht ohne absolute Stützmessungen folgendes Problem: Kehrt ein Fahrzeug nach einer Fahrt wieder an den Ausgangspunkt zurück, so entspricht die Schätzung der Pose unter Umständen nicht wieder der Ausgangspose, wie erwartet. Über die Erkennung und Wiedererkennung von Merkmalen in der Umgebung lässt sich dieses „Loop Closing“-Problem lösen. Dieses Verfahren wird *Selbstlokalisierung* genannt.

Zur Selbstlokalisierung werden Sensoren eingesetzt, welche die Umgebung geeignet erfassen. Durch Änderungen in der wahrgenommenen Umgebung während der Fahrt lassen sich Posenänderungen des Fahrzeuges schätzen. Dies kann durch die Registrierung zweier zeitlich aufeinanderfolgender Messungen erfolgen oder durch die Registrierung der aktuellen Messung auf eine Repräsentation vorangegangener Sensormessungen, die beispielsweise durch eine Belegungskarte dargestellt wird. Mit einer Posenschätzung basierend auf einer Belegungskarte ist zudem eine relative Positionierung zu statischen Objekten möglich. Die Zuordnung zu einer Fahrspur, die Bestimmung des Abstands zum Fahrbahnrand oder die Bestimmung des Abstands und die Orientierung zu einer Parklücke sind mögliche Anwendungen.

Weiterhin ist die Kenntnis über den Zustand des eigenen Fahrzeuges Grundvoraussetzung für den Aufbau einer Belegungskarte. Für die Erstellung der in Kapitel 3 eingeführten Belegungskarte wird vorausgesetzt, dass die Pose des eigenen Fahrzeuges bekannt ist. Erst die Kenntnis über die exakte Position und Orientierung zum Zeitpunkt der Messung ermöglicht die Erstellung einer präzisen Karte. In der Praxis stehen jedoch weder eine genaue Karte noch genaue Poseninformationen zur Verfügung. Somit sind beide Aufgaben simultan zu lösen.

Abbildung 4.1 veranschaulicht das SLAM-Problem. Die zur Kartierung notwendige Pose wird durch die Lokalisierung ermittelt, die wiederum eine genaue statische Karte benötigt. Das Problem wird vereinfacht, wenn neben dem Distanzprofil eines Umgebungssensors weitere Informationen zur Verfügung stehen: Die Odometrie kann zur Initialisierung der Selbstlokalisierung dienen und GPS kann absolute Stützmessungen beitragen.

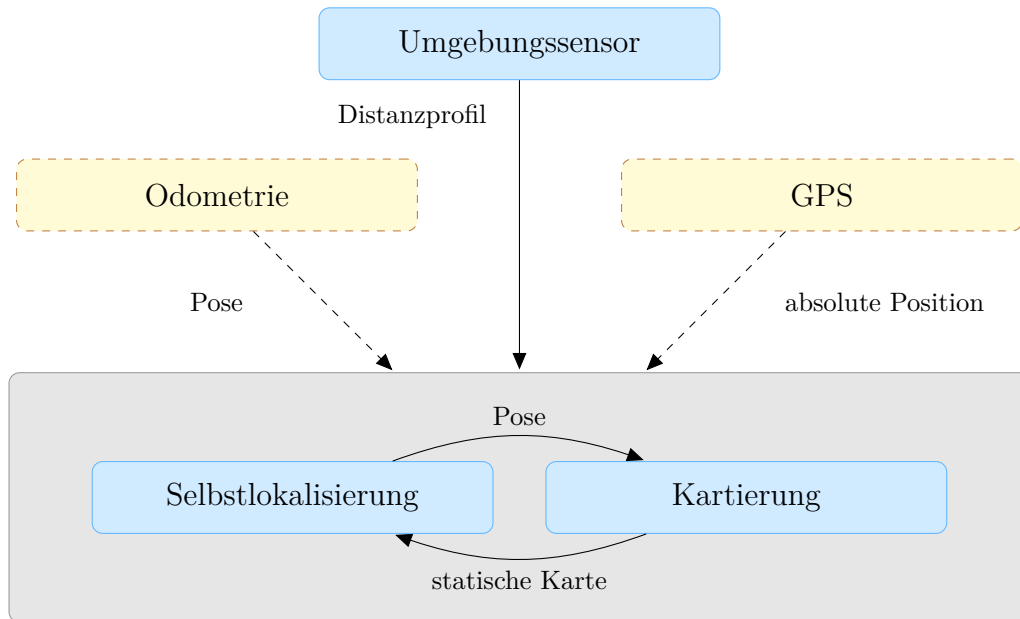


Abbildung 4.1 – Architektur: Simultane Selbstlokalisierung und Kartierung.

Zielsetzung für die Selbstlokalisierung

Ein GPS ist im automobilen Kontext in der Regel verfügbar, um eine grobe absolute Lokalisierung durchzuführen. Es wird deshalb ein Verfahren gesucht, um diese Posenschätzung lokal zu verbessern, da die Genauigkeit von GPS für zukünftige Fahrerassistenzsysteme in der Regel nicht ausreicht und die Odometrie dazu tendiert, durch Kalibrierfehler und Reifenschlupf ungenaue Werte zu liefern. Dennoch soll es möglich sein, die Messungen von GPS oder der Odometrie einfließen zu lassen.

Für globale Navigationsaufgaben ist das „Loop-Closing“-Problem zu lösen. Navigationsaufgaben werden in dieser Arbeit jedoch nicht betrachtet, das Loop-Closing-Problem wird deshalb für die Selbstlokalisierung außer Acht gelassen. Für eine lokale Kartierung und Posenschätzung muss dieses Problem auch nicht gelöst werden, da hier die absoluten Fehler vernachlässigt werden können. Dies wird anhand der folgenden Anwendungen deutlich: Einen PKW in eine Parklücke zu führen, eine Reaktion auf potentiell kollidierende Verkehrsteilnehmer in der näheren Umgebung (PreCrash), die Querführung auf der Autobahn (Spurwechselassistent) oder die Reaktion auf Fußgänger, die möglicherweise die Straße betreten - diese Funktionen greifen lediglich auf eine lokale Karte und die darin befindlichen bewegten Objekte zurück.

Ebenso außer Acht gelassen werden die Besonderheiten der Selbstlokalisierung durch die Analyse aufeinanderfolgender Kamerabilder, auch als visuelle Odometrie bekannt. Die Selbstlokalisierung soll stattdessen sowohl mit hochaufgelösten sphärischen Distanzprofilen als auch mit wenigen Messpunkten aus einer Ebene arbeiten. Weiterhin werden Möglichkeiten zur dreidimensionalen Selbstlokalisierung auf Basis einer dreidimensionalen Karte untersucht.

Kapitelübersicht

In Abschnitt 4.2 dieses Kapitels wird der Stand der Technik für SLAM-Ansätze beschrieben. Nach einer Einführung in die Eigenbewegungsschätzung durch ein erweitertes Kalman-Filter in Abschnitt 4.3 wird die Integration der Messungen der Odometrie in Abschnitt 4.4 vorgestellt. Als Lokalisierungsverfahren werden in Abschnitt 4.5 rasterbasierte Verfahren beschrieben. Die Pose des Fahrzeuges wird dabei durch die Registrierung von Sensorrohdaten auf die statische Belegungskarte geschätzt. In Abschnitt 4.6 wird eine Evaluierung der vorgestellten Ansätze durchgeführt, indem die geschätzten Fahrzeugpositionen mit der Referenzsensorik verglichen werden. Anschließend wird in Abschnitt 4.7 ein Verfahren zur absoluten Lokalisierung durch einen dreidimensionalen Kartenabgleich skizziert. Das Kapitel endet mit einer Zusammenfassung und einem Ausblick in Abschnitt 4.8.

4.2 Stand der Technik

Thrun gibt in [127] einen historischen Rückblick über die Kartierung in der Robotik. In [130] werden mehrere Verfahren vertieft und detailliert beschrieben. Seit den 1990er Jahren dominieren probabilistische Methoden zur Kartierung.

4.2.1 Kalman-SLAM

Smith beschreibt 1990 [114] ein Lokalisierungsverfahren namens „stochastische Karte“. Beziehungen zwischen Objekten werden probabilistisch modelliert. Die Unsicherheiten vergrößern sich bei Bewegungen des Roboters und verringern sich entsprechend bei neuen Beobachtungen. Als Filterverfahren wird das Kalman-Filter eingesetzt, das die Verbundwahrscheinlichkeit über die Position des Roboters und die Position von K Landmarken im Zustandsvektor schätzt. Die Rechenzeit steigt quadratisch mit der Anzahl der Landmarken, die typischerweise zwischen 10 und 100 liegt. Das Verfahren ist in der Literatur auch unter dem Namen Kalman-SLAM bekannt.

Durch die Weiterentwicklung von Kalman-SLAM zu FastSLAM von Montemerlo [84] und Thrun [126] lässt sich die Rechenzeit auf $O(\log K)$ reduzieren. Der Vorteil der Kalman-SLAM Algorithmen ist, dass sie streng konvergieren. Der Hauptnachteil ist jedoch, dass das Korrespondenz-Problem nicht gelöst ist und gleiche Landmarken nicht wiedererkannt werden.

Himmelsbach [58] stellt ein neues SLAM-Verfahren namens Topological FastSLAM vor. Dieses ist eine Erweiterung von FastSLAM um lokale Unterkarten, um das Problem des Loop-Closing lösen. Die Auswertung erfolgt anhand simulierter Daten.

Selbstlokalisierung über Punktkorrespondenzen

Gutmann [56] vergleicht unterschiedliche Verfahren zur Registrierung von Laserscannermessungen (engl. Scan Matching) für Roboter im Innenraum. Er setzt eine Kombination aus zwei Verfahren ein: Beim Cox-Verfahren werden aus Messdaten Geraden extrahiert, welche aufeinander registriert werden. Dieses Verfahren funktioniert nur in Umgebungen, die durch ebene Wände begrenzt sind. Beim ICP (engl. Iterative Closest Point)-Algorithmus hingegen werden direkt die Abstände zwischen korrespondierenden Punkten minimiert. Dieser Algorithmus arbeitet auch in Umgebungen ohne ebene Wände, zeigt jedoch deutliche Nachteile bei langen Korridoren. Gutmann setzt, je nach vorgefundener Situation, den Algorithmus ein, der die besseren Ergebnisse liefert.

Der ICP-Algorithmus [7] dient dazu, Punktwolken zu registrieren. Dabei wird die Summe der quadratischen Abstände zwischen einzelnen Punkten minimiert. Im ursprünglichen Algorithmus wird davon ausgegangen, dass jeder einzelne Punkt in einer Menge jeweils eine exakte Korrespondenz in der anderen Menge hat. Dies ist bei einer Fahrt mit einem Laserscanner jedoch nicht der Fall. Abbildung 4.2 schildert dies an einem Extremfall: In einem Tunnel ohne Merkmale außer den Wänden ändert sich die Umgebung für einen Laserscanner während der Geradeausfahrt nicht. Somit ist eine Selbstlokalisierung nicht ohne weiteres möglich. Der ICP-Algorithmus würde eine exakte Übereinstimmung zwischen den Punktmengen ermitteln, aber keine Positionsänderung.

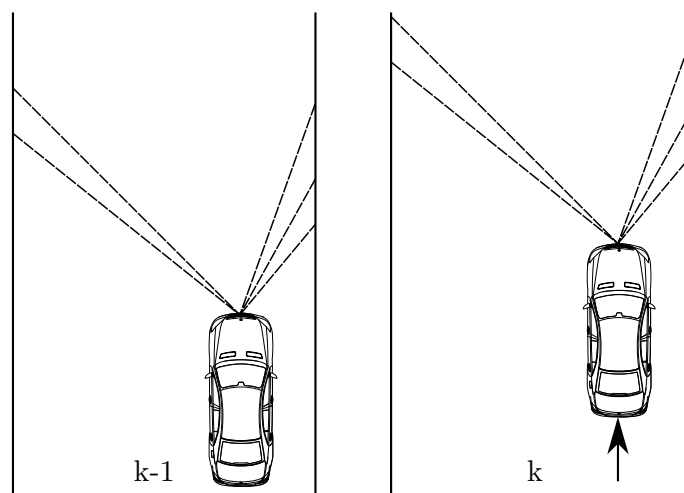


Abbildung 4.2 – Fahrt durch einen Tunnel. Es sind keine Merkmale zur longitudinalen Lokalisierung vorhanden. Die Umgebung sieht an der Position zum Zeitpunkt $k - 1$ genauso aus wie an der Position zum Zeitpunkt k .

Mit solchen schwachen Merkmalen ist es generell nicht möglich, eine vollständige Selbstlokalisierung durchzuführen. Es lassen sich Informationen über die laterale Translation und das Gieren ableiten. Über die longitudinale Translation jedoch lässt sich nichts aussagen. Dieses Nicht-Wissen muss bei der Selbstlokalisierung berücksichtigt werden, um an dieser Stelle

beispielsweise die genaueren Messungen von der Odometrie einfließen zu lassen. Wang [136] führt für den ICP-Algorithmus hierzu vielfach Registrierungen mit unterschiedlichen Initialisierungen durch.

Die Lösung des SLAM-Problems in drei Dimensionen erfordert die Bestimmung von sechs Parametern: Die drei Raumkoordinaten sowie den Roll-, Nick- und den Gierwinkel. Deshalb wird das Problem auch als 6D-SLAM bezeichnet [93]. Zur Lösung des 6D-SLAM-Problems werden von Nüchter [94] einzelne 3D-Scans mit dem ICP-Algorithmus [3] registriert. Als Laserscanner kommt dabei der AIS 3D Laserscanner [121] zum Einsatz. Hierbei handelt es sich um einen einzeiligen SICK LMS 200, der durch einen seitlich angebrachten Servo Nickbewegungen durchführen kann und somit in der Lage ist 3D-Scans zu erstellen. Für einen Scan mit $181 \cdot 256$ Datenpunkten benötigt dieser 3,4 Sekunden. Eine initiale Schätzung für die Transformation wird mit Hilfe von Octrees berechnet: Hier werden überlagerte 3D-Punktmengen und die daraus resultierenden Octrees registriert. Alleine dieser Schritt benötigt für 20.000 – 300.000 Punkte bis zu 2 Sekunden Rechenzeit [93]. Zur Steigerung der Geschwindigkeit wird die Anzahl der Messpunkte reduziert. Benachbarte Punkte eines 2D-Scans werden durch deren Mittelpunkt ersetzt; anschließend wird nur jede dritte Ebene verwendet um den 3D-Scan zu bilden. Um benachbarte Punkte schneller zu finden, erfolgt die Suche durch einen approximierenden kd-Baum. Eine Einführung in kd-Bäume findet sich in Abschnitt 3.5.2. Im Vergleich zwischen einer zweidimensionalen und einer dreidimensionalen Variante hat sich herausgestellt, dass die dreidimensionale Variante im Vorteil ist, da hier wesentlich mehr Merkmale zur genauen Registrierung vorhanden sind [93]. Ein interessanter Einsatzzweck wird in [94] verfolgt: Die Kartierung einer verlassenen Mine, von der keine genauen 3D-Karten existieren, mit einem autonomen Roboter.

In Sequenzen, in denen gute Merkmale und dichte Daten vorhanden sind, ist es möglich mit einem auf Punktkorrespondenzen basierenden Algorithmus sehr gute Ergebnisse zu erhalten, wie beispielsweise von Moosmann [86] gezeigt. Hier wird ein Velodyne HDL-64E verwendet, der sehr dichte Punktwolken liefert. Es wird eine Sequenz mit sehr starken Merkmalen in einer Stadt untersucht.

Rasterbasierte Selbstlokalisierung

C.-C. Wang [136] stellt einen Ansatz vor, um Selbstlokalisierung und die Detektion bewegter Objekte integriert zu lösen. Dazu nutzt er eine 2D-Belegungskarte für innerstädtische Szenarien. Das Loop-Closing Problem löst er über eine merkmalsbasierte Darstellung, wogegen die Selbstlokalisierung über kurze Distanzen über Korrelationsmethoden auf 2D-Kartenabschnitten gelöst wird.

Weiss benutzt Belegungskarten zur Selbstlokalisierung im Kontext Fahrerassistenzsysteme [141, 139]. Zur Lokalisierung wird zunächst die Eigenbewegung verwendet, die durch Drehrate, Radgeschwindigkeiten und Lenkwinkel geschätzt wird. Die Lokalisierung wird durch einen gitterbasierten Ansatz gestützt. Hier wird der Freiheitsgrad geschätzt, der am meisten zu einer verbesserten Schätzung beiträgt: Der Gierwinkel. Um die Unsicherheit der Fahrzeugpose in Betracht zu ziehen, wird diese mit den Messunsicherheiten des Sensors überlagert. Als Sensor wird ein Laserscanner Ibeo Alasca XT (siehe Abschnitt 2.4.2) verwendet. Um auch beim Schleudern des Fahrzeuges korrekte Ergebnisse zu erhalten, erfolgt eine Lokalisierung mit einem hybriden SLAM-Ansatz [138, 139, 140]. Hierzu wird bei der Detektion von Schleudern des eigenen Fahrzeuges durch die ESP-Sensorik auf einen aufwändigeren Algorithmus umgeschaltet: Ein merkmalsbasierter Ansatz mit Punktlandmarken namens TrAss (von engl. Triangle Association). Aus der Belegungskarte werden Punktmerkmale extrahiert, von denen jeweils drei zu einem Dreieck zusammengefasst werden. Diese können durch Lagebeziehungen assoziiert werden, was die Eigenlokalisierung verbessert. Weiss erreicht eine Genauigkeit von 1 m in der Positionsschätzung mit Landmarkennavigation, im Vergleich zum Fehler von 20 m nur mit der Odometrie [139].

Vu [133] schlägt ein Verfahren namens „Voting SLAM“ vor, bei dem eine Gütefunktion maximiert wird, welche die Überdeckung des aktuellen Scans mit der bestehenden Karte beschreibt. Dazu wird die eigene Pose bis zu 400-fach zufällig variiert und das Ergebnis mit der besten Güte ausgewählt. Konzeptuell findet der Algorithmus selten exakt das Maximum, hat aber dafür den Vorteil, dass er nicht in lokalen Maxima stecken bleibt.

Levinson [79] kalibriert den Velodyne-Laserscanner, so dass alle 64 Ebenen eine homogene Intensität liefern. Für einzelne Zellen einer Karte wird neben der mittleren Intensität aller angefallenen Messungen auch deren Varianz gespeichert. Statische Objekte kennzeichnen sich durch eine geringe Varianz, wogegen die Fahrbahn typischerweise eine hohe Varianz aufweist. In einer Offline-Verarbeitung mit mehreren Messfahrten erhält er einen mittleren Positionsfehler von 0,09 m.

Selbstlokalisierung über Monte-Carlo-Methoden

Eliazar [36] schätzt mit einem Verfahren namens DP (engl. Distributed Particle)-SLAM die Karte und die Roboterpose simultan. Dazu wird ein Partikelfilter verwendet. Jedes Partikel der multivariaten Verteilung drückt sowohl die Roboterpose als auch die Beobachtungen aus. Um die resultierenden Karten nicht duplizieren zu müssen, werden in jeder Zelle alle Beobachtungen in einem balancierten Baum abgelegt. Das Verfahren findet mit einem Laserscanner im Innenraum eines Gebäudes Anwendung.

Fairfield et al. [42] benutzen ein Partikelfilter zur Lösung des SLAM-Problems mit Octree-Evidence-Grids. Ziel ist die Erkundung von wassergefüllten Kalksteinlöchern (Cenoten). Als Sensoren werden Ultraschallsensoren eingesetzt.

Frei verfügbare Implementierungen zur Lösung des SLAM-Problems werden durch das Open-SLAM.org-Projekt [115] zur Verfügung gestellt.

Weitere Lokalisierungsmethoden im Kontext

Fahrerassistenzsysteme

Über die Registrierung einer Belegungskarte mit einer digitalen Karte, genannt Kartenabgleich (engl. Map Matching), lässt sich die Schätzung der Fahrzeugpose durch absolute Stütz-messungen verbessern.

Konrad [75] und Szczot [122] beschreiben eine Fahrbahnverlaufsschätzung basierend auf der Registrierung von einer lokalen Belegungskarte zu einer digitaler Karte. Dazu wird ein Laserscanner eingesetzt. Ziel ist, basierend auf einer initialen GPS-Messung, eine exakte Positionierung zu erlauben und mit den vorhandenen digitalen Kartendaten eine erweiterte Fahrbahnvorausschau zu erreichen. Dies dient unter anderem dazu, die Relevanz eines Objektes zu bestimmen. Wenn sich ein Fußgänger auf der Fahrbahn befindet, muss anders auf ihn reagiert werden, als wenn er sich neben der Fahrbahn befindet. Schüle [109] untersucht den gleichen Sachverhalt, basierend auf den Daten eines bildgebenden Radarsensors.

In der Forschungsinitiative Ko-FAS (Kooperative Fahrerassistenzsysteme) werden im Teilprojekt Ko-TAG kooperative Transponder untersucht. Schwarz stellt diese in [108] vor. Die Transponder können dazu genutzt werden, die Fahrzeugposition zu bestimmen. Mit Abstandsmessungen zu drei Transpondern kann die eigene Position trianguliert werden. Weiterhin können die Transponder ihre absolute Position übermitteln, so dass eine absolute Positionierung möglich ist.

4.3 Schätzung des Fahrzeugzustandes

Informationen über den Fahrzeugzustand können häufig nicht exakt gemessen werden. Eine Fahrt durch einen Tunnel verhindert beispielsweise GPS-Messungen. Odometriedaten werden durch Schleuderbewegungen sehr ungenau. Das Ziel einer genauen Eigenbewegungsschätzung ist es daher, möglichst alle verfügbaren Informationen über die eigene Pose und deren Änderung über die Zeit einfließen zu lassen. Als eine Informationsquelle dient die Fahrzeugodometrie. Diese wird vom ESP (Elektronisches Stabilitätsprogramm)-Steuergerät ermittelt und

beinhaltet beispielsweise die Geschwindigkeit v und die Gierrate $\dot{\Psi}$ des Fahrzeugs. Weiterhin können durch GPS ermittelte Positionsinformationen einfließen sowie Posenänderungen, die durch die Umgebungssensorik erfasst werden können. Das erweiterte Kalman-Filter eignet sich zur Fusion dieser mit Unsicherheiten behafteten Messungen. Zur Schätzung des Zustands des Fahrzeugs werden die zu schätzenden Parameter in einem Zustandsvektor zusammengefasst.

4.3.1 Zustandsraum

Um im dreidimensionalen kartesischen Raum die Position eines Objektes anzugeben, sind drei Koordinaten notwendig: x, y und z . Um zusätzlich die Orientierung des Objektes im Raum anzugeben, sind drei weitere Parameter notwendig. Hier können unter anderem die drei Euler-Winkel genutzt werden: Die Rotation um die x-Achse Φ (Rollen), die Rotation um die y-Achse Θ (Nicken) und die Rotation um die z-Achse Ψ (Gieren). Die Position und Orientierung zusammengenommen wird als Pose bezeichnet und wie in Gleichung (3.3) dargestellt als \mathbf{x}_F geschrieben.

Um die Pose des Fahrzeugs präzisieren zu können, sind weiterhin Informationen über die Dynamik des Fahrzeugs notwendig. Die Änderung der Orientierung kann durch die Winkelgeschwindigkeiten $\dot{\Phi}, \dot{\Theta}$ und $\dot{\Psi}$ beschrieben werden. Die Geschwindigkeit des Objektes in Fahrtrichtung wird mit v bezeichnet. Von Beschleunigungssensoren kann gegebenenfalls die Beschleunigung eines Objektes gemessen werden, diese erhöht die Genauigkeit einer Prädiktion. Die Beschleunigung wird mit a bezeichnet. Geschwindigkeiten oder Beschleunigungen, welche auf die y-Achse oder auf die z-Achse wirken, werden nicht modelliert, sondern fließen als Rauschen in die Zustandsschätzung ein.

Abbildung 4.3 zeigt den Zustand des Fahrzeuges in Bezug auf das ENU-Koordinatensystem, wie in Abschnitt 3.3.1 beschrieben. Der Zustandsvektor \mathbf{x} , der den Fahrzeugzustand beschreibt, wird aus allen Komponenten zusammengesetzt:

$$\mathbf{x} = \left[x \ y \ z \ \Phi \ \Theta \ \Psi \ \dot{\Phi} \ \dot{\Theta} \ \dot{\Psi} \ v \ a \right]^T \quad (4.1)$$

4.3.2 Prozessmodell

Die Positionsänderung des Fahrzeuges zwischen zwei Zeitpunkten wird über die drei Translationskomponenten $\Delta x, \Delta y$ und Δz angegeben. Zur Bestimmung dieser Komponenten wird ein Fahrzeugmodell ähnlich wie in Mählich [91] verwendet. Das Modell wurde lediglich auf die dritte Raumdimension erweitert.

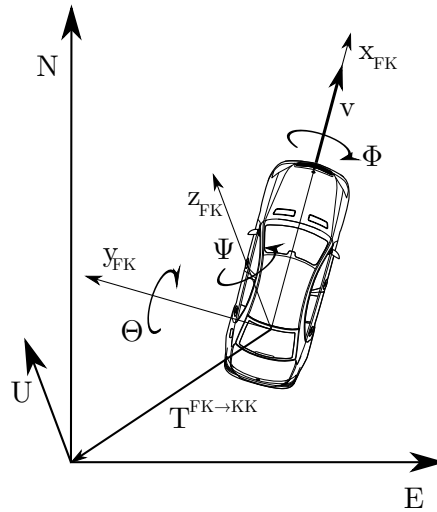


Abbildung 4.3 – Pose des Fahrzeuges im Raum.

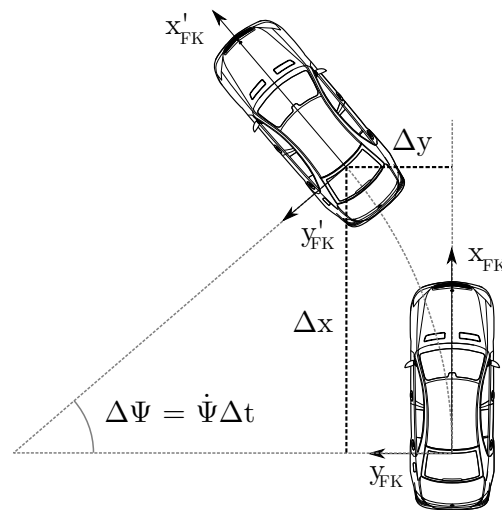


Abbildung 4.4 – Koordinierte Fahrt auf einer Kreisbahn.

In Abbildung 4.4 ist die Bewegung eines Fahrzeuges auf einer Kreisbahn skizziert. Aus dieser Bewegung ergeben sich für die Translation folgende Komponenten:

$$\Delta x = v \sin(\dot{\Psi} \Delta t) \dot{\Psi}^{-1} \quad (4.2)$$

$$\Delta y = v(1 - \cos(\dot{\Psi} \Delta t)) \dot{\Psi}^{-1} \quad (4.3)$$

In Abbildung 4.5 ist die modellierte Nickbewegung zu sehen. Hieraus ergeben sich für die Translation folgende weitere Komponenten:

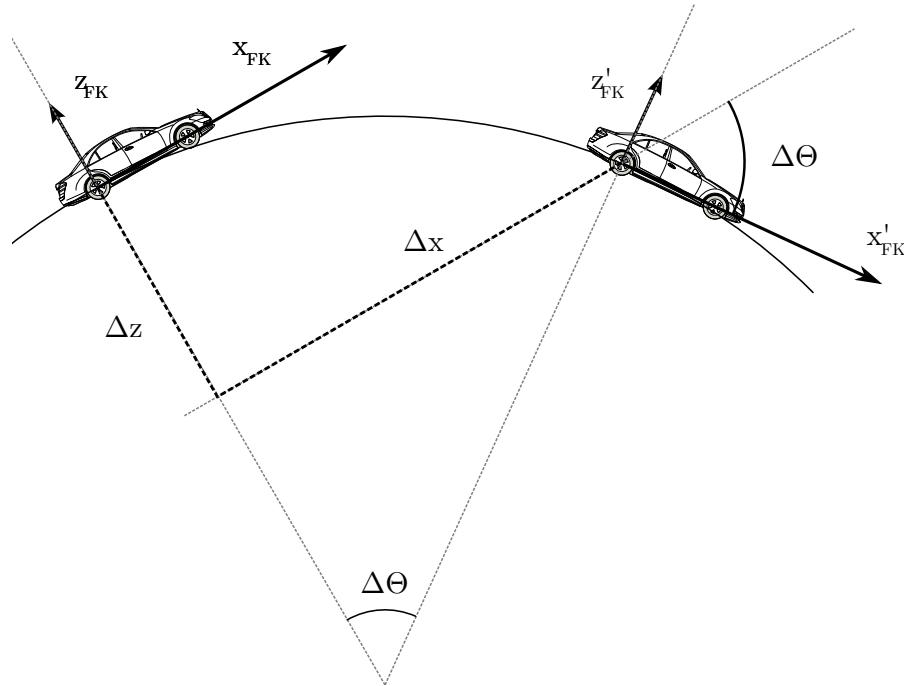


Abbildung 4.5 – Modellierter Nickbewegung des Fahrzeuges bei unebener Straße.

$$\Delta x = v \sin(\dot{\Theta} \Delta t) \dot{\Theta}^{-1} \quad (4.4)$$

$$\Delta z = v(1 - \cos(\dot{\Theta} \Delta t)) \dot{\Theta}^{-1} \quad (4.5)$$

Die Rotation des Fahrzeuges wird durch die kartesische 3x3-Rotationsmatrix R_{xyz} beschrieben, siehe Gleichung (3.5). Mit den Gleichungen (4.2)–(4.5) und der Annahme, dass sich die Winkel Φ , Θ , Ψ und die Geschwindigkeit v linear mit der Zeit ändern und die Winkelgeschwindigkeiten $\dot{\Phi}$, $\dot{\Theta}$, $\dot{\Psi}$ und die Beschleunigung a konstant sind, kann die Zustandsübergangsfunktion $f(\mathbf{x}_t, \Delta t)$ aufgestellt werden:

$$\mathbf{x}_{t+\Delta t} = f(\mathbf{x}_t, \Delta t) = \begin{bmatrix} x \\ y \\ z \\ \Phi + \dot{\Phi} \Delta t \\ \Theta + \dot{\Theta} \Delta t \\ \Psi + \dot{\Psi} \Delta t \\ \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \\ v + a \Delta t \\ a \end{bmatrix} + s(v, \dot{\Theta}, \dot{\Psi}, \Delta t) \quad (4.6)$$

Dabei fasst die Funktion $s(v, \dot{\Theta}, \dot{\Psi}, \Delta t)$ die drei Translationskomponenten zusammen:

$$s(v, \dot{\Theta}, \dot{\Psi}, \Delta t) = R_{xyz} \begin{bmatrix} v \sin(\dot{\Psi} \Delta t) \dot{\Psi}^{-1} + v \sin(\dot{\Theta} \Delta t) \dot{\Theta}^{-1} \\ v(1 - \cos(\dot{\Psi} \Delta t)) \dot{\Psi}^{-1} \\ v(1 - \cos(\dot{\Theta} \Delta t)) \dot{\Theta}^{-1} \end{bmatrix} \quad (4.7)$$

Hierbei sei erwähnt, dass jeweils Sonderbehandlungen auszuführen sind, wenn entweder die Nickrate $\dot{\Theta}$, die Gierrate $\dot{\Psi}$ oder beide Null werden, da ansonsten die Translationskomponenten nicht mehr definiert sind. Dies ist entsprechend durch bedingte Verzweigungen gelöst, die jeweils eine entsprechend einfachere Zustandsübergangsfunktion realisieren. Aufgrund der Übersichtlichkeit wird an dieser Stelle auf eine Darstellung verzichtet.

Zur Verwendung mit einem erweiterten Kalman-Filter ist des Weiteren die Linearisierung der Zustandsübergangsfunktion notwendig, die über eine Jacobimatrix berechnet wird:

$$\mathbf{F} = \frac{\partial f(\mathbf{x}, \Delta t)}{\partial \mathbf{x}} \quad (4.8)$$

Mit diesem Wissen kann nun der Systemzustand \mathbf{x} und die Prozessfehlerkovarianzmatrix \mathbf{P} im Prädiktionsschritt des EKF prädiziert werden:

$$\hat{\mathbf{x}}_k^* = f(\hat{\mathbf{x}}_{k-1}, \Delta t) \quad (4.9)$$

$$\mathbf{P}_k^* = \mathbf{F}_{k-1} \hat{\mathbf{P}}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \quad (4.10)$$

Kovarianzmatrix Prozessrauschen

Die in der Zustandsübergangsfunktion (4.6) als konstant angenommenen Komponenten sind in der Realität im Allgemeinen nicht konstant. Um dies im EKF abzubilden, werden die Änderungen dieser Komponenten in der Kovarianzmatrix des Prozessrauschens \mathbf{Q} als Rauschen berücksichtigt. \mathbf{Q} ist definiert als Diagonalmatrix:

$$\mathbf{Q} = \text{diag} \left(0, 0, 0, 0, 0, 0, \sigma_{\dot{\Phi}}^2, \sigma_{\dot{\Theta}}^2, \sigma_{\dot{\Psi}}^2, 0, \sigma_a^2 \right) \quad (4.11)$$

Zur konkreten Wahl der Größen $\sigma_{\dot{\Phi}}$, $\sigma_{\dot{\Theta}}$, $\sigma_{\dot{\Psi}}$ und σ_a sei auf Abschnitt 4.6.2 verwiesen.

4.4 Odometrie

Die Odometrie selbst zählt nicht zu den Lokalisierungsverfahren, sondern beschreibt die erwartete Änderung der Pose des Sensorfahrzeuges basierend auf den Aktoren im Fahrzeug: Den Radumdrehungen und dem Lenkwinkel. Da die Odometrie einen guten Anhaltspunkt über die Bewegung des Sensorfahrzeuges liefert, wird sie häufig als Initialisierung für Selbstlokalisierungsalgorithmen verwendet. Das ESP-Steuergerät ermittelt aus den einzelnen Rad-drehzahlen bereits eine kombinierte Fahrzeuggeschwindigkeit v_o . Ein Gierratensensor liefert die Gierrate $\dot{\Psi}_o$.

4.4.1 Modellierung einer Messung

Um den Zustand des Fahrzeuges \mathbf{x} zu schätzen, werden Messungen der Odometrie in dem Messvektor \mathbf{y}_o zusammengefasst:

$$\mathbf{y}_o = \begin{bmatrix} v_o & \dot{\Psi}_o \end{bmatrix}^T \quad (4.12)$$

Das Messrauschen \mathbf{R}_o wird wie folgt definiert:

$$\mathbf{R}_o = \begin{bmatrix} \sigma_{v_o}^2 & 0 \\ 0 & \sigma_{\dot{\Psi}_o}^2 \end{bmatrix} \quad (4.13)$$

Zur Wahl der konkreten Größen von v_o und $\sigma_{\dot{\Psi}_o}$ sei auf Abschnitt 4.6.2 verwiesen.

4.4.2 Integration der Messungen

Eine Messung des ESP-Sensors $\mathbf{y}_{o,t}$ vom Zeitpunkt t wird über die Messmatrix \mathbf{G}_o in das EKF integriert, die den Zustandsvektor direkt auf den Messvektor abbildet:

$$\mathbf{G}_o = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (4.14)$$

Über diese Messungen erreicht die Eigenbewegungsschätzung bereits akzeptable Ergebnisse, die häufig auch direkt so verwendet werden. Über die Zeit akkumulieren sich Fehler jedoch stark auf, die beispielsweise durch Reifenschlupf oder Unterschiede im Reifendruck entstehen.

Um den entstehenden Fehler zu minimieren und sowohl Schleuderbewegungen als auch Nick- und Wankbewegungen des Fahrzeuges erfassen zu können, werden im Folgenden Möglichkeiten untersucht, um zusätzlich Messungen aus der Umgebungserfassung in die Fahrzeugpositionsschätzung zu integrieren.

4.5 Rasterbasierte Selbstlokalisierung

Bei der rasterbasierten Selbstlokalisierung wird ein möglichst hoher Grad der Überlagerung zwischen zwei Belegungskarten oder einer bestehenden Belegungskarte und der aktuellen Messung gesucht. Genauer gesagt wird die Transformationsmatrix $\mathbf{T}^{\text{FK}_{t-1} \rightarrow \text{FK}_t}$ gesucht, die eine Überführung der Fahrzeugpose zum Zeitpunkt $t - 1$ zur Fahrzeugpose zum Zeitpunkt t darstellt. Diese Transformation wird in der Regel über eine Registrierung der aktuellen Messung entweder mit der vorangegangenen Messung oder aber mit einer akkumulierten Repräsentation einer Historie vorangegangener Messungen ermittelt. Durch diese Registrierung liegt direkt eine Messung der gesuchten Parameter x, y, z, Φ, Θ und Ψ vor.

Um bekannte Optimierungsverfahren wie den Simplex-Algorithmus einsetzen zu können, wird eine Gütefunktion definiert, die den Grad der Überlagerung angibt. Um die aktuelle Messung nicht zuerst in eine rasterbasierte Darstellung überführen zu müssen, was Diskretisierungsfehler mit sich bringen würde, wird eine Gütefunktion gesucht, welche die Überlagerung einer Messung im kontinuierlichen Raum zu einer Belegungskarte liefert.

4.5.1 Definition der Gütefunktion

Da die Gütefunktion die Überlagerung einer Messung mit der Belegungskarte beschreibt, sind Abfragen von Zellen der Karte notwendig. Dies erfolgt unter Berücksichtigung der Messunsicherheiten. Es wird zunächst das in Abschnitt 3.5.4 beschriebene Verfahren benutzt, um die Menge aller Zellen $n_{i,d} \in \mathbf{N}_d$ in einem Bereich Q_d um den d -ten Messpunkt \mathbf{p}_d der Messung \mathbf{y} zu bestimmen. $m(n_i)$ bezeichnet hierbei den Mittelpunkt des Knotens i in Kartenkoordinaten. Der Bereich Q wird jeweils bestimmt als umschreibender Quader um das 3σ -Ellipsoid des d -ten Messpunktes \mathbf{p}_d .

Die Gütefunktion h ist dann definiert als die Summe der Wahrscheinlichkeiten aller belegten Zellen, die von Messungen überstrichen werden, gewichtet nach deren Messunsicherheiten (4.15). Diese Gütefunktion hat zwei Vorteile: Zum einen muss nicht zuerst aufwändig eine rasterbasierte Repräsentation der Messung erstellt werden, zum anderen werden damit Diskretisierungsfehler vermieden.

Die Gütefunktion h ist abhängig von der Transformationsmatrix $\mathbf{T}^{\text{SK} \rightarrow \text{KK}}$, die wiederum aus der Sensoreinbaulage $\mathbf{T}^{\text{SK} \rightarrow \text{FK}}$ (konstant) und aus der Matrix $\mathbf{T}^{\text{FK} \rightarrow \text{KK}}$ zusammengesetzt ist. $\mathbf{T}^{\text{FK} \rightarrow \text{KK}}$ kann aus der Pose \mathbf{x}_F berechnet werden. Eine Maximierung der Gütefunktion resultiert daher in einer Bestimmung der Pose des Sensorfahrzeuges, bei der die aktuelle Messung optimal zur bestehenden Karte passt.

$$h(\mathbf{x}_F) = \sum_d \sum_i \left(p(n_{i,d}) - \frac{1}{2} \right) \cdot \exp \left\{ -\frac{1}{2} \left(m(n_{i,d}) - \mathbf{p}_d^{\text{KK}} \right)^T \mathbf{C}_{mcd}^{-1} \left(m(n_{i,d}) - \mathbf{p}_d^{\text{KK}} \right) \right\} \quad (4.15)$$

wobei

$$\mathbf{p}_d^{\text{KK}} = \mathbf{T}^{\text{FK} \rightarrow \text{KK}}(\mathbf{x}_F) \cdot \mathbf{T}^{\text{SK} \rightarrow \text{FK}} \cdot \mathbf{p}_d \quad (4.16)$$

Die Gütefunktion h ist nichtlinear. Im Folgenden werden daher nichtlineare Optimierungsverfahren zur Maximierung der Gütefunktion betrachtet. Da diese häufig als Minimierungsprobleme formuliert sind, wird das Maximierungsproblem in ein Minimierungsproblem überführt und eine Kostenfunktion $\text{cost}(\mathbf{x}_F)$ definiert:

$$\text{cost}(\mathbf{x}_F) = -h(\mathbf{x}_F) \quad (4.17)$$

4.5.2 Definition der Messung

Um die Pose des Fahrzeuges \mathbf{x}_F zu schätzen, wird die Messung der Pose des Fahrzeuges in einem Messvektor \mathbf{y}_{F_g} zusammengefasst. Dieser wird durch eine Registrierung der aktuellen Messung mit der bestehenden Umgebungskarte bestimmt:

$$\mathbf{y}_{F_g} = \left[x_g \quad y_g \quad z_g \quad \Phi_g \quad \Theta_g \quad \Psi_g \right]^T \quad (4.18)$$

Messrauschen

Die Messunsicherheit für die geschätzte Pose des Fahrzeuges im Raum ist notwendig, um eine Aussage darüber treffen zu können, wie zuverlässig die Schätzung in unterschiedlichen Richtungen ist. Diese Information ist insbesondere bei der Fusion von Messungen unterschiedlicher Quellen hilfreich.

Wenn die geschätzte Pose des Fahrzeuges als Zufallsvariable betrachtet wird und die Gütefunktion h als zugehörige Wahrscheinlichkeitsdichteverteilung, kann durch eine Varianzschätzung die Messunsicherheit bestimmt werden. Die Messunsicherheit wird in der 6x6-Kovarianzmatrix \mathbf{R}_g angegeben. Vereinfachend werden die Unsicherheiten je Koordinatenachse als unkorreliert angenommen. Die daraus resultierende Überschätzung der wahren Unsicherheit wird zu Gunsten der höheren Effizienz in Kauf genommen. \mathbf{R}_g kann dann wie folgt definiert werden:

$$\mathbf{R}_g = \text{diag} \left(\sigma_{x_g}^2, \sigma_{y_g}^2, \sigma_{z_g}^2, \sigma_{\Phi_g}^2, \sigma_{\Theta_g}^2, \sigma_{\Psi_g}^2 \right) \quad (4.19)$$

Die einzelnen Varianzen werden über einen Varianzschätzer geschätzt. Die Varianz einer stetigen Zufallsvariablen X ist wie folgt definiert:

$$\text{Var}(X) = \int_{-\infty}^{\infty} (x - \mu)^2 f(x) dx \quad (4.20)$$

Da die Funktion h nicht analytisch verfügbar ist, wird sie nur an einigen diskreten Stellen $\mathbf{x}_F + s_k$ ausgewertet. Durch Addition von s_k wird jeweils nur eine Komponente von \mathbf{x}_F variiert, beispielsweise die erste Komponente:

$$s_k = \begin{bmatrix} s_0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad (4.21)$$

Dabei wird s_0 aus einem definierten Intervall um 0 mit konstanter Schrittweite gewählt. Es gilt dann:

$$\text{Var}(X) = \sum_k \left(\mathbf{x}_F + s_k - \mathbf{y}_{F_g} \right)^2 h(\mathbf{x}_F + s_k). \quad (4.22)$$

Um die Eigenschaft

$$\sum_k h(\mathbf{x}_F + s_k) = 1 \quad (4.23)$$

einer Wahrscheinlichkeitsdichteverteilung einzuhalten, wird die Summe durch den Faktor η auf 1 normiert. Als Mittelwert wird der durch das Optimierungsverfahren ermittelte Punkt maximaler Güte \mathbf{p}_b , welcher der gemessenen Pose \mathbf{y}_{F_g} entspricht, verwendet. Die Varianz $\sigma_{x_g}^2$ kann dann approximiert werden durch:

$$\sigma_{x_g}^2 = \eta \sum_k \left(\mathbf{x}_F + s_k - \mathbf{y}_{F_g} \right)^2 h(\mathbf{x}_F + s_k) \quad (4.24)$$

Die Varianzen $\sigma_{y_g}^2$, $\sigma_{z_g}^2$, $\sigma_{\Phi_g}^2$, $\sigma_{\Theta_g}^2$ und $\sigma_{\Psi_g}^2$ werden analog ermittelt.

Messmatrix

Eine Messung $\mathbf{y}_{g,t}$ zum Zeitpunkt t kann direkt über die Messmatrix \mathbf{G}_g in das EKF integriert werden, wobei \mathbf{G}_g die Elemente aus \mathbf{x} passend auf \mathbf{y} abbildet:

$$\mathbf{G}_g = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.25)$$

4.5.3 Nichtlineare Optimierungsverfahren

Gradientenverfahren

Das Gradientenverfahren ist ein Optimierungsverfahren ohne Nebenbedingungen, das eingesetzt werden kann, um eine reellwertige, differenzierbare Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$ zu minimieren.

Die Kostenfunktion cost erfüllt diese Bedingungen, der Gradient ∇cost kann an der Stelle \mathbf{x}_F wie folgt berechnet werden:

$$\nabla \text{cost} = \begin{pmatrix} \frac{\delta \text{cost}}{\delta x} \\ \frac{\delta \text{cost}}{\delta y} \\ \frac{\delta \text{cost}}{\delta z} \\ \frac{\delta \text{cost}}{\delta \Phi} \\ \frac{\delta \text{cost}}{\delta \Theta} \\ \frac{\delta \text{cost}}{\delta \Psi} \end{pmatrix} \quad (4.26)$$

Für eine effiziente Implementierung muss die Kostenfunktion cost jedoch approximiert werden, indem die Messunsicherheiten jeweils nur im 3σ -Intervall betrachtet werden. Dies führt zu Unstetigkeitsstellen, so dass ein Gradientenverfahren nicht ohne weiteres einsetzbar ist.

Downhill-Simplex-Verfahren

Das Downhill-Simplex-Verfahren nach Nelder und Mead [92] dient der Minimierung nichtlinearer Funktionen und kommt dabei ohne eine Ableitung aus. Die Anzahl der Freiheitsgrade wird hier mit N bezeichnet. Im Fall der 2D-Lokalisierung mit $N = 3$ gibt es vier Anfangspunkte, bei der 3D-Lokalisierung mit $N = 6$ gibt es sieben Anfangspunkte.

Das Verfahren besteht aus mehreren Schritten:

1. Das Simplex S wird aus $(N + 1)$ Anfangspunkten aufgebaut. Ein Anfangspunkt ist die im Prädiktionsschritt berechnete Fahrzeugpose, die anderen N Punkte werden durch eine Schrittweite S variiert.
2. Die Funktionswerte an diesen Punkten werden berechnet. Es wird jeweils der beste Punkt $\mathbf{p}_b := \min_{\mathbf{p} \in S} \text{cost}(\mathbf{p})$ und der schlechteste Punkt $\mathbf{p}_w := \max_{\mathbf{p} \in S} \text{cost}(\mathbf{p})$ unter ihnen bestimmt.
3. Ist \mathbf{p}_b ausreichend nahe am Minimum (der Gradient an dieser Stelle ist kleiner ϵ), wird das Verfahren erfolgreich beendet. Ansonsten wird der Punkt \mathbf{p}_w mit dem schlechtesten Funktionswert durch einen neuen Punkt ersetzt, der durch das Verfahren von Nelder und Mead bestimmt wird. Das Verfahren wird im Folgenden beschrieben.
4. Fahre mit Schritt 2 fort.

Im Verfahren nach Nelder und Mead werden neben den $(N + 1)$ Punkten des Simplex folgende weitere zwei Punkte betrachtet: Das Zentrum der N besten Punkte, bezeichnet mit z :

$$z := \frac{1}{N} \sum_{i, i \neq w} \mathbf{p}_i \quad (4.27)$$

und der Punkt \mathbf{p}_r , der durch Reflexion des schlechtesten Punktes am Zentrum entsteht:

$$\mathbf{p}_r := z + \alpha(z - \mathbf{p}_w) \quad (4.28)$$

Hierbei setzt man im Allgemeinen $\alpha = 1$. Der Punkt mit dem zweitschlechtesten Funktionswert sei mit \mathbf{p}_s bezeichnet.

Pro Iterationsschritt des Verfahrens werden drei Fälle unterschieden:

- Wenn $\text{cost}(\mathbf{p}_b) \leq \text{cost}(\mathbf{p}_r) \leq \text{cost}(\mathbf{p}_s)$, also der reflektierte Punkt zwischen dem besten Punkt und dem zweitschlechtesten Punkt liegt, wird ein Reflexionsschritt durchgeführt. Dieser ist exemplarisch in Abbildung 4.6a für $N = 2$ skizziert.

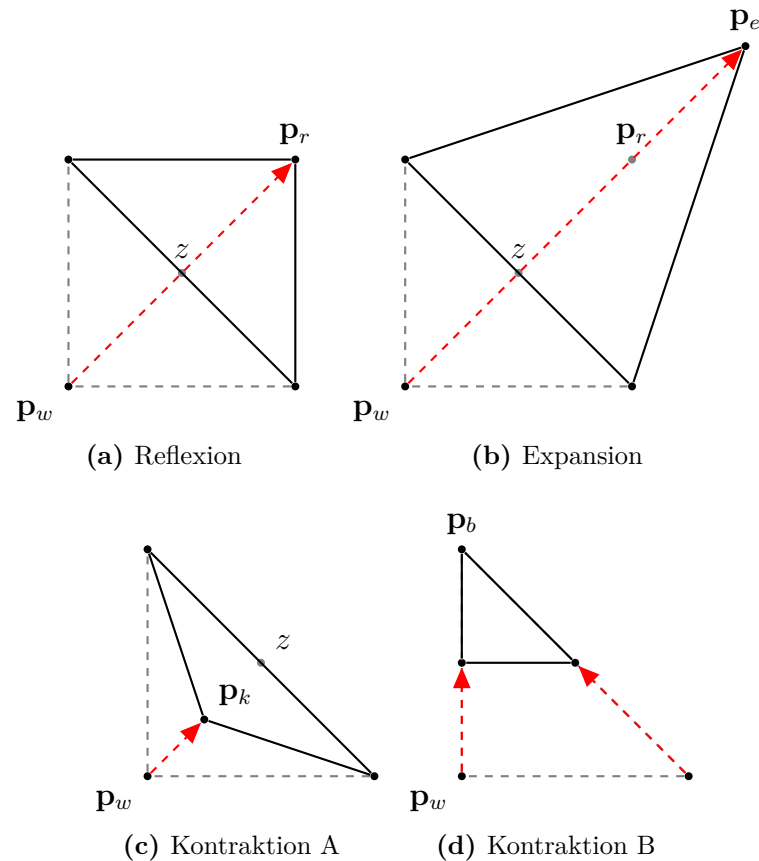


Abbildung 4.6 – Mögliche Schritte im Downhill-Simplex-Verfahren nach Nelder und Mead.

- Wenn $\text{cost}(\mathbf{p}_r) < \text{cost}(\mathbf{p}_b)$, also der reflektierte Punkt besser als der bisher beste Punkt ist, wird ein weiterer expandierter Punkt \mathbf{p}_e betrachtet:

$$\mathbf{p}_e := z + \beta(z - \mathbf{p}_w)$$

Im Allgemeinen wird $\beta = 2$ gewählt. Gilt $\text{cost}(\mathbf{p}_e) < \text{cost}(\mathbf{p}_r)$, so wird ein Expansionschritt durchgeführt; dieser ist in Abbildung 4.6b skizziert. Andernfalls wird lediglich die Reflexion durchgeführt.

- Wenn $\text{cost}(\mathbf{p}_r) > \text{cost}(\mathbf{p}_s)$, also der reflektierte Punkt schlechter als der bisher zweit-schlechteste Punkt ist, wird der kontrahierte Punkt \mathbf{p}_k betrachtet. Dieser ist definiert als:

$$\mathbf{p}_k := \begin{cases} z + \gamma(\mathbf{p}_w - z), & \text{cost}(\mathbf{p}_r) > \text{cost}(\mathbf{p}_w) \\ z + \gamma(z - \mathbf{p}_w), & \text{sonst} \end{cases} \quad (4.29)$$

Dabei wird γ im Allgemeinen zu $\frac{1}{2}$ gewählt. Ist $\text{cost}(\mathbf{p}_k) < \text{cost}(\mathbf{p}_w)$, also der kontrahierte Punkt besser als der bisher schlechteste, so wird \mathbf{p}_w durch \mathbf{p}_k ersetzt. Diese Kontraktion Typ A ist in Abbildung 4.6c skizziert. Ist der kontrahierte Punkt schlechter

als der bisher schlechteste, so wird das Simplex in Richtung des Punktes \mathbf{p}_b kontrahiert. Dieser Schritt ist in Abbildung 4.6d skizziert. Die neuen Punkte \mathbf{p}_i^* werden berechnet durch:

$$\mathbf{p}_i^* := \frac{\mathbf{p}_i + \mathbf{p}_b}{2}, \mathbf{p}_i \in S, i \neq b$$

Die Kostenfunktion cost besitzt im Allgemeinen mehrere Minima. Die Zeitschritte Δt , zu denen die Selbstlokalisierung durchgeführt wird, haben in der Regel einen geringen Abstand von beispielsweise 0,1 s. Zudem wird der Zustand auf den aktuellen Zeitpunkt prädiziert und gegebenenfalls durch einen Innovationsschritt durch die Odometrie aktualisiert. Deshalb wird angenommen, dass die Lokalität zur Auswertung des absoluten Minimums erfüllt ist. Abbildung 4.7 zeigt exemplarisch die Gütefunktion einer Beispielsequenz.

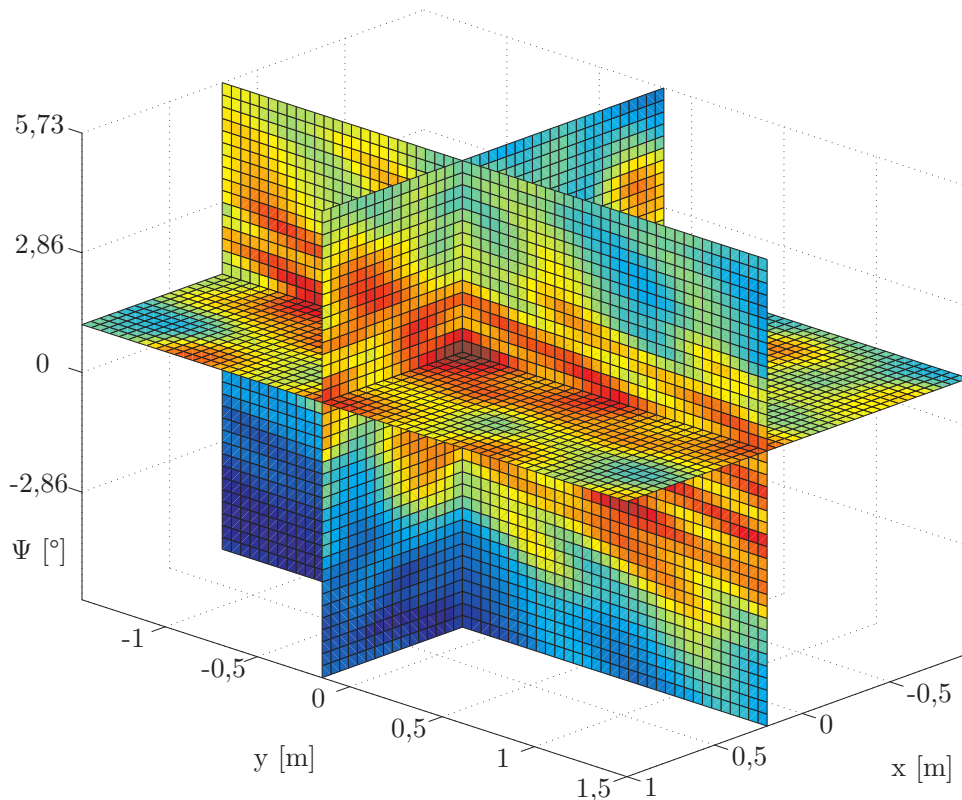


Abbildung 4.7 – Visualisierung der dreidimensionalen Gütefunktion für eine Beispielsequenz. Im Maximum sind jeweils Schnitte durch die x-y-, die x- Ψ - und die y- Ψ -Ebene gelegt. Neben dem Maximum zeigen sich mehrere Nebenmaxima.

In dieser Arbeit wird auf die Implementierung des Downhill-Simplex-Verfahrens aus der GNU Mathematikbibliothek (GSL) [1] zurückgegriffen.

4.6 Ergebnisse

4.6.1 Versuchsträger zur Realdatenauswertung

Zur Auswertung erfolgt eine Installation des implementierten Systems auf einem Versuchsträger. Das Serienfahrzeug wurde mit einem Messtechnikrechner und der notwendigen Sensorik ausgestattet. Abbildung 4.8a zeigt den Versuchsträger, eine Mercedes-Benz S-Klasse [19], auf dem Gelände des Daimler-Forschungszentrums in Ulm. In Abbildung 4.8b ist der Dachgepäckträger zu sehen, auf dem ein Velodyne HDL-64E, eine GPS- und eine GSM-Antenne montiert sind. Sowohl die GPS- als auch die GSM-Antenne gehören zur Inertialmesseinheit, die im Kofferraum installiert ist.



(a) Der Versuchsträger.



(b) Links im Bild ist der Velodyne zu sehen, im Vordergrund die GSM-Antenne und rechts die GPS-Antenne.

Abbildung 4.8 – Versuchsträger zur Realdatenauswertung.

Umgebungserfassende Sensorik

Zur Umgebungserfassung wird ein Velodyne HDL-64E verwendet, wie in Abschnitt 2.4.2 vorgestellt. Für die zweidimensionale Selbstlokalisierung wird der Einfachheit halber jedoch lediglich eine der 64 Ebenen genutzt, und zwar diejenige, die parallel zur Fahrbahn verläuft.

Referenzsensorik: ADMA

Weiterhin ist das Fahrzeug mit einem „Automotive Dynamic Motion Analyzer“ (ADMA) [52] der GeneSys Elektronik GmbH aus Offenburg ausgerüstet.

Bei dem ADMA handelt es sich um eine auf Laserkreisel basierende Inertialsensorik (IMU). Diese wird von einem GPS mit Differentialsignal (DGPS) gestützt und arbeitet damit mit einer Genauigkeit von 1-2 Zentimetern in der Positionsschätzung. Abbildung 4.9 zeigt die Inertialmesseinheit. Die Beschleunigungssensoren arbeiten mit einer Genauigkeit von 1 mg. Die faseroptischen Kreisel haben einen Drift von $6^\circ/\text{h}$. Die Aufzeichnungen erfolgen per CAN-Bus mit 100 Hz.

Zur Korrektur des GPS-Signals werden per GSM-Verbindung fortwährend GPS-Korrekturdaten des Satellitenpositionierungsdienstes SAPOS [27] empfangen und ausgewertet. Um eine korrekte Arbeitsweise sicherzustellen wurde sowohl die Einbauposition im Kofferraum als auch die Position der GPS-Antenne auf dem Dach exakt vermessen und im Gerät konfiguriert.



Abbildung 4.9 – ADMA

4.6.2 Bestimmung von Mess- und Prozessrauschen

Über den ADMA kann sowohl das Messrauschen des ESP-Sensors als auch das Prozessrauschen der Eigenbewegungsschätzung ermittelt werden.

Messrauschen der Odometrie

In Abschnitt 4.4.1 blieb die Wahl der Größen σ_{v_o} und $\sigma_{\dot{\psi}_o}$ offen. Zur Bestimmung des Messrauschens der Odometriedaten aus dem ESP-Sensor werden diese mit den Referenzdaten aus dem ADMA verglichen.

Mit dem Versuchsträger wurden dazu mehrere Manöver unter Extrembedingungen aufgezeichnet, das heißt mit maximaler Beschleunigung, Verzögerung und Drehrate auf trockenem Asphalt. In Abbildung 4.10 sind die zugehörigen Messwerte aufgezeichnet. Das Messrauschen der Referenzsensorik (R_v und $R_{\dot{\psi}}$) wird additiv berücksichtigt. Über alle Sequenzen ergeben sich folgende Standardabweichungen:

$$\sigma_{v_{o,\text{extrem}}} = 0,305 \left[\frac{m}{s} \right] + R_v \quad (4.30)$$

$$\sigma_{\dot{\Psi}_{o,\text{extrem}}} = 1,332 \left[\frac{\circ}{s} \right] + R_{\dot{\Psi}} \quad (4.31)$$

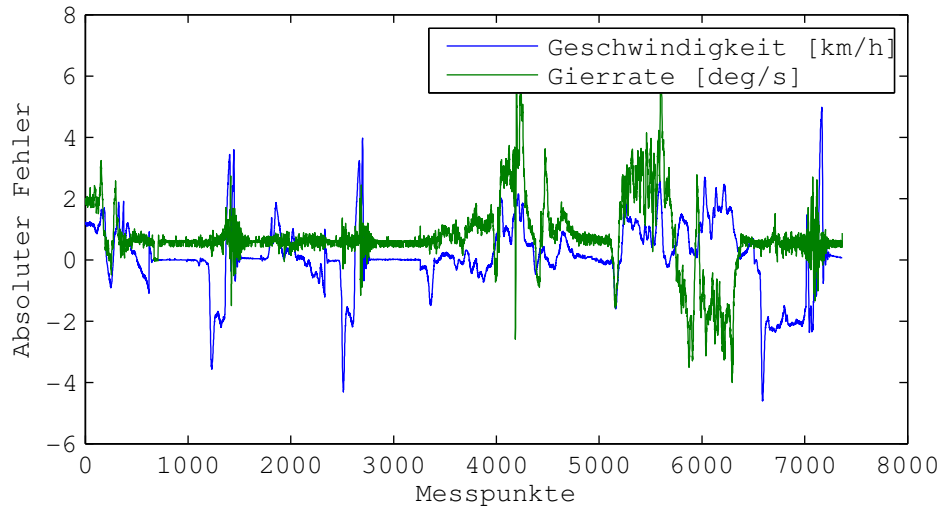


Abbildung 4.10 – Absoluter Fehler der Odometrie im Vergleich zur Referenz in Extremsituationen auf trockenem Asphalt.

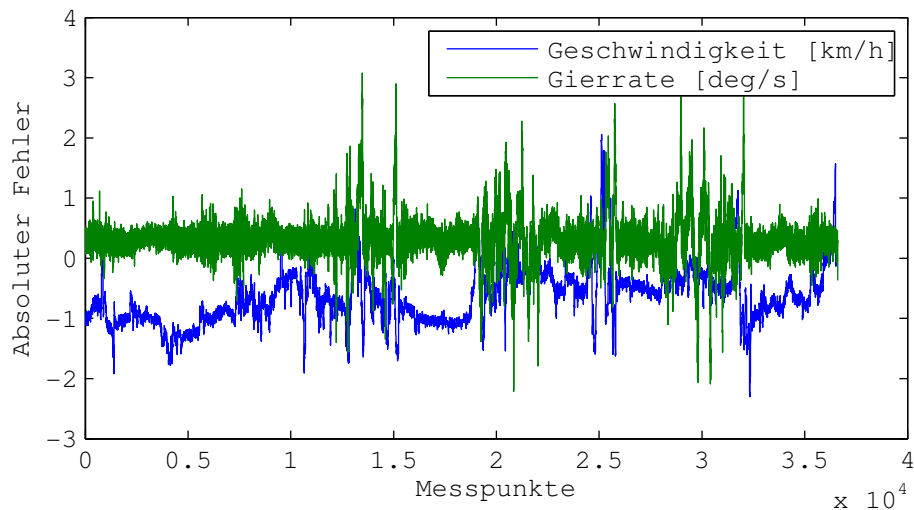


Abbildung 4.11 – Absoluter Fehler der Odometrie im Vergleich zur Referenz in normalen Fahrsituationen auf trockenem Asphalt.

Des Weiteren wurden mehrere Fahrten mit normaler Fahrweise ausgewertet. In Abbildung 4.11 sind die hier zugehörigen Messwerte aufgezeichnet. Die resultierenden Standardabweichungen sind erwartungsgemäß kleiner:

$$\sigma_{v_o} = 0,219 \left[\frac{m}{s} \right] + R_v \quad (4.32)$$

$$\sigma_{\dot{\Psi}_o} = 0,476 \left[\frac{^\circ}{s} \right] + R_{\dot{\Psi}} \quad (4.33)$$

Für die Evaluierung werden die Werte σ_{v_o} und $\sigma_{\dot{\Psi}_o}$ eingesetzt. Um mit Extremsituationen umgehen zu können, ist eine adaptive Anpassung auf $\sigma_{v_o, \text{extrem}}$ und $\sigma_{\dot{\Psi}_o, \text{extrem}}$ denkbar, basierend auf v und $\dot{\Psi}$. Diese Betrachtung ist auf eine trockene Fahrbahn eingeschränkt. Auf nasser oder eisglatter Fahrbahn würden sich andere Werte ergeben, die in dieser Arbeit jedoch nicht betrachtet werden.

Prozessrauschen

In Abschnitt 4.3.2 ist die konkrete Wahl der Standardabweichungen $\sigma_{\dot{\Phi}}$, $\sigma_{\dot{\Theta}}$, $\sigma_{\dot{\Psi}}$ und σ_a für die Prozessfehlerkovarianzmatrix \mathbf{Q} offen geblieben. Mit dem ADMA lassen sich diese näherungsweise bestimmen, indem die maximalen Werte der zu Grunde liegenden Größen ermittelt werden.

Über mehrere Fahrten in Grenzsituationen wurden für den Versuchsträger die absoluten Beschleunigungsänderungen und die absoluten Rotationsbeschleunigungen ermittelt. Hierbei wurde von einer Abtastzeit von $T = 0,1 \text{ s}$ ausgegangen:

$$\ddot{\Phi}_{\max} = 13,47 \left[\frac{^\circ}{s^2} \right] \quad (4.34)$$

$$\ddot{\Theta}_{\max} = 12,39 \left[\frac{^\circ}{s^2} \right] \quad (4.35)$$

$$\ddot{\Psi}_{\max} = 12,03 \left[\frac{^\circ}{s^2} \right] \quad (4.36)$$

$$\dot{a}_{\max} = 10,18 \left[\frac{m}{s^3} \right] \quad (4.37)$$

Werden die Maxima dieser Werte jeweils als 3σ -Intervall angenommen, führt dies zu den Standardabweichungen:

$$\sigma_{\ddot{\Phi}} = 4,49 \left[\frac{\circ}{s^2} \right] \quad (4.38)$$

$$\sigma_{\ddot{\Theta}} = 4,13 \left[\frac{\circ}{s^2} \right] \quad (4.39)$$

$$\sigma_{\ddot{\Psi}} = 4,01 \left[\frac{\circ}{s^2} \right] \quad (4.40)$$

$$\sigma_{\ddot{a}} = 3,39 \left[\frac{m}{s^3} \right] \quad (4.41)$$

Unter der Voraussetzung, dass die diskrete Schrittweite $\Delta k = 0,1 s$ beträgt, erhält man die Standardabweichungen zu den Drehraten und der Beschleunigung pro Zeitschritt:

$$\sigma_{\dot{\Phi}} = 0,45 \left[\frac{\circ}{s} \right] \quad (4.42)$$

$$\sigma_{\dot{\Theta}} = 0,41 \left[\frac{\circ}{s} \right] \quad (4.43)$$

$$\sigma_{\dot{\Psi}} = 0,4 \left[\frac{\circ}{s} \right] \quad (4.44)$$

$$\sigma_{\dot{a}} = 0,34 \left[\frac{m}{s^2} \right] \quad (4.45)$$

4.6.3 Auswertung mit realen Messdaten

Die gitterbasierte Selbstlokalisierung wurde mit realen Messdaten zweier unterschiedlicher Sequenzen evaluiert.

Sequenz A: Drei Achten vor einem Gebäude

Die erste Sequenz zur Evaluierung der Selbstlokisierungsalgorithmen besteht aus drei Achten, die auf dem Gelände des Daimler Forschungszentrums abgefahren wurden. In Abbildung 4.12 ist die Sequenz in der Draufsicht dargestellt. Im Hintergrund ist das Satellitenbild zu sehen, im Vordergrund die abgefahrne Trajektorie: Die abgefahrne Strecke entspricht drei „Achten“. Die Positionen, in denen nur einfacher GPS-Empfang ohne Korrekturdaten möglich war, sind rot gekennzeichnet. Die Positionen, bei denen zumindest grober DGPS-Empfang vorhanden war, sind in Gelb dargestellt. Sehr guter DGPS-Empfang war gar nicht vorhanden. Diese Sequenz ist sowohl für GPS mit den starken Abschattungen durch das hohe Gebäude als auch für die Odometrie herausfordernd, da nur geringe Geschwindigkeiten und Kurven mit maximal eingeschlagenem Lenkrad gefahren werden.

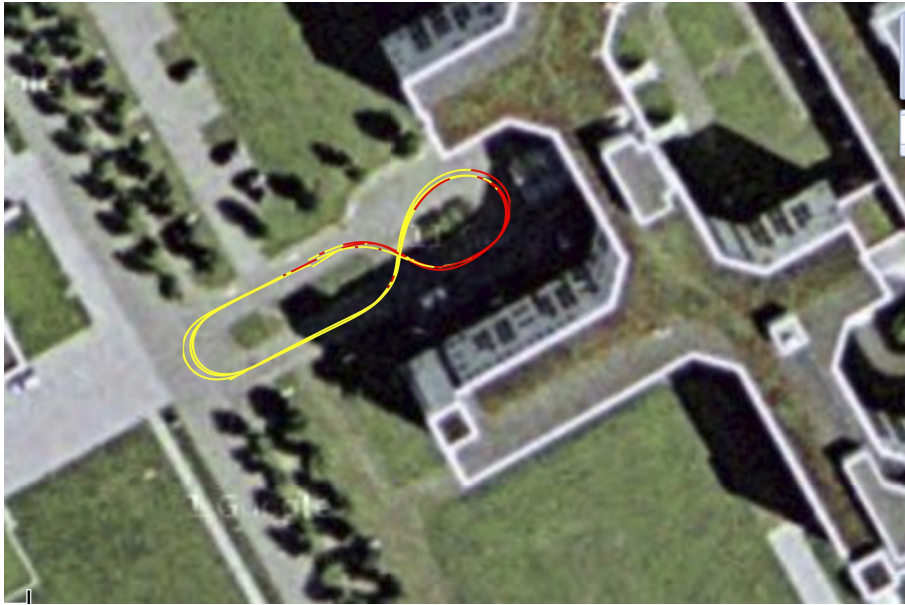


Abbildung 4.12 – Einfahrt vor einem Gebäude. Im Hintergrund ist das Satellitenbild zu sehen. Die Qualität der GPS-Daten in jeder Position ist farblich kodiert.

Abbildung 4.13 zeigt die zu dieser Sequenz geschätzten Positionsfehler des ADMA. Mit Korrekturdaten liegt der geschätzte Fehler konstant bei 0,03 m. Sobald aufgrund von Verschattungen durch Gebäude nicht mehr genügend Satelliten sichtbar sind, fällt die Positionskorrektur weg und die geschätzten Fehler wachsen schnell auf bis zu 0,3 m an.

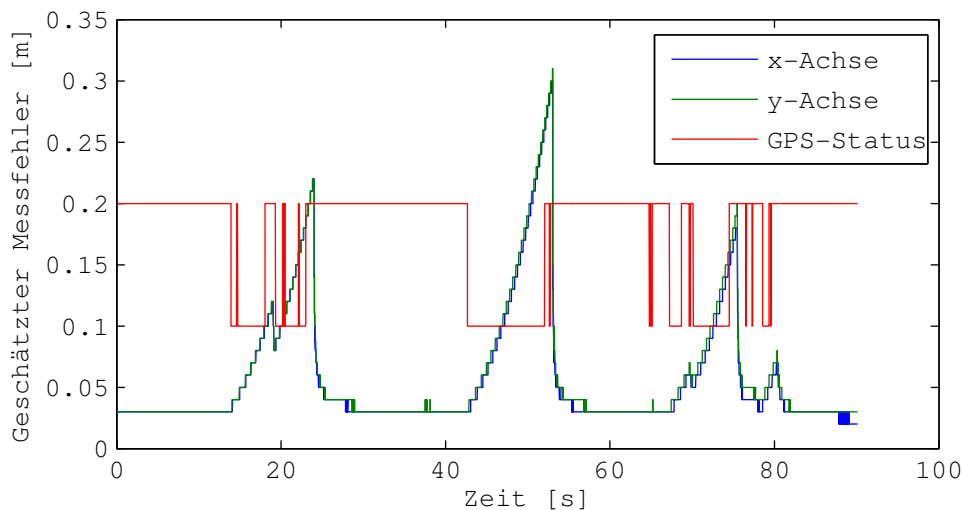


Abbildung 4.13 – Vom ADMA geschätzter Positionsfehler über die Sequenz in Abbildung 4.12. Mit Korrekturdaten (GPS-Status 0,2) liegt der geschätzte Fehler konstant bei 0,03 m. Sind jedoch nicht mehr genügend Satelliten sichtbar, fällt die Positionskorrektur weg (Status 0,1) und die geschätzten Fehler wachsen in dieser Sequenz schnell auf bis zu 0,3 m.

Evaluierung der zweidimensionalen Selbstlokalisierung vor einem Gebäude

Die rasterbasierte Selbstlokalisierung erfolgt durch das Downhill-Simplex-Verfahren. Die maximale Anzahl an Iterationen wird auf 50 beschränkt, um Echtzeitfähigkeit zu gewährleisten. Zur zweidimensionalen Selbstlokalisierung werden die Komponenten z , Φ und Θ auf 0 festgehalten, ansonsten erfolgt das Verfahren analog zur dreidimensionalen Selbstlokalisierung. Die Belegungskarte hat eine Auflösung von 0,2 m und ist auf einen Radius von 60 m um das Fahrzeug beschränkt. Damit ist die Szene fast zu jedem Zeitpunkt vollständig vom Laserscanner einsehbar.

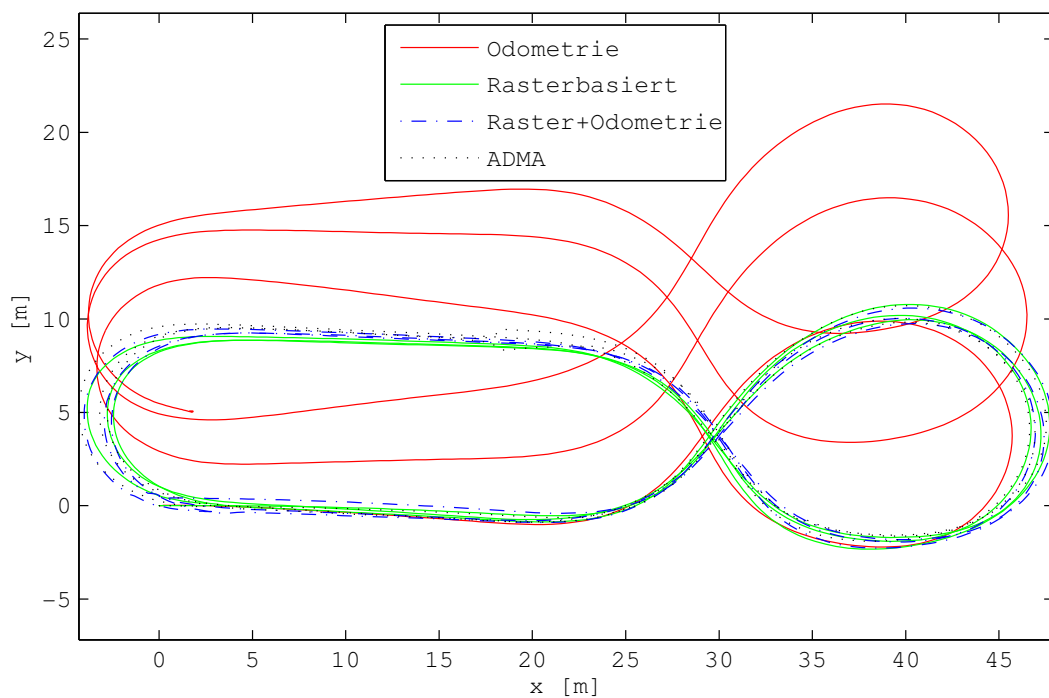


Abbildung 4.14 – Dreifach abgefahrene Acht vor einem Gebäude. Vergleich von rasterbasierter Selbstlokalisierung zu Odometrie und Referenz durch den ADMA. Startpunkt ist im Ursprung bei (0,0).

Abbildung 4.14 zeigt die Trajektorien der unterschiedlichen Verfahren. Die Trajektorie des ADMA bildet die Referenz. Die Fehler, die durch die Odometrie entstehen, sind am größten; sie akkumulieren sich unbegrenzt. Die Varianten der rasterbasierten Selbstlokalisierung liefern deutlich bessere Ergebnisse, die sehr nahe an der Referenz liegen. Abbildung 4.15 zeigt die mittleren Fehler der drei Varianten. Die Odometrie liefert über die gesamte Sequenz einen mittleren Fehler von 2,24 m zur Referenz. Durch die Fusion von rasterbasierter Selbstlokalisierung und Odometrie wird die Genauigkeit deutlich besser, hier liegt sie bei 0,64 m.

Durch den Vergleich der Abbildungen 4.16 und 4.17 wird deutlich, was dies für die Kartierung bedeutet. Abbildung 4.16 zeigt die resultierende Belegungskarte, die lediglich auf den

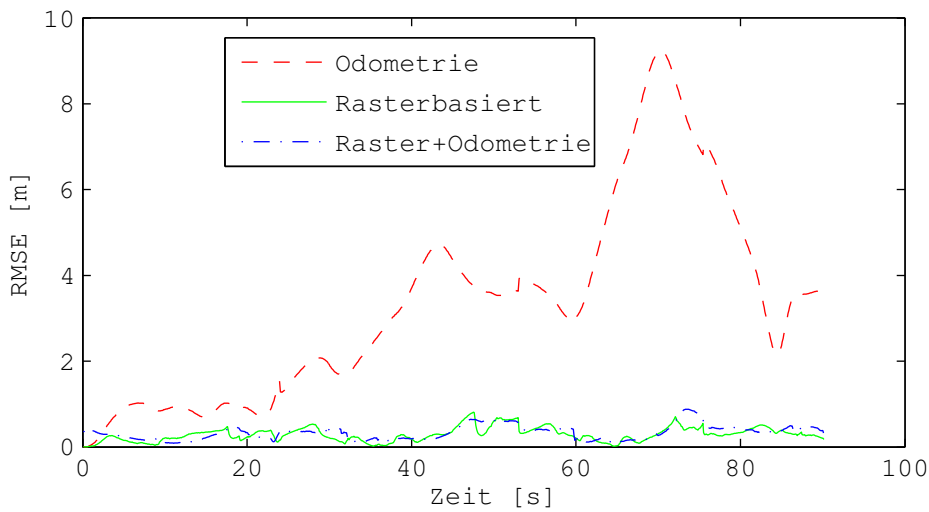


Abbildung 4.15 – Fehler in der Positionsschätzung. Verglichen werden Odometrie, rasterbasierte Selbstlokalisierung und rasterbasierte Selbstlokalisierung mit Odometrie jeweils zur Referenz durch den ADMA.

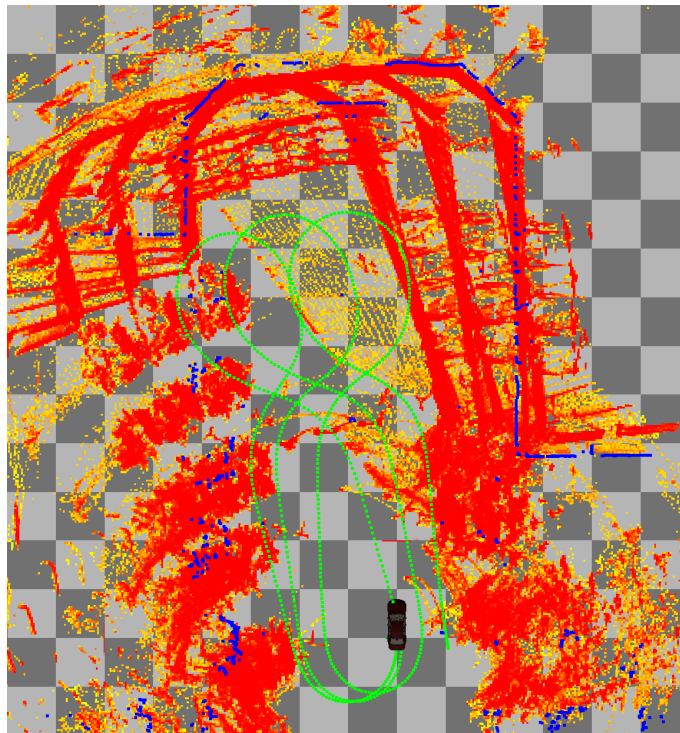
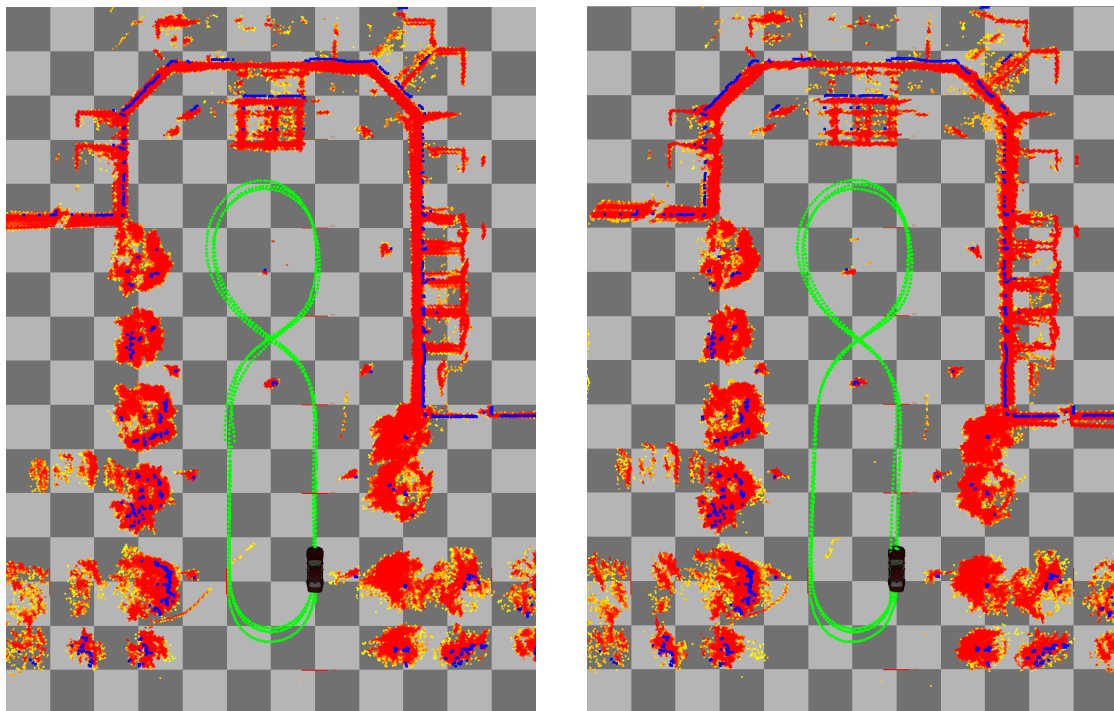


Abbildung 4.16 – Posenbestimmung durch die Odometrie. Durch die häufigen Richtungswechsel, starke Lenkwinkel und die langsame Fahrt ist die Odometrie alleine nicht geeignet, um eine genaue Belegungskarte aufzubauen.

Informationen der Odometrie beruht. Die sich stark akkumulierenden Posenfehler resultieren in einer stark verwischten Belegungskarte.



(a) Belegungskarte über Referenztrajektorie durch den ADMA

(b) Belegungskarte über Selbstlokalisierung: Fusion der Odometrie mit der rasterbasierten Methode

Abbildung 4.17 – Durch die Selbstlokalisierung mit der rasterbasierten Methode in Kombination mit der Odometrie (b) kann eine sehr genaue Positionsschätzung erreicht werden. Es lassen sich Ergebnisse erzielen, die der Referenz durch den ADMA (a) sehr nahe kommen.

In 4.17 werden zwei Belegungskarten gegenübergestellt: Abbildung 4.17a zeigt die Belegungskarte, die durch die Referenzsensorik entstand. Die Wände sind scharf und deutlich konturiert abgebildet. Über die drei Runden ergeben sich keine nennenswerten Unterschiede, da ständig absolute Positionsmessungen zur Verfügung stehen. Abbildung 4.17b zeigt die Belegungskarte, die durch das vorgeschlagene Verfahren zur Selbstlokalisierung entsteht. Obwohl keine absoluten Positionsmessungen zur Verfügung stehen, ist der Fehler deutlich geringer als bei der Odometrie. Die resultierende Karte reicht in ihrer Qualität nahezu an die der Referenzsensorik heran.

Evaluierung der dreidimensionalen Selbstlokalisierung vor einem Gebäude

Bei der dreidimensionalen Selbstlokalisierung gibt es sechs Freiheitsgrade. Damit konvergiert das Optimierungsverfahren in der Regel nicht so früh: Die maximale Anzahl an Iterationen wird deshalb auf 100 festgelegt. In der Draufsicht ähneln sich die Trajektorien stark denen in Abbildung 4.14. Interessant wird es in der Seitenansicht, wie in Abbildung 4.18 veranschaulicht. Hier werden die dreidimensionalen Trajektorien von der Seite gezeigt. Die drei

zusätzlichen Freiheitsgrade resultieren in einer größeren Abweichung zur Referenz als bei der zweidimensionalen Selbstlokalisierung.

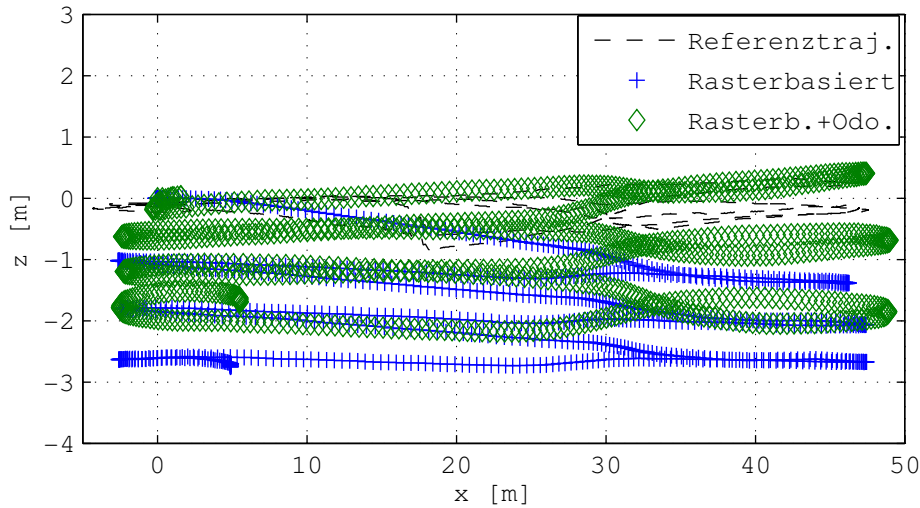


Abbildung 4.18 – Dreifach abgefahrene Acht vor einem Gebäude. Vergleich von rasterbasierter Selbstlokalisierung mit sechs Freiheitsgraden zu Referenz in der Seitenansicht. Startpunkt ist im Ursprung bei (0,0,0).

In Abbildung 4.19 sind die mittleren Fehler des dreidimensionalen Verfahrens dargestellt. Über die gesamte Sequenz erzeugt die Odometrie einen mittleren Fehler von 2,24 m und die rasterbasierte Selbstlokalisierung einen mittleren Fehler von 1,76 m. Mit der Fusion aus beiden Quellen lässt sich der mittlere Fehler auf 1,5 m reduzieren.

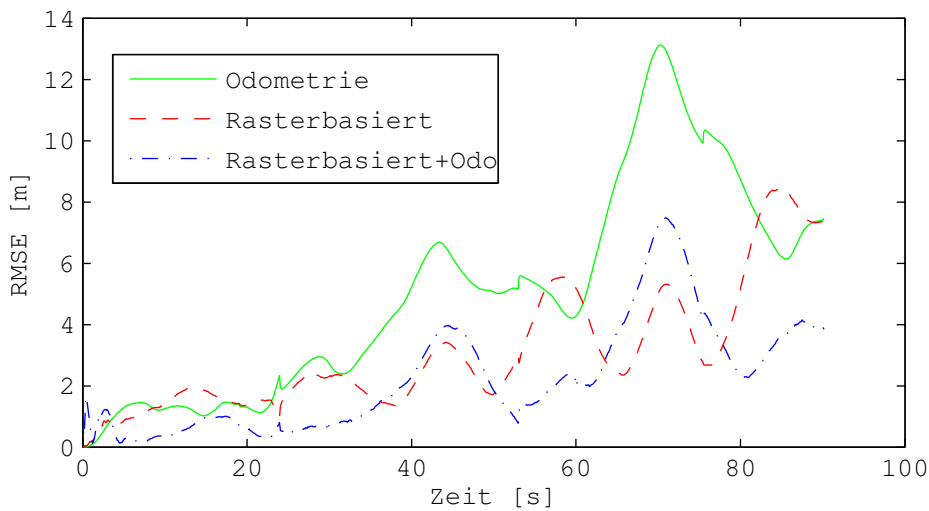


Abbildung 4.19 – Dreifach abgefahrene Acht vor einem Gebäude. Vergleich des mittleren Fehlers der rasterbasierten Selbstlokalisierung mit sechs Freiheitsgraden zur Referenz. Startpunkt ist im Ursprung bei (0,0,0).

Sequenz B: Fahrt auf einer Landstraße

Die zweite evaluierte Sequenz entstand auf einer Fahrt bei Ulm von der Sporthalle Nord nach Mähringen und zurück. Die Sequenz zeichnet sich durch starke Höhendifferenzen aus.

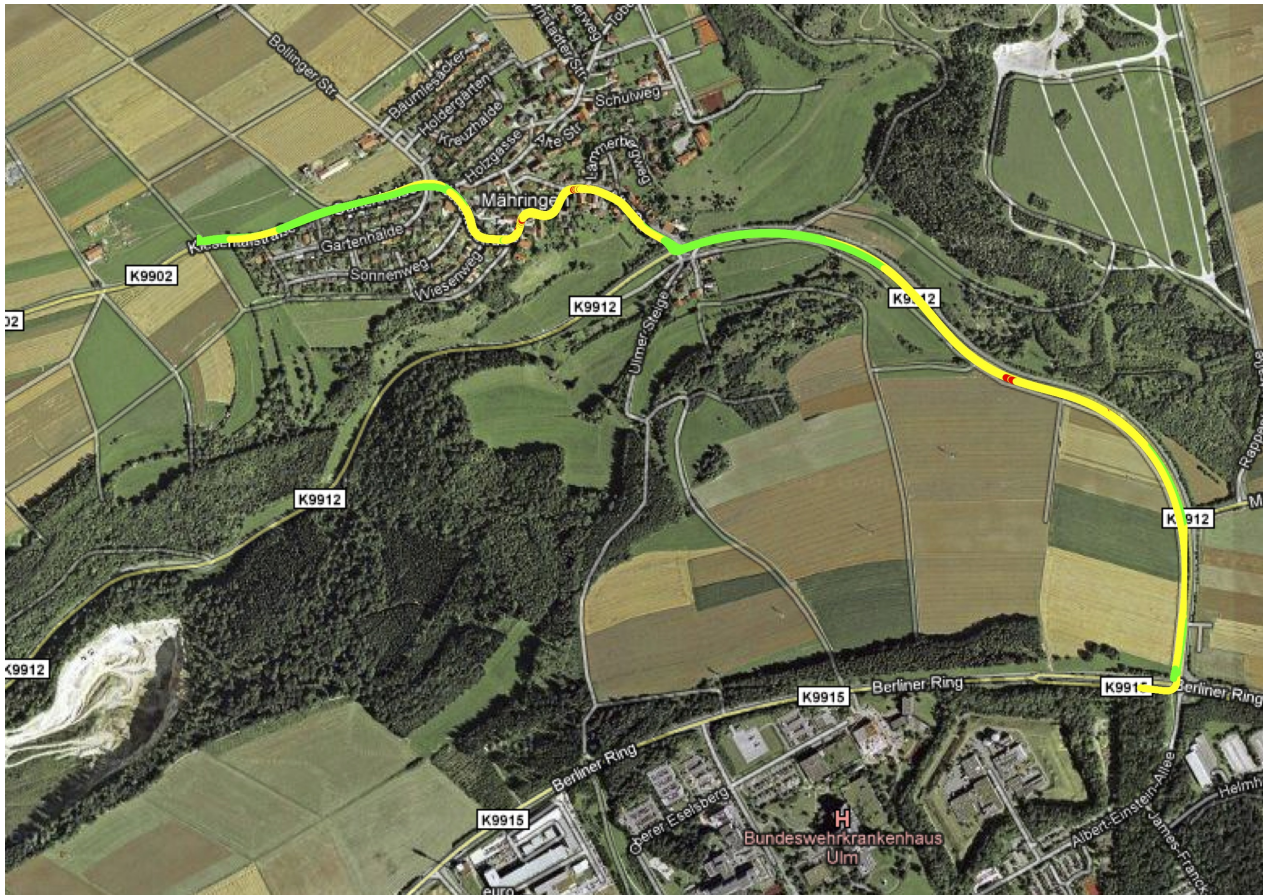


Abbildung 4.20 – Fahrt von der Sporthalle Nord nach Mähringen und zurück. Im Hintergrund ist das Satellitenbild zu sehen. Die Qualität der GPS-Positionen sind für jeden Punkt farblich kodiert.

Abbildung 4.20 zeigt die gefahrene Strecke. Im Hintergrund ist das Satellitenbild zu sehen. In Rot sind die Positionen eingezeichnet, in denen nur einfacher GPS-Empfang ohne Korrekturdaten möglich war. In Gelb sind die Positionen eingezeichnet, bei denen zumindest grober DGPS-Empfang vorhanden war und in Grün sind Positionen mit bestem DGPS-Empfang dargestellt. Abbildung 4.21 zeigt den vom ADMA geschätzten Fehler. Sind nicht genügend GPS-Satelliten sichtbar, so wächst der geschätzte Positionsfehler schnell auf 0,3 m an.

In Abbildung 4.22 ist das Höhenprofil der Sequenz dargestellt. Die maximale absolute Höhendifferenz beträgt 87 m.

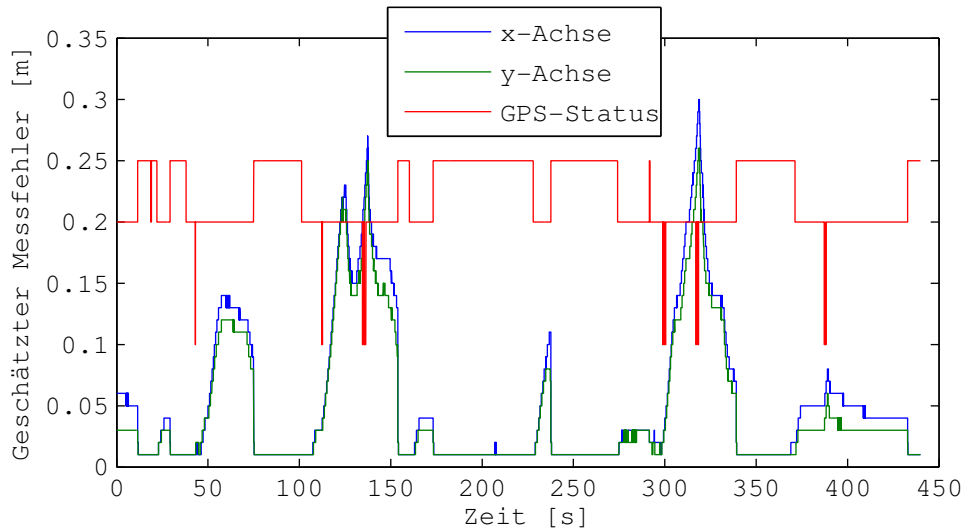


Abbildung 4.21 – Vom ADMA geschätzte Positionsfehler für die in Abbildung 4.20 dargestellte Sequenz. Mit Korrekturdaten im besten GPS-Status (0,25) liegt der geschätzte Fehler bei konstant 0,01 m. Mit etwas weniger Satelliten (GPS-Status 0,2) wächst der geschätzte Fehler bis auf etwa 0,1 m. Sind kurzzeitig noch weniger Satelliten sichtbar, fällt die Positionskorrektur weg (Status 0,1) und die geschätzten Fehler wachsen schnell auf bis zu 0,3 m.

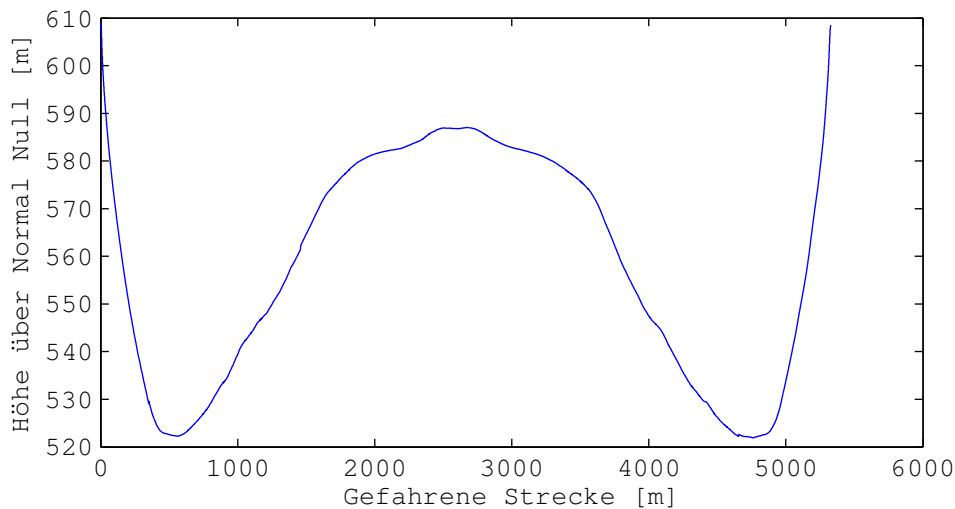


Abbildung 4.22 – Höhenprofil der Strecke. Das Profil ist nahezu symmetrisch, da derselbe Rückweg gewählt wurde. Der maximale Höhenunterschied beträgt 87 m.

Evaluierung der zweidimensionalen Selbstlokalisierung auf der Landstraße

Abbildung 4.23 zeigt die geschätzte Trajektorie der Odometrie und der zweidimensionalen Selbstlokalisierung im Vergleich zur Referenz. Durch die lange Sequenz von 5,2 km akkumulieren sich die Fehler in der Position stark auf. Durch die Selbstlokalisierung kann die Posenschätzung im Vergleich zur Odometrie deutlich verbessert werden.

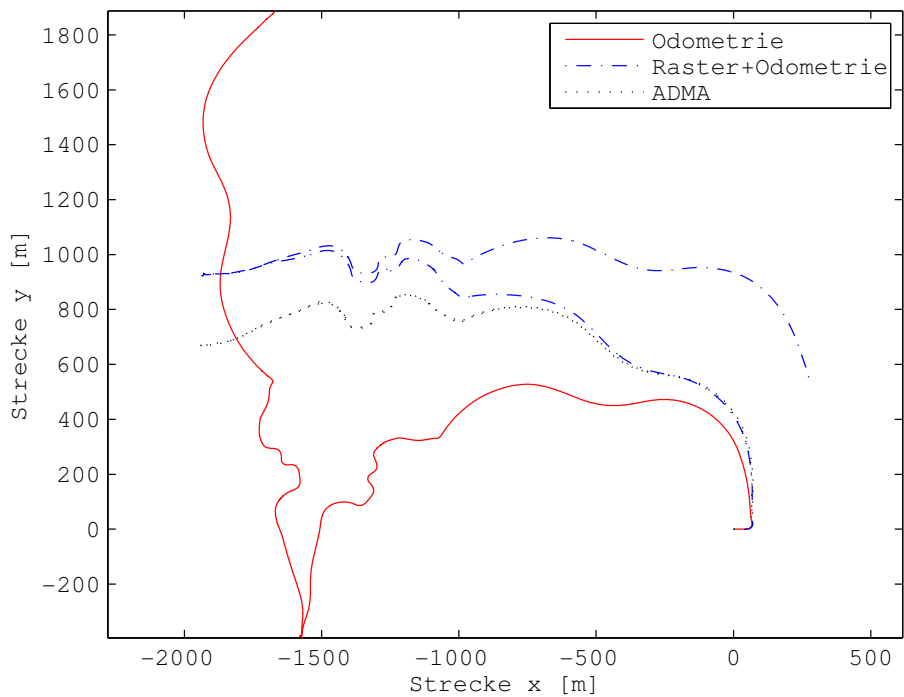


Abbildung 4.23 – Trajektorien der Selbstlokalisierung in Sequenz B. Es wird die rasterbasierte Selbstlokalisierung und die Odometrie mit der Referenz verglichen. Dadurch, dass die Sequenz die Hin- und Rückfahrt auf derselben Strecke enthält, sollte der Endpunkt wieder auf dem Startpunkt liegen. Startpunkt ist im Ursprung bei (0,0). Bei der Referenztrajektorie ist dies der Fall; bei der rasterbasierten Selbstlokalisierung ist schon eine deutliche Abweichung zu erkennen. Die rein auf der Odometrie basierende Trajektorie weicht noch viel stärker ab.

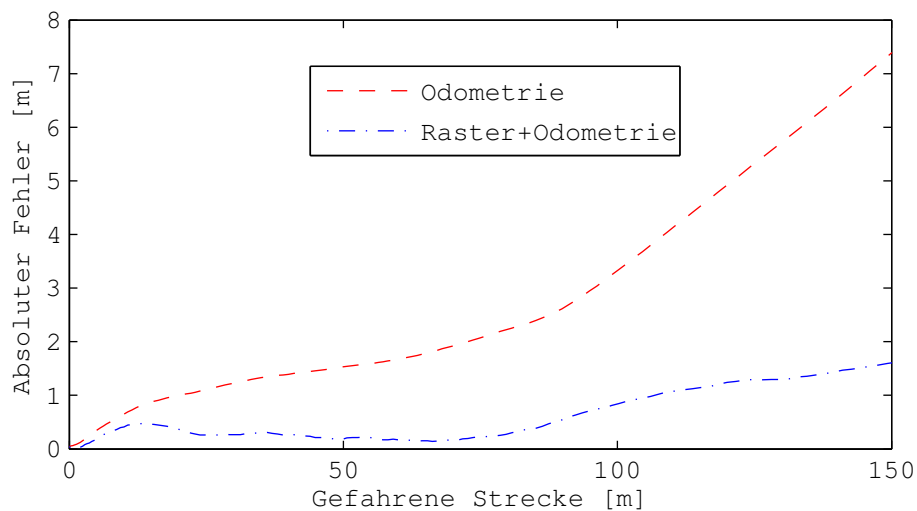


Abbildung 4.24 – Absoluter Fehler in der Positionsschätzung. Vergleich Odometrie und rasterbasierte Selbstlokalisierung mit Odometrie jeweils zur Referenz durch ADMA in zwei Dimensionen.

Der RMSE über die 2D-Position wird in Abbildung 4.24 dargestellt. Für die Evaluierung werden nur die Fehler betrachtet, die auf den ersten 100 m der Strecke auftreten. Dieses Szenario ist dafür geeignet, kurze Ausfälle des GPS zu kompensieren. Allein durch die Odometrie beträgt der absolute Fehler, der während den ersten 100 m Strecke entsteht, 3,35 m. Mit der Selbstlokalisierung kann dieser Wert deutlich reduziert werden auf 0,85 m.

Evaluierung der dreidimensionalen Selbstlokalisierung auf der Landstraße

Abbildung 4.25 zeigt die mittleren Fehler für die dreidimensionale Selbstlokalisierung. Im Dreidimensionalen kommen drei zusätzliche Freiheitsgrade hinzu. Der mittlere absolute Fehler, der durch die Odometrie entsteht, beträgt 4,86 m nach den ersten 100 gefahrenen Metern. Dieser Wert ist höher als in zwei Dimensionen, da durch die Odometrie die Höhendifferenzen nicht betrachtet werden. Mit der Selbstlokalisierung in drei Dimensionen kann dieser Wert auf 2,17 m reduziert werden.

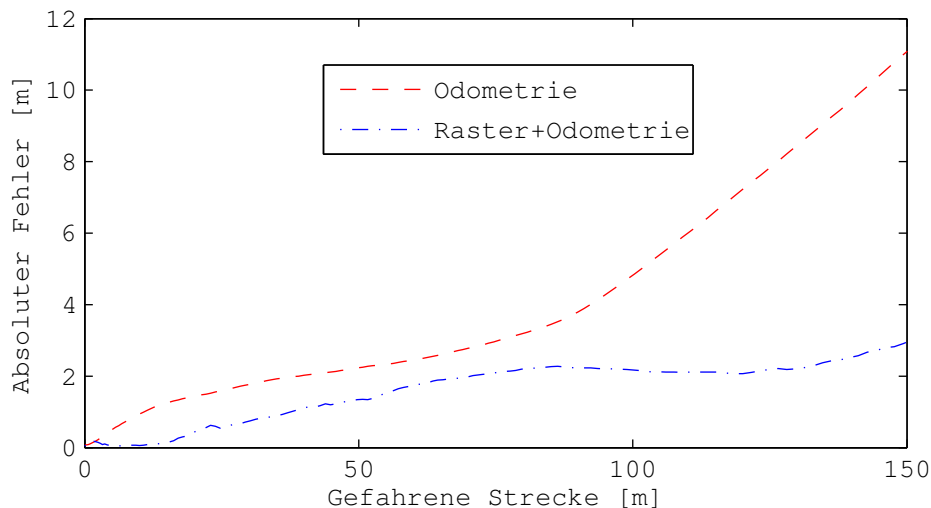


Abbildung 4.25 – Absoluter Fehler in der Positionsschätzung. Vergleich der Odometrie und der rasterbasierten Selbstlokalisierung mit Odometrie, jeweils zur Referenz durch den ADMA in drei Dimensionen.

4.7 Dreidimensionaler Kartenabgleich

Ein Kartenabgleich (engl. Map Matching) wird dazu verwendet, die Pose des Fahrzeuges durch die Registrierung einer Umgebungskarte mit einer digitalen Karte zu schätzen. Durch eine digitale Karte mit absoluten Ortsinformationen lässt sich somit auch die absolute Pose ermitteln. Dieses Verfahren wird bereits mit zweidimensionalen digitalen Karten eingesetzt, unter anderem in [75, 122, 109].

In letzter Zeit kommen jedoch auch digitale Karten mit Höheninformationen auf. Für die USA gibt es beispielsweise mit dem „National Elevation Dataset“ (NED) Höhendaten [53]. Google hat 2010 die „Elevation API“ [54] für Google Maps vorgestellt, die es ermöglicht Höhenprofile für bestimmte Routen abzufragen.

Ress et al. [101] betrachten Höheninformationen für Kartendaten speziell im Kontext Fahrerassistenzsysteme. Der ADAS-Horizont ist eine Vorausschau auf Streckendaten, die aus einer digitalen Karte stammen. Mittels der aktuellen Position und der Fahrtrichtung werden aus einer digitalen Karte alle relevanten Informationen wie Abzweigungen ermittelt, die sich auf der vorausliegenden Strecke befinden und auf einer Schnittstelle bereitgestellt. Dabei ist im Protokoll ADASIS Version 2 bereits die Übertragung von Höheninformationen vorgesehen.

Diese Höhendaten lassen in Verbindung mit einer dreidimensionalen Belegungskarte, wie sie in Kapitel 3 beschrieben wird, einen dreidimensionalen Kartenabgleich zu. Abbildung 4.26 zeigt ein Beispiel der dreidimensionalen Karte.

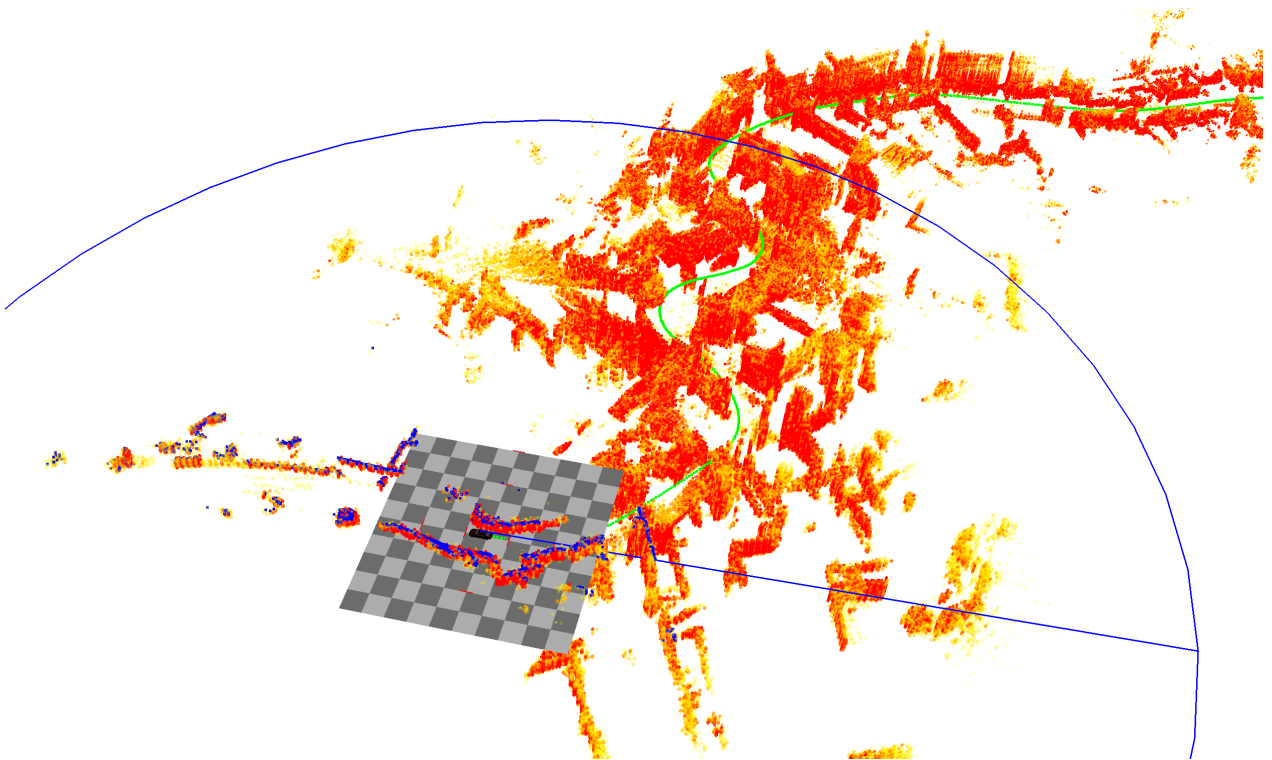


Abbildung 4.26 – Dreidimensionale Karte einer Fahrt durch Mähringen.

Aus einer Belegungskarte extrahierte Kurvenradien sind im Zweidimensionalen geeignete Merkmale, um eine Registrierung der Belegungskarte zu einer digitalen Karte herzustellen, siehe [75, 109]. Im Dreidimensionalen stehen neben den Kurvenradien zusätzlich Neigungswinkel und das Höhenprofil der Straße in Fahrtrichtung als Merkmale für die Registrierung zur Verfügung. Abbildung 4.27 zeigt eine Skizze des dreidimensionalen Kartenabgleichs über eine digitale Karte mit Höheninformationen.

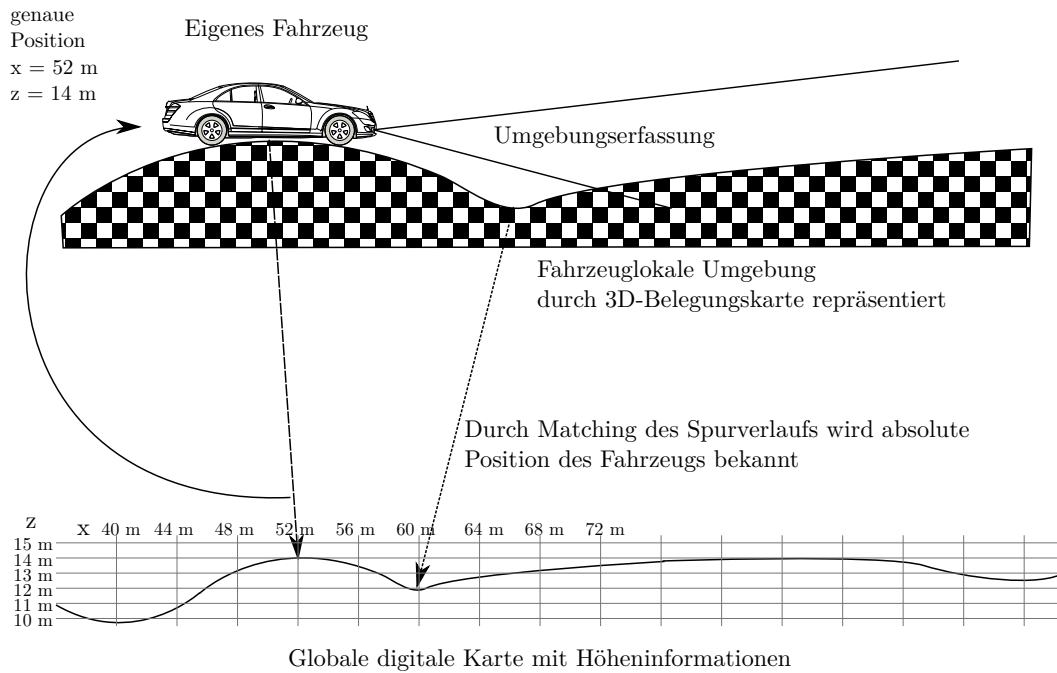


Abbildung 4.27 – Skizze des dreidimensionalen Kartenabgleichs. Aus einer dreidimensionalen Belegungskarte extrahierte Merkmale werden mit denen einer digitalen Karte abgeglichen. So kann zum einen die absolute Fahrzeugposition ermittelt werden, zum anderen sind aus der digitalen Karte Daten der bevorstehenden Strecke ablesbar.

Es ist davon auszugehen, dass durch die dreidimensionale Registrierung genauere Ergebnisse erzielt werden können, insbesondere auf Strecken, in denen wenige Kurven vorkommen. Das Verfahren wurde zum Patent angemeldet [150], es wurde im Rahmen dieser Arbeit jedoch nicht evaluiert.

4.8 Zusammenfassung und Ausblick

Zusammenfassung

In diesem Kapitel wurde eine rasterbasierte Selbstlokalisierung für die zwei- und dreidimensionale Kartierung untersucht. Die Methoden wurden implementiert und auf ihre Genauigkeit überprüft.

Die rasterbasierte Selbstlokalisierung arbeitet auf Basis der in Kapitel 3 vorgestellten hierarchischen Belegungskarte. In der zweidimensionalen Variante läuft die Selbstlokalisierung in Echtzeit. Für die in Abschnitt 4.6.3 beschriebene Sequenz lässt sich die Genauigkeit der Positionsschätzung deutlich verbessern. Die Odometrie liefert in einem städtischen lokal begrenzten Szenario einen mittleren Fehler von 2,24 m zur Referenz. Mit der rasterbasierten Selbstlokali-

sierung bei 0,2 m Auflösung wird eine deutliche Verbesserung auf eine Genauigkeit von 0,64 m erzielt. Auf der Landstraße stehen deutlich weniger Objekte zur Selbstlokalisierung zur Verfügung, die gefahrenen Manöver sind jedoch einfacher. Der Fehler der Odometrie nach 100 m beträgt 3,35 m, dieser konnte mit der Selbstlokalisierung auf 0,85 m reduziert werden.

Die dreidimensionale Selbstlokalisierung ist, durch die zusätzlichen Freiheitsgrade und die komplexere Karte, noch nicht echtzeitfähig. Im städtischen Szenario liefert die Odometrie eine Standardabweichung von 2,24 m, welche sich durch die rasterbasierte Selbstlokalisierung auf 1,5 m verbessern lässt. Auf der Landstraße nach den ersten 100 m beträgt der Fehler durch die Odometrie 4,86 m, welcher durch die Selbstlokalisierung auf 2,17 m reduziert werden kann. Der Fehler der dreidimensionalen Selbstlokalisierung ist durch die zusätzlichen drei Freiheitsgrade größer als der Fehler der zweidimensionalen Selbstlokalisierung.

Ausblick

- Zur weiteren Verbesserung der Selbstlokalisierung ist eine Kombination der vorgestellten rasterbasierten Lokalisierung mit einer punktlandmarkenbasierten Lokalisierung, wie von Weiss [139] beschrieben denkbar. Damit sollte die Genauigkeit in Situationen, in denen geeignete Punktmerkmale zur Lokalisierung vorhanden sind, weiter erhöht werden können.
- Der in Abschnitt 4.7 vorgeschlagene dreidimensionale Kartenabgleich wurde nicht evaluiert. In zwei Dimensionen wird dieser Ansatz bereits erfolgreich angewandt [75, 109]. In drei Dimensionen stehen für den Abgleich mehr Merkmale zur Verfügung, von daher sollte eine Registrierung genauer erfolgen können. Die Skizzierung dieses Ansatzes stellt eine Basis für zukünftige Realisierungen und Auswertungen dar. Eine große Herausforderung wird dabei die Extraktion des dreidimensionalen Fahrbahnverlaufs aus der Belegungskarte sein.
- Der ADMA ist ein kostspieliges Referenzgerät und steht deshalb nicht in jedem Versuchsträger zur Verfügung. Einen Hinweis auf die Güte eines SLAM-Algorithmus ohne absolute Referenz durch eine IMU bekommt man durch die Schärfe der resultierenden Belegungskarte. Je schlechter die Posenschätzung ist, desto stärker verwischen Objekte in der Karte. Ein Verfahren zur Bewertung von Lokalisierungsalgorithmen basierend auf der Schärfe der resultierenden Belegungskarten wäre daher von Interesse.
- In Abschnitt 4.7 wurde ein neues Verfahren vorgestellt, um einen dreidimensionalen Kartenabgleich durchzuführen. Das Verfahren profitiert von einer dreidimensionalen Belegungskarte der Umgebung. Des Weiteren wird eine digitale Karte mit Höheninformationen vorausgesetzt, wie sie beispielsweise über das ADASIS-Protokoll zur Verfügung gestellt werden kann. Das Verfahren wurde zum Patent angemeldet [150]. Die

Erhöhung der Genauigkeit der Lokalisierung mit diesem Verfahren bleibt mit einer Evaluierung zu zeigen.

5

Detektion und Verfolgung bewegter Objekte

In diesem Kapitel wird ein neues Verfahren vorgestellt, um bewegte Objekte mittels einer Belegungskarte zu detektieren und deren Zustand zu schätzen. In der Literatur sind ähnliche Verfahren unter dem Begriff „DATMO“ (engl. Detection and Tracking of Moving Objects) bekannt.

5.1 Motivation

Viele Fahrerassistenz- und Sicherheitsfunktionen basieren auf den Dynamikdaten von Verkehrsteilnehmern in der Umgebung. Beispiele hierfür sind der Abstandsregeltempomat, eine PreCrash-Funktion oder ein Spurwechselassistent. Eine zuverlässige Detektion und Zustandsschätzung bewegter Verkehrsteilnehmer ist daher essentiell.

Belegungskarten sind sehr gut geeignet, um eine Freiraumanalyse durchzuführen, die beispielsweise für einen Parkhilfeassistenten benötigt wird oder für ein Sicherheitssystem, das mögliche Ausweichtrajektorien bestimmen muss. Bei Belegungskarten wird im Allgemeinen die Grundannahme getroffen, dass ausschließlich Messungen von statischen Objekten integriert werden, insbesondere bei der in Kapitel 3 eingeführten Belegungskarte. Wird diese Annahme nicht erfüllt, so entstehen Artefakte in der Karte, die von bewegten Objekten stammen. Diese Artefakte sind in der Regel unerwünscht und können eine zuverlässige Frei-

raumanalyse unmöglich machen. Im automobilen Umfeld ist die Annahme einer statischen Umgebung nur selten erfüllt. Bewegte Objekte müssen daher detektiert und bei der Erstellung einer statischen Belegungskarte ausgeschlossen werden.

Es gibt Verfahren, um bewegte Objekte in Belegungskarten zu detektieren und damit die beiden geschilderten Problemstellungen auf einmal zu lösen: Die Detektion und Verfolgung bewegter Objekte (DATMO) in Belegungskarten. Diese Verfahren haben den Vorteil, dass sowohl der Zustand anderer Verkehrsteilnehmer geschätzt, als auch eine korrekte statische Karte für eine Freiraumanalyse aufgebaut werden kann.

Zielsetzung

Dieses Kapitel verfolgt zwei Ziele: Eine artefaktfreie statische Karte zu generieren sowie den Dynamikzustand anderer Verkehrsteilnehmer zu schätzen. Artefakte in der statischen Karte sollen unterdrückt werden, indem bewegte Objekte detektiert werden und verhindert wird, dass Messungen von bewegten Objekten in die statische Karte gelangen.

Die Detektion bewegter Objekte soll dabei nicht für mögliche Fehler in der Segmentierung von Sensordaten anfällig sein. Des Weiteren soll die Verfolgung modellfrei erfolgen, so dass Fehler in der Modellbildung und in der Merkmalsextraktion ausgeschlossen werden. Die Verfolgung soll deshalb auf einer niedrigen Abstraktionsstufe stattfinden, noch bevor eine sichere Detektion eines bewegten Objektes oder eine Segmentierung der Sensordaten stattgefunden hat. Mit diesem „Track-before-detect“-Ansatz können Objektgeschwindigkeiten geschätzt werden, ohne dass Vorkenntnisse über Objektgröße, -form und Fahrtrichtung notwendig sind.

Kapitelübersicht

Im nächsten Abschnitt 5.2 folgt der Stand der Technik zur Detektion und Verfolgung bewegter Objekte in Belegungskarten. In Abschnitt 5.3 wird ein Ansatz zur Detektion inkonsistenter Zellen in dreidimensionalen und hierarchischen Belegungskarten eingeführt, die in der Regel von bewegten Objekten stammen. In Abschnitt 5.4 werden Ansätze zur Verfolgung der detektierten Zellen und die Objektbildung beschrieben. Eine Evaluierung mit realen Messdaten und Referenzinformationen erfolgt in Abschnitt 5.5. Das Kapitel endet mit einer Zusammenfassung und einem Ausblick in Abschnitt 5.6.

5.2 Stand der Technik

Das motivierte Problem der Detektion bewegter Objekte ist gelöst, wenn jedem Messpunkt entweder die Klasse „stammt von bewegtem Objekt“ (DYN) oder die Klasse „stammt von statischem Objekt“ (STAT) zugeordnet werden kann. Dann müssen lediglich Messungen aus der Klasse DYN in nachfolgenden Algorithmen zur Zustandsschätzung betrachtet werden, während ausschließlich Messungen aus der Klasse STAT in die Belegungskarte integriert werden.

Eine solche Klassifikation ist allerdings nicht ohne weiteres zu erreichen. Mit einem Laserscanner ist diese in folgendem Extremfall sogar unmöglich: Ein langer Bus fährt mit derselben Geschwindigkeit wie das Sensorfahrzeug neben diesem her. In den Messdaten ist diese Situation nicht von einer Schutzplanke am Fahrbahnrand zu unterscheiden, dennoch haben beide unterschiedliche Geschwindigkeiten. Im Folgenden wird zunächst auf Lidar-Sensoren eingegangen, da diese Geschwindigkeiten nicht direkt messen können, dafür jedoch eine deutlich höhere Auflösung besitzen als andere aktive Sensoren. Anschließend wird auf weitere Sensoren im automobilen Kontext eingegangen.

Für Lidar-Sensoren existieren in der Literatur im Wesentlichen zwei unterschiedliche Vorgehensweisen: Die objektbasierte Verfolgung und die rasterbasierte Verfolgung. Zunächst wird die objektbasierte Verfolgung beschrieben. Der Übersichtlichkeit halber wird die rasterbasierte Verfolgung bewegter Objekte in zwei Abschnitte unterteilt: Zuerst werden Arbeiten zur Detektion bewegter Objekte in Belegungskarten beschrieben, anschließend Arbeiten zur Verfolgung bewegter Objekte aus Belegungskarten.

5.2.1 Objektbasierte Verfolgung mit Lidar-Sensoren

Bei der objektbasierten Verfolgung von Verkehrsteilnehmern wird von den Rohdaten des Laserscanners schrittweise abstrahiert. Die einzelnen Schritte sind die Segmentierung, die Merkmalsextraktion und das Aufstellen einer Objekthypothese. Jedes hypothetische Objekt wird anschließend verfolgt, siehe hierzu beispielsweise Skutek [112, 100]. Abbildung 5.1 skizziert den Ablauf. Eine naheliegende Lösung ist, über die Geschwindigkeit der resultierenden verfolgten Objekte eine Klassifikation in die Klassen STAT oder DYN durchzuführen. Alle Messungen, die zu Objekten ohne nennenswerte Geschwindigkeit gehören, können demnach in eine Belegungskarte zur Freiraumanalyse integriert werden. Diese Lösung hat mehrere Nachteile: Die Segmentierung basiert auf mehreren Schwellwerten, die auf einen Sensor und eine Applikation abgestimmt werden müssen. Eine Änderung der Bedingungen führt schnell zu falsch segmentierten Daten, so dass zwangsläufig auch bei nachfolgenden Schritten Fehler auftreten. Weiterhin werden alle Segmente verfolgt, auch diejenigen von nicht bewegten

Objekten, was unnötig Rechenzeit kostet.

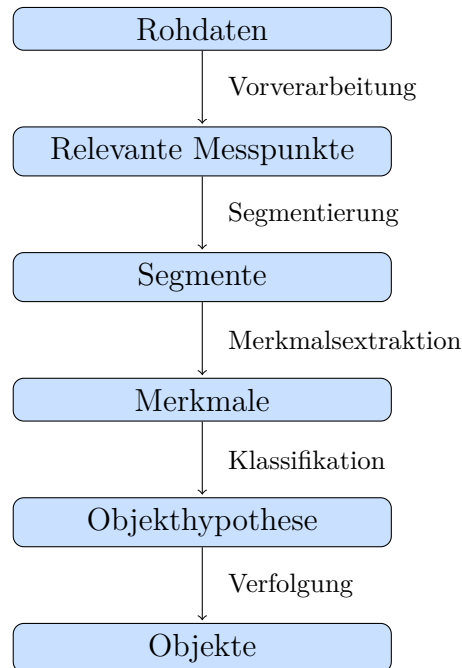


Abbildung 5.1 – Verarbeitung von Laserscannerdaten für die objektbasierte Verfolgung.

5.2.2 Detektion bewegter Objekte in Belegungskarten mit Lidar-Sensoren

Wang [136] stellt ein Konzept vor, um SLAM und DATMO zu kombinieren. Es werden zeitlich aufeinanderfolgende Messungen betrachtet und deren Konsistenz analysiert. Die Abstandsänderung eines Messpunktes pro Winkelschritt in Polarkoordinaten lässt auf bewegte Objekte schließen, wenn die Eigenbewegung bereits kompensiert wurde. Beinhaltet ein Segment mehr als 50% dynamische Messungen, wird das ganze Segment als dynamisch klassifiziert. Fußgänger sind meist zu langsam, um durch diesen Ansatz detektiert zu werden. Deshalb wird in der Arbeit eine spezielle Karte vorgeschlagen, um dieses Problem zu lösen: Eine Karte, die lediglich bewegte Objekte enthält. Befindet sich ein Objekt in einem Bereich, der bereits von einem bewegten Objekt durchquert wurde, so muss es sich dabei um ein dynamisches Objekt handeln.

Vu et al. [135] gehen analog vor. Messpunkte werden segmentiert, indem Punkte mit einem Abstand kleiner als 0,3 m zusammengefasst werden. Eine separate dynamische Karte enthält die Information, wie viele bewegte Objekte jeweils eine Zelle durchquert haben. So lassen sich zukünftige Messungen schneller klassifizieren.

Weiss [141] detektiert bewegte Objekte in Belegungskarten über deren „Schweif“. Ein Objekt, das sich radial zum Laserscanner bewegt, hinterlässt in der Karte eine Reihe von Spuren. Diese sind charakteristisch für dynamische Objekte und lassen sich mit Methoden der Bildverarbeitung detektieren. Über den Abstand der einzelnen Spuren lässt sich die Objektgeschwindigkeit ermitteln. Die Anwendbarkeit dieser Methode auf querenden Verkehr wurde noch nicht gezeigt.

Petrovskaya und Thrun [97] verwenden einen virtuellen zweidimensionalen Scan, um Messungen zu modellieren. Die Umgebung des Fahrzeuges wird rundum in $0,5^\circ$ große Sektoren eingeteilt. Für jeden virtuellen „Strahl“ wird der Bereich bis zum nächsten Objekt als frei betrachtet, der Bereich hinter dem nächsten Objekt hingegen als verdeckt. Als Hauptmerkmal bei der Detektion eines bewegten Objektes wird der Bereich genutzt, der hinter einem bewegten Objekt frei wird. Zur Detektion sind drei Zeitschritte notwendig: Mit zwei Zeitschritten wird generell auf ein bewegtes Objekt geschlossen, der dritte Zeitschritt wird zur Plausibilisierung genutzt. Schwarze Objekte absorbieren in großen Teilen die Lichtpulse des Laserscanners und liefern deshalb typischerweise wenige Messungen. Diese werden in einem 3D-Scan detektiert: Die Abwesenheit von Bodenmessungen lässt auf ein schwarzes Objekt schließen, das Lichtpulse absorbiert. Die Distanz zum schwarzen Objekt entspricht somit in etwa dem Abstand der letzten Bodenmessung in dieser Richtung. Das Verfahren funktioniert, so lange genügend Bodenmessungen vorhanden sind. Mit dem Velodyne beträgt die Reichweite 25-30 m.

Bouzouraa und Hofmann [10, 11] identifizieren dynamische Zellen durch eine Differenzbildung von aktueller Messung und bestehender Belegungskarte. Diese werden anschließend segmentiert. Um die Segmente wird ein umschreibender Quader gelegt, der die Ausmaße und Orientierung eines Objektes modelliert.

Konrad und Dietmayer [76] beschreiben zwei Ansätze zur Detektion bewegter Objekte in 2D-Karten: Der erste Ansatz ist die verzögerte Kartografierung, bei der lediglich Messungen in die Karte eingetragen werden, die entweder auf eine belegte Zelle fallen oder mit einer Messung aus dem vorherigen Zeitschritt korrespondieren. Messungen, die erstmalig auftreten, werden nicht berücksichtigt. Der zweite beschriebene Ansatz ist die rasterbasierte Verfolgung. Hierzu wird für jede Messung des Laserscanners eine temporäre Belegungskarte aufgestellt, die sogenannte „Messkarte“. Aus den Messkarten zweier aufeinanderfolgender Zeitschritte kann nun eine Differenzkarte bestimmt werden. Zellen der Differenzkarte werden anschließend binarisiert und segmentiert. Für jedes Segment wird ein umschließendes Rechteck ermittelt, das eine Objekthypothese darstellt.

Fulgenzi et al. [50] erweitern den Belegungskartenansatz um die Modellierung von Objektgeschwindigkeiten. Dazu erhalten einzelne Zellen der Belegungskarte Geschwindigkeitsvektoren. Bewegte Zellen werden über die Zeit durch die Karte propagiert, um zu bestimmen, ob es zu

einer Kollision kommen wird.

Richter et al. benutzen eine 2D-Karte und berechnen Geschwindigkeitsvektoren für jede Zelle [102]. Des Weiteren untersuchen sie eine vierdimensionale Karte, die neben den beiden Raumdimensionen zwei Dimensionen für die kartesische Geschwindigkeit umfasst. Hier entsteht auch für die Geschwindigkeitsschätzung eine Diskretisierung.

Moras [87] detektiert bewegte Objekte in DST-basierenden Belegungskarten. Eine Zelle mit hoher Evidenz für „frei“ erhält eine Evidenz für „belegt“, wenn sich ein Objekt hineinbewegt. Diese Inkonsistenz wird genutzt, um bewegte Objekte zu identifizieren.

5.2.3 Detektion bewegter Objekte mit weiteren Sensoren

Auch mit Radarsensoren und Stereokameras können bewegte Objekte detektiert werden. In den folgenden Abschnitten wird darauf kurz eingegangen.

Radarsensoren

Radarsensoren haben in der Regel deutlich höhere Messunsicherheiten als Lidar-Sensoren. Weiterhin existieren auf einem PKW oder LKW mehrere Radarstreuzentren. Das bedeutet, dass die Unsicherheit eines Messpunktes dadurch erhöht wird, dass in mehreren Auswertungsschritten unterschiedliche Punkte auf einem Fahrzeug angemessen werden, wie zum Beispiel das Kennzeichen, die Radkästen oder die Felgen.

Radarsensoren messen in der Regel über den Dopplereffekt die Radialgeschwindigkeit v_{rad} von Objekten. Messungen mit $v_{\text{rad}} > 0$ können direkt der Klasse DYN zugeordnet werden. Eine Radialgeschwindigkeit von $v_{\text{rad}} = 0$ ist eine notwendige Bedingung für die Klasse STAT, aber keine hinreichende: Objekte, die beispielsweise an Kreuzungen quer zum eigenen Fahrzeug fahren haben ebenfalls eine Radialgeschwindigkeit $v_{\text{rad}} = 0$.

Die meisten Radarsensoren bieten darüber hinaus direkt eine Objektschnittstelle, bei der Rohdatenziele bereits durch ein Filter wie das Kalman-Filter verfolgt werden. Ein Beispiel ist der in Abschnitt 2.4.1 beschriebene Radarsensor. In dieser Arbeit wird die Detektion bewegter Objekte durch Radarsensoren in Belegungskarten deshalb nicht betrachtet.

Stereokamera

Die Detektion bewegter Objekte basierend auf Stereokameradaten wird in dieser Arbeit ebenfalls nicht betrachtet. Im Spezialfall der Stixel-Repräsentation sei hierfür auf Erbs [38] ver-

wiesen.

5.2.4 Verfolgung bewegter Objekte in Belegungskarten

In diesem Abschnitt wird ein Überblick über Methoden zur Schätzung des Zustandes der detektierten Objekte gegeben.

Kalman-Filter

Wang [136] verfolgt bewegte Objekte mit einem IMM (engl. Interacting Multiple Model)-Filter. Dazu verwendet er das CA (engl. Constant Acceleration)- und das CV (engl. Constant Velocity)-Modell für bewegte Objekte.

Effertz [30] beschreibt eine Kombination von Objekten mit einer Belegungskarte. Objekte werden durch lokale Objektkarten repräsentiert, um eine beliebige Objektkontur abbilden zu können. Zur Vereinfachung wird ein umhüllender Polygonzug um belegte Zellen definiert. Bei der Datenassoziation werden die Linien des Polygonzugs mit berücksichtigt. Liefert ein Sensor lediglich eine Punktmessung, wird diese auf den Polygonzug projiziert. Die Verfolgung erfolgt durch ein Kalman-Filter pro Stützstelle des Polygonzugs.

Bouzouraa und Hofmann [10, 11] verfolgen bewegte Objekte durch ein Kalman-Filter. Als Assoziationsverfahren wird das GNN (Globaler Nächster Nachbar)-Verfahren eingesetzt. Der Zustandsvektor enthält die aus dem Quader-Modell für ein bewegtes Objekt ermittelte Pose und Ausmaße, sowie die Geschwindigkeit und die Beschleunigung des Objektes. Es wird das CA-Modell mit unabhängiger Beschleunigung für die einzelnen Koordinatenachsen eingesetzt. Zusätzlich werden für jedes Objekt die zugehörigen Zellen aus der Belegungskarte mitgeführt, so dass sich die Kontur von Objekten akkumuliert.

Konrad [76] schätzt mit einem Kalman-Filter und einem CV-Bewegungsmodell die Mittelpunkte von rechteckigen Objekthypothesen. Zur Evaluierung werden bewegte Zellen manuell markiert und die Detektionsleistung der Verfahren damit verglichen.

Monte-Carlo-Methoden

Vu et al. [134] verwenden den Metropolis-Hastings-Algorithmus [57] zur Verfolgung bewegter Objekte. Tracks ω werden so gesucht, dass die Wahrscheinlichkeit $p(\omega|\mathbf{y})$ maximal ist. Dabei bezeichnet \mathbf{y} die Messungen aus den n letzten Zeitschritten. Es werden vier Objektklassen unterschieden: Bus, PKW, Fahrrad und Fußgänger. Für Fußgänger wird ein Punktmodell angenommen, für die anderen Objektklassen ein Box-Modell. Diese Modelle werden in die

segmentierten Messungen des Laserscanners eingepasst. Es werden mehrere Bewegungsmodelle in einem IMM-Ansatz kombiniert: CV, CA, CT und statisch.

Petrovskaya und Thrun [97] schlagen einen modellbasierten Ansatz vor. Das Modell ist auf rechteckige Objekte zugeschnitten. Als Bewegungsmodell wird das CV-Modell verwendet. Die Objektverfolgung erfolgt durch ein Rao-Blackwellized Partikel-Filter, wobei die Pose und die Geschwindigkeit des Objektes durch Monte-Carlo-Methoden geschätzt werden und die Objektdimensionen in jedem Partikel durch ein Kalman-Filter geschätzt werden.

5.3 Detektion bewegter Objekte

5.3.1 Überblick

Ein Laserscanner, wie in Abschnitt 2.4.2 beschrieben, misst lediglich Distanzen, eine Objektgeschwindigkeit kann nicht direkt gemessen werden. Über die Konsistenz aufeinanderfolgender Messungen eines Laserscanners kann jedoch auf bewegte Objekte in der Umgebung geschlossen werden. Bewegte Objekte führen zu Änderungen im Distanzprofil des Laserscanners. Diese Änderungen werden in dieser Arbeit über den Ansatz einer Karte mit dynamischen Zellen festgehalten.

Um Änderungen des Distanzprofils durch die Eigenbewegung möglichst auszuschließen, werden Messungen des Laserscanners stets im globalen Kartenkoordinatensystem KK betrachtet. Die aktuelle Pose des Fahrzeugs mit dem Sensor wird durch das in Kapitel 4 vorgestellte Verfahren zur Selbstlokalisierung möglichst genau ermittelt. Anschließend werden mehrere neue Karten definiert, um die statische Karte frei von bewegten Objekten zu halten und separat bewegte Objekte verfolgen zu können: Die statische Karte K_{static} , die Messkarte K_{meas} und die Karte mit dynamischen Messungen K_{dyn} . Der Aufbau dieser Karten wird in den folgenden Abschnitten beschrieben.

5.3.2 Die Messkarte

Zur Erstellung der Messkarte K_{meas} werden dieselben Algorithmen genutzt, wie in Kapitel 3 eingeführt. Es wird jedoch zu jedem Zeitschritt lediglich die neueste Messung in die Messkarte integriert. Für jeden Zeitschritt k entsteht somit eine neue Messkarte K_{meas}^k . Die Messkarte wird nur in zwei Dimensionen betrachtet, eine dreidimensionale Messung wird deshalb auf eine zweidimensionale Ebene projiziert und in die zweidimensionale Karte integriert. Bodenmessungen aus dem dreidimensionalen Distanzprofil müssen vorher jedoch entfernt werden. Bei ebener Fahrbahn kann dazu mit dem Ransac-Algorithmus eine Ebene in die Laserscan-

nerdaten eingepasst werden. Alle Punkte, die sich innerhalb einer Fehlerschranke um diese Ebene befinden, werden dann entfernt. Für eine unebene Fahrbahn ist ein aufwändigerer Algorithmus notwendig, wie beispielsweise von Himmelsbach [59] beschrieben.

Abbildung 5.2 zeigt zwei Beispiele von Messkarten. In 5.2a wird lediglich eine Ebene des Velodyne Laserscanners verwendet, um die Messkarte aufzubauen. In 5.2b hingegen werden alle 64 Ebenen des Laserscanners auf eine zweidimensionale Ebene projiziert und in die Messkarte integriert.

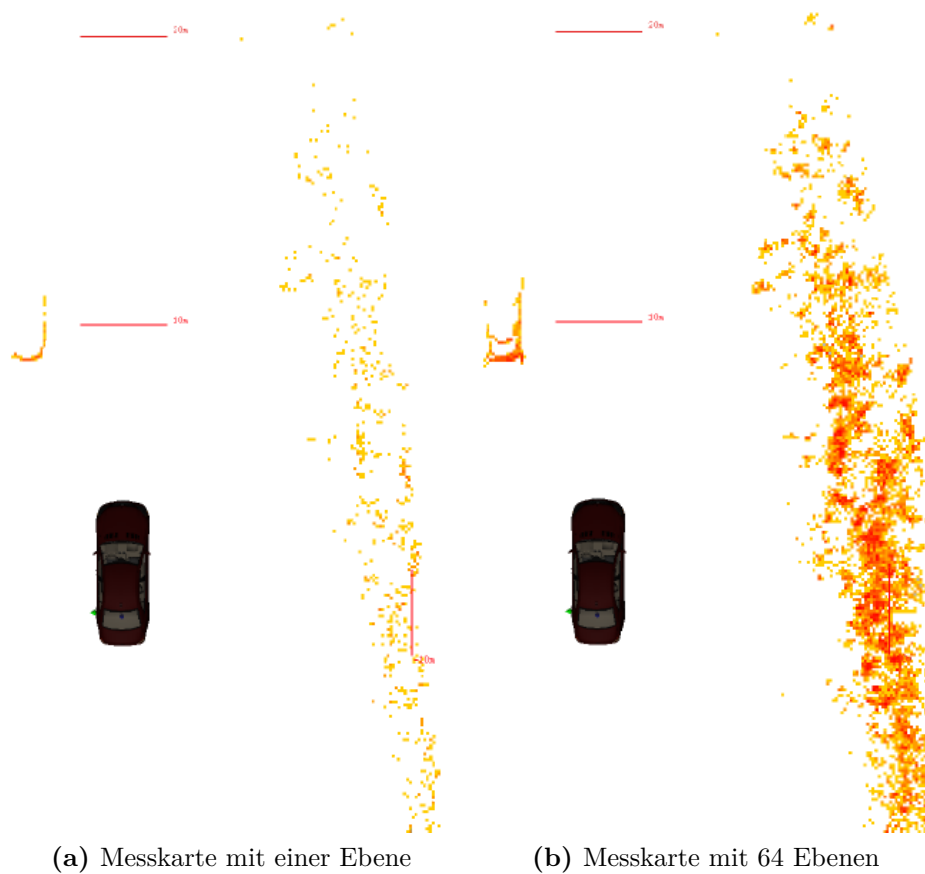


Abbildung 5.2 – Messkarten K_{meas} mit einer unterschiedlichen Anzahl an betrachteten Ebenen des Laserscanners.

5.3.3 Die statische Belegungskarte

Die in Kapitel 3 eingeführte Belegungskarte zur Abbildung statischer Objekte aus der Umgebung wird nachfolgend mit K_{static} bezeichnet. Diese wird dahingehend modifiziert, dass lediglich Zellen, die in mehreren aufeinanderfolgenden Zeitschritten belegt sind, integriert werden. Zur Bestimmung der wiederholt belegten Zellen wird die Schnittmenge aus zeitlich aufeinanderfolgenden Messkarten bestimmt: $K_{\text{meas}}^k \cap K_{\text{meas}}^{k-1}$. Dazu werden die in Abschnitt 3.7.5

definierten Mengenoperationen auf Belegungskarten genutzt. Die statische Belegungskarte K_{static} zum Zeitpunkt k kann demnach rekursiv wie folgt definiert werden:

$$K_{\text{static}}^k = K_{\text{static}}^{k-1} \cup \left(K_{\text{meas}}^k \cap K_{\text{meas}}^{k-1} \right) \quad (5.1)$$

Die statische Belegungskarte zu Beginn K_{static}^0 wird als leer angenommen. Anschließend wird die statische Karte rekursiv um die Zellen erweitert, die in den beiden letzten Messkarten auftraten. Damit akkumulieren sich belegte Bereiche in der statischen Karte ohne großen Einfluss von dynamischen Objekten.

Aufgrund von Fehlmessungen oder Witterungseinflüssen wie Regen werden statische Objekte möglicherweise jedoch nicht in jedem Zeitschritt beobachtet. Um statische Objekte mit kleiner Ausdehnung nicht zu übersehen wird die Vorgehensweise erweitert. Es fließen deshalb die drei letzten Messkarten in die Aktualisierung der statischen Karte ein:

$$K_{\text{static}}^k = K_{\text{static}}^{k-1} \cup \left(K_{\text{meas}}^k \cap K_{\text{meas}}^{k-1} \right) \cup \left(K_{\text{meas}}^k \cap K_{\text{meas}}^{k-2} \right) \quad (5.2)$$

Andererseits kommt es vor, dass nicht alle dynamischen Objekte aus der statischen Karte entfernt werden. Dies liegt daran, dass sich ein Teil der Längsseite eines PKW oder LKW in mehreren Zeitschritten über dieselben Zellen der Belegungskarte bewegt. Diese Zellen werden fälschlicherweise zunächst auch in die statische Belegungskarte integriert. In dem später folgenden Abschnitt 5.4.6 wird eine Methode vorgestellt, die dieses Problem löst.

5.3.4 Belegungskarte mit dynamischen Messungen

Die Karte mit dynamischen Messungen K_{dyn} wird aus der aktuellen statischen Karte K_{static}^k sowie der Messkarte K_{meas}^k aufgebaut. Dazu werden ebenfalls die in Abschnitt 3.7.5 definierten Mengenoperationen auf Belegungskarten genutzt.

Durch die Subtraktion der statischen Karte von der aktuellen Messkarte bleiben bewegte Objekte übrig:

$$K_{\text{dyn}}^k = K_{\text{meas}}^k - K_{\text{static}}^k \quad (5.3)$$

Bei dieser Lösung bleiben jedoch statische Objekte unberücksichtigt, die neu in der Messkarte enthalten sind: Diese bleiben in der dynamischen Karte erhalten. Dies tritt beispielsweise regelmäßig bei Schutzplanken auf, die erstmalig in den Sichtbereich geraten.

Dieser Effekt kann vermieden werden, wenn auf die nächste noch aktuellere Messung gewartet wird:

$$\mathbf{K}_{\text{dyn}}^{k-1} = \left\{ \mathbf{K}_{\text{meas}}^{k-1} - \mathbf{K}_{\text{static}}^k \right\} - \mathbf{K}_{\text{meas}}^k \quad (5.4)$$

Diese Lösung hat zum Nachteil, dass bewegte Objekte erst mit einer Verzögerung von einem Zeitschritt detektiert werden, was zu Gunsten der robusteren Detektion jedoch in Kauf genommen wird.

Ein weiteres Problem für eine hoch aufgelöste Belegungskarte stellen feingliedrige Objekte wie Äste oder Zäune dar, die nicht aus jeder Perspektive in jedem Winkelschritt gesehen werden. Über die vorgestellte Methode zur Erstellung einer dynamischen Belegungskarte werden diese als mögliche bewegte Objekte detektiert und stellen somit Rauschen dar. In dieselbe Kategorie fallen sich im Wind bewegende Objekte wie Blätter: Diese sind genau genommen zwar beweglich, sollen aber dennoch nicht als bewegte Objekte verfolgt werden und sind in der dynamischen Karte unerwünscht.

Um den Einfluss von Rauschen zu minimieren wird deshalb ein Operator „dil“ (von engl. dilate: dehnen, aufweiten) definiert, der belegte Zellen in der statische Karte aufweitet. Diese Vorgehensweise ist legitim, wenn davon ausgegangen wird, dass bewegte und statische Objekte einen gewissen Mindestabstand nicht unterschreiten. Aus der Bildverarbeitung sind Verfahren bekannt, die in einem Octree alle Nachbarknoten zu einem gegebenen Knoten bestimmen. Da die Octrees jedoch in einem dünn besetzten Voxelgitter angeordnet sind, treten an den Rändern der Octree-Wurzelknoten Sonderbedingungen auf. Um diese zu umgehen, werden zur Umsetzung der dil-Funktion nicht direkt die benachbarten Zellen bestimmt, sondern die in Kapitel 3.7.3 beschriebene Methode zur Integration von Messungen herangezogen. Mit dieser wird für jede belegte Zelle eine Kugel mit einem definierten Radius integriert. Die resultierende Gleichung zur Erstellung der dynamischen Karte lautet damit:

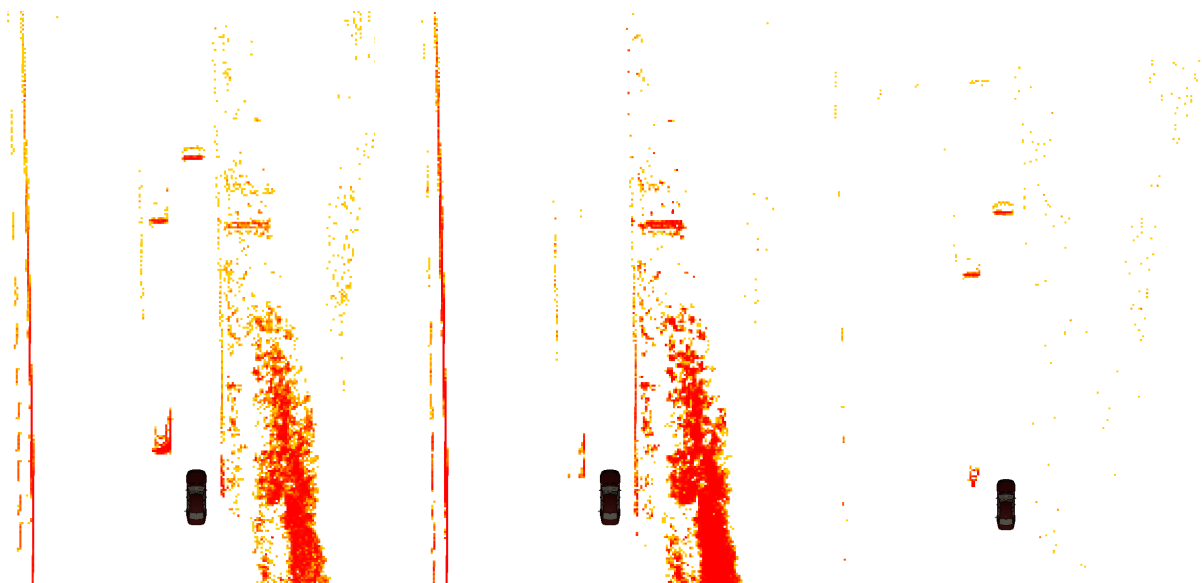
$$\mathbf{K}_{\text{dyn}}^{k-1} = \left\{ \mathbf{K}_{\text{meas}}^{k-1} - \text{dil} \left(\mathbf{K}_{\text{static}}^k \right) \right\} - \mathbf{K}_{\text{meas}}^k \quad (5.5)$$

Abbildung 5.3 zeigt exemplarisch das Resultat der Detektion bewegter Objekte. In Abbildung 5.3b ist die aktuelle Messkarte \mathbf{K}_{meas} mit allen Messungen des Velodyne Laserscanners aus 64 Ebenen dargestellt. Auf Basis der Messkarten wird die statische Karte $\mathbf{K}_{\text{static}}$ akkumuliert, siehe Abbildung 5.3c. Abbildung 5.3d zeigt schließlich die resultierende dynamische Karte \mathbf{K}_{dyn} . \mathbf{K}_{dyn} enthält lediglich die bewegten Objekte sowie etwas Rauschen. Man kann erkennen, dass das bewegte Fahrzeug, welches sich direkt links neben dem Sensorfahrzeug befindet, ebenso Spuren in der statischen Karte hinterlässt, siehe Abbildung 5.3c. Dabei handelt es sich um die Seitenwand eines Fahrzeuges, die sich über mehrere Zeitschritte über

dieselbe Stelle bewegt; die zugehörigen Distanzmessungen ändern sich deshalb nicht und werden als vermeintlich statisch identifiziert. Dieser Fehler wird in einem nachfolgenden Schritt korrigiert, siehe Abschnitt 5.4.6.



(a) Bild der Referenzkamera



(b) Aktuelle Messkarte K_{meas}

(c) Statische Karte K_{static}

(d) Dynamische Karte K_{dyn}

Abbildung 5.3 – Die in Kapitel 3 beschriebene Karte wird separiert in eine Messkarte (b), eine statische Karte (c) und eine dynamische Karte (d).

Abbildung 5.4 zeigt den schematischen Ablauf der Erstellung der dynamischen Karte K_{dyn} und der statischen Karte K_{static} über Mengenoperationen.

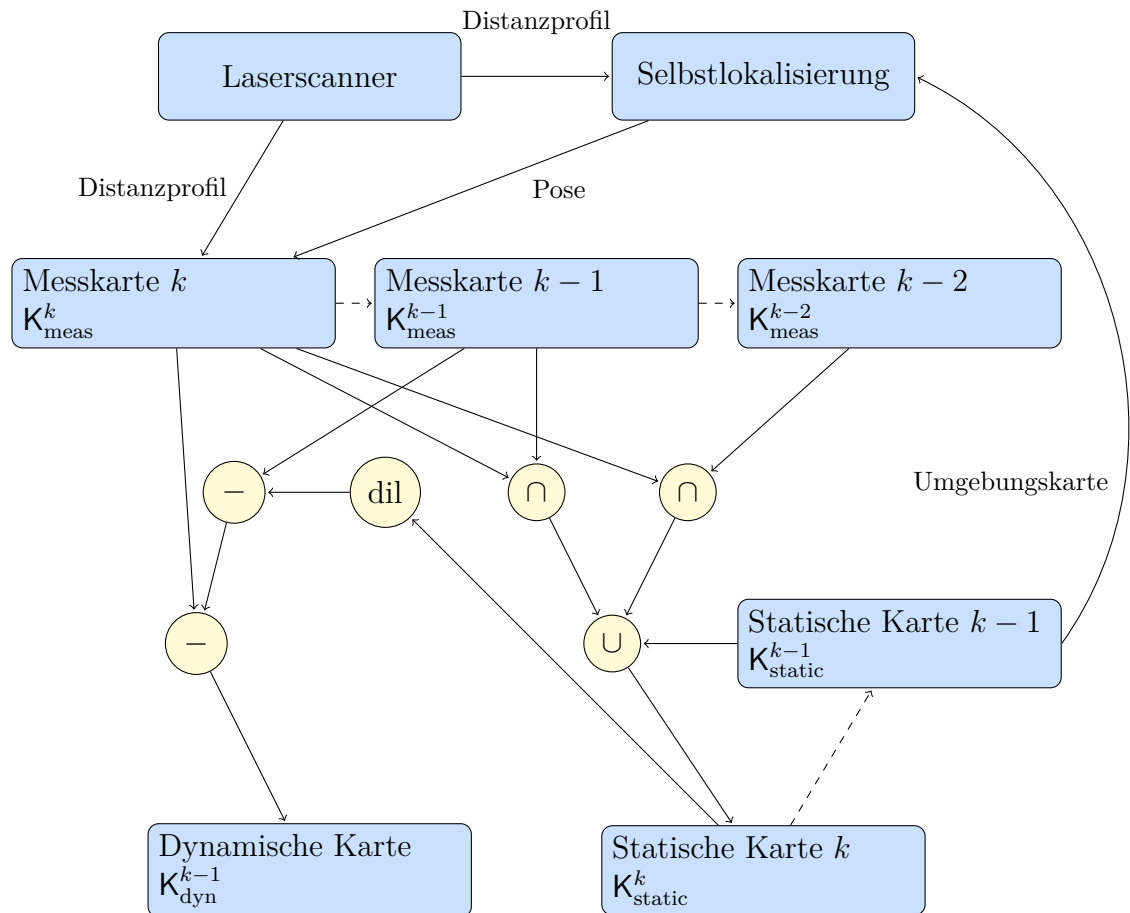


Abbildung 5.4 – Architektur Detektion bewegter Objekte.

5.3.5 Zusammenfassung

Die definierten Funktionen sind sowohl für zwei- als auch dreidimensionale Karten gültig. Obwohl neue statische Objekte und Rauschen erfolgreich aus der dynamischen Karte entfernt werden können, gibt es nach wie vor einen Nachteil der beschriebenen Vorgehensweise: Da die Seitenwand beispielsweise eines LKW sich über mehrere Zeitschritte über dieselbe Stelle bewegt, ändern sich die zugehörigen Distanzmessungen nicht. Selbst durch Betrachtung mehrerer aufeinanderfolgender Messungen werden bewegte Objekte deshalb nicht komplett erfasst. Stattdessen wird die Seitenwand des LKW in der statischen Karte eingetragen, in der sie unerwünscht ist. Dieser Fehler muss in einem nachfolgenden Schritt korrigiert werden, da eine zuverlässige Freiraumanalyse ansonsten nicht möglich ist.

Die dynamische Karte ist ausreichend, um einen Hinweis auf alle bewegten Objekte in einer Szene zu geben, erfasst diese jedoch nicht komplett. Um die beschriebenen Probleme zu lösen sowie um den Zustand von Verkehrsteilnehmern zu schätzen, wird im nächsten Abschnitt die Verfolgung von bewegten Objekten behandelt.

5.4 Verfolgung bewegter Objekte

Zur Schätzung des Zustandes von Verkehrsteilnehmern sind keine Systemeingangsdaten bekannt, Verkehrsteilnehmer können lediglich durch Sensoren wahrgenommen werden. Das in Abschnitt 2.2 vorgestellte allgemeine Bayes-Netzwerk vereinfacht sich damit deutlich zu dem in Abbildung 5.5 gezeigten Netzwerk.

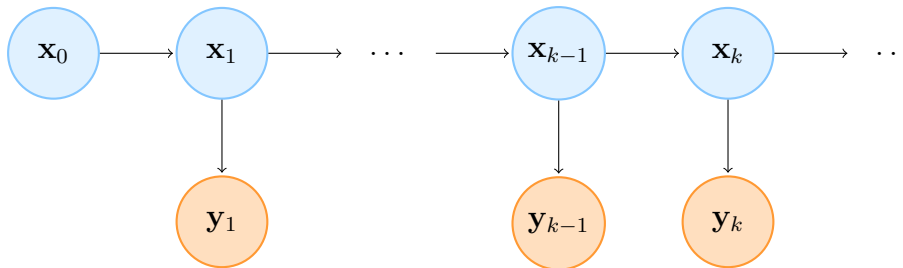


Abbildung 5.5 – Bayes-Netzwerk für die Verfolgung von Verkehrsteilnehmern.

Im vorigen Abschnitt wurde die dynamische Belegungskarte K_{dyn} eingeführt. Diese enthält mit inkonsistenten Zellen Hinweise auf bewegte Objekte und stellt die Messung für das Bayes-Netzwerk dar. Auf Basis dieser soll in diesem Abschnitt eine möglichst robuste Zustandsschätzung aller im Umfeld befindlichen Verkehrsteilnehmern erreicht werden. Für die Zustandsschätzung eines einzelnen Objektes wird zunächst der Systemzustand definiert.

5.4.1 Systemzustand

Im Straßenverkehr treten neben PKW und LKW auch Busse, Motorräder, Fahrräder und Fußgänger auf. Auch wenn nicht alle Verkehrsteilnehmer für jede Funktion relevant sind, so sorgen diese dennoch für Artefakte in der Belegungskarte, wenn sie nicht verfolgt und entfernt werden. Ziel dieses Abschnittes ist somit, alle Verkehrsteilnehmer verfolgen zu können.

Ohne Verwendung eines kamerabasierten Objektklassifikators ist es jedoch äußerst schwierig diese Objektklassen, basierend auf den Messungen des Laserscanners, zuzuordnen. PKW können mit einem 3D-Laserscanner wie dem Velodyne vom Hintergrund separiert werden [60], mit automotive-tauglichen Laserscannern ist dies jedoch nicht ohne weiteres robust möglich [106]. Wender [143] zeigt eine robuste Klassifikation, die jedoch unter anderem auf den dynamischen Parametern eines bereits verfolgten Objektes beruht.

Um alle diese Objektklassen mit einem gemeinsamen Zustandsmodell beschreiben zu können, wird ein sehr einfaches Zustandsmodell gewählt, welches lediglich die Position und die Geschwindigkeit in kartesischen Koordinaten beschreibt:

$$\mathbf{x} = \begin{bmatrix} x & y & v_x & v_y \end{bmatrix} \quad (5.6)$$

5.4.2 Bewegungsmodell

Die Bewegung für einen PKW kann sehr genau modelliert werden, wie es auch im Kapitel Selbstlokalisierung für das Sensorfahrzeug beschrieben ist, siehe Abschnitt 4.3. Die Bewegung von Fußgängern ist deutlich schwieriger zu modellieren, nicht zuletzt aufgrund der geringeren Trägheit. Um für alle Verkehrsteilnehmer ein gemeinsames Bewegungsmodell angeben zu können, kommen nur sehr allgemeine Bewegungsmodelle in Frage. In dieser Arbeit wird das Bewegungsmodell mit konstanter Geschwindigkeit basierend auf dem Zustandsvektor (5.6) betrachtet, wie auch in [76] angewandt.

Bewegungsmodell mit konstanter Geschwindigkeit

Beim Bewegungsmodell mit konstanter Geschwindigkeit wird die Geschwindigkeit als konstant angenommen, wohingegen die Position sich linear mit der Geschwindigkeit ändert. Beschleunigungen eines Verkehrsteilnehmers müssen über das Prozessrauschen in das System einfließen. Hieraus ergibt sich folgendes Modell:

$$x_k = x_{k-1} + v_{x,k-1} \Delta t \quad (5.7)$$

$$y_k = y_{k-1} + v_{y,k-1} \Delta t \quad (5.8)$$

$$v_{x,k} = v_{x,k-1} \quad (5.9)$$

$$v_{y,k} = v_{y,k-1} \quad (5.10)$$

5.4.3 Zustandsschätzung

Zur Schätzung der in Abbildung 5.5 gezeigten Systemzustände können die in Kapitel 2 beschriebenen Bayes-Filter genutzt werden. Im Allgemeinen ist davon auszugehen, dass sich jeweils mehrere Verkehrsteilnehmer im Sichtfeld der Sensoren aufhalten. Die inkonsistenten Zellen aus der Belegungskarte, welche von bewegten Objekten resultieren, enthalten also in der Regel Informationen über mehrere Objekte. Weiterhin ist je nach Szene mehr oder weniger Rauschen in der Karte enthalten. Die dynamische Belegungskarte K_{dyn} kann, entsprechend normiert, somit als diskrete multimodale Wahrscheinlichkeitsdichteverteilung betrachtet werden.

Das Kalman-Filter ist ein optimaler Schätzer für unimodale und normalverteilte Wahrscheinlichkeitsdichtefunktionen. Befinden sich in einer Messung mehrere Objekte, so müssen diese zunächst separiert werden. Das Kalman-Filter ist also auch in diesem Fall unter gewissen Annahmen dazu in der Lage, die Objektzustände zu schätzen. Die notwendige Vorverarbeitung basiert jedoch in der Regel häufig auf anwendungs- und sensorspezifischen Schwellwerten.

Ein Partikelfilter kann direkt multimodale Wahrscheinlichkeitsdichtefunktionen schätzen und ist daher für eine generische Verarbeitung besser geeignet. Mit dem „Track-Before-Detect“ (TBD)-Ansatz [103] kann weiterhin Rauschen eliminiert werden, indem Messungen verfolgt werden, bevor ein Objekt tatsächlich bestätigt wird. Diese Bestätigung erfolgt dann nicht nur auf dem Abstand zwischen Messpunkten, wie bei der Segmentierung von Laserscannerdaten [119, 112], sondern zusätzlich auf der Homogenität der Geschwindigkeitsvektoren.

5.4.4 Partikelfilter

Als Partikelfilter wird der im Grundlagenkapitel vorgestellte SIR-Partikelfilter eingesetzt, siehe Abschnitt 2.3.4.

Die Gewichtung $q_k^{(m)}$ eines Partikels wird definiert als die Summe der mit den Belegungswahrscheinlichkeiten gewichteten Abstände des Partikels zu dynamischen Zellen in der Umgebung.

$$q_k^{(m)} := p\left(\mathbf{K}_{\text{dyn}}^{k-1} | \chi_k^{(m)}\right) \quad (5.11)$$

$$= \sum_{\{n \in \mathbf{K}_{\text{dyn}}^{k-1}\}} p(n) \cdot p\left(m(n) | \chi_k^{(m)}\right) \quad (5.12)$$

Die Wahrscheinlichkeit $p(n)$ bezeichnet dabei die Belegungswahrscheinlichkeit des Knotens n und lässt sich direkt aus \mathbf{K}_{dyn} ermitteln. Die Wahrscheinlichkeit $p\left(m(n) | \chi_k^{(m)}\right)$ hingegen wird über eine Gewichtungsfunktion realisiert, die den gewichteten Abstand des Mittelpunktes $m(n)$ des Knotens n zu einem gegebenen Partikel $\chi_k^{(m)}$ bestimmt.

Gewichtungsfunktion

Die Gewichtungsfunktion sorgt dafür, dass „gute“ Partikel beim Resampling höher gewichtet werden als andere. „Gut“ ist ein Partikel dann, wenn es möglichst nahe an Messungen, das heißt an inkonsistenten Zellen liegt. Als Gewichtungsfunktionen werden in der Regel Kern-dichtefunktionen verwendet. Eine häufig gewählte Gewichtungsfunktion ist der Gaußkern. Hier gilt:

$$p(\mathbf{y}_k | \chi_k^{(m)}) = \eta \exp \left\{ -\frac{u^2}{2} \right\} \quad (5.13)$$

Als quadratische Funktion, die effizienter zu berechnen ist, kommt der vereinfachte Epanechnikov-Kern [37] in Frage:

$$p(\mathbf{y}_k | \chi_k^{(m)}) = \begin{cases} \frac{3}{4}(1 - u^2) & , \text{ falls } |u^2| \leq 1 \\ 0 & , \text{ sonst} \end{cases} \quad (5.14)$$

Abbildung 5.6 zeigt die beiden untersuchten Kerndichtefunktionen, die Normalverteilung und den vereinfachten Epanechnikov-Kern.

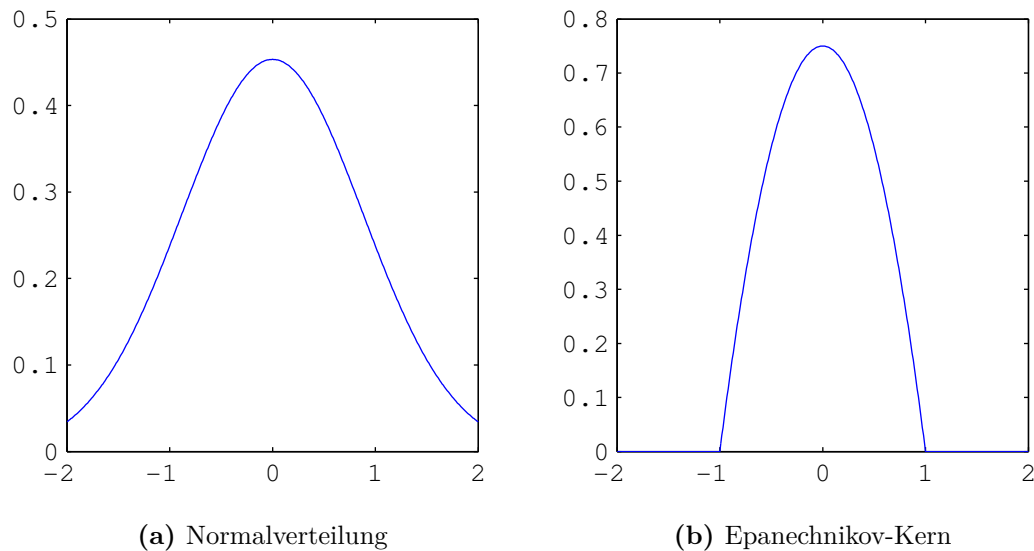


Abbildung 5.6 – Die zwei untersuchten Kerndichtefunktionen.

Verfolgung mehrerer Objekte

Mit einem Partikelfilter können theoretisch multimodale Verteilungen geschätzt werden. Dies funktioniert für die Verfolgung mehrerer Objekte aber nur, wenn folgende Annahmen erfüllt sind [107]: Alle Objekte sind dauerhaft sichtbar, haben ein ähnliches Signal-Rausch-Verhältnis, die Anzahl der Objekte ist bekannt und statisch und alle Objekte folgen ähnlichen Bewegungsmodellen. Weiterhin ist eine sehr hohe Anzahl an Partikeln notwendig. Da diese Annahmen für die in Abschnitt 5.3 beschriebene dynamische Karte \mathbf{K}_{dyn} nicht erfüllt sind, tendieren die Partikel dazu, auf das Objekt mit der höchsten Gewichtung zu konvergieren [82].

Zur Lösung dieses Problems werden in der Literatur in der Regel mehrere Partikelfilter eingesetzt: Schulz [107] beschreibt die Kombination mehrerer Partikelfilter mit JPDA (engl. Joint Probabilistic Data Association), um mehrere Personen im Innenraum von Gebäuden zu verfolgen. Jung [66] nutzt ebenfalls mehrere Partikelfilter, um mehrere Personen in einem Kamerabild zu verfolgen. MacCormick nutzt einen partitionierten Partikelfilter [82].

In dieser Arbeit wird ebenfalls die Zustandsschätzung mehrere Objekte über mehrere Partikelfilter gewählt. Für die Initialisierung der Partikelfilter wird demnach eine Schätzung über die Anzahl der vorhandenen Objekte benötigt. Weiterhin muss beobachtet werden, ob neue Objekte im Sichtbereich auftauchen oder alte Objekte diesen verlassen. Für diesen Zweck wird die dynamische Karte K_{dyn} segmentiert. Als Segmentierungsverfahren für die dynamischen Zellen wird der DBSCAN-Algorithmus [39] verwendet. Zu Beginn wird für jede Häufung von inkonsistenten Zellen ein Partikelfilter aufgesetzt. Taucht im weiteren Verlauf ein neues Objekt im Sichtbereich auf, das noch nicht durch einen konvergierten Partikelfilter verfolgt wird, so wird an dieser Stelle ein neuer Partikelfilter aufgesetzt. Verlässt ein Objekt den Sichtbereich hingegen, so wird das zugehörige Partikelfilter nach einiger Zeit gelöscht. Um festzustellen, ob ein Partikelfilter bereits auf einem Objekt konvergiert ist, wird dessen Streuung ermittelt.

Bestimmung der Streuung eines Partikelfilters

Wird pro Partikelfilter lediglich ein reales Objekt verfolgt, so konvergiert die Partikelmenge auf diesem Objekt. Über die Streuung der Partikel, die über eine Kovarianzmatrix angegeben werden kann, kann darauf geschlossen werden, ob die Partikelmenge bereits konvergiert ist oder nicht. Der gewichtete Mittelwert der Partikelmenge gibt Aufschluss über den geschätzten Zustand des Objektes.

Für ein Partikelfilter werden im Folgenden neben den Partikeln und den zugehörigen Gewichten ein gewichteter Mittelwert $\bar{\chi}$ und eine Kovarianzmatrix \mathbf{C}_χ betrachtet. Dabei wird \mathbf{C}_χ aus \mathbf{C}_{pos} und \mathbf{C}_v zusammengesetzt. Der Übersichtlichkeit wegen wird an dieser Stelle auf den Zeitindex verzichtet.

$$\bar{\chi} = \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{v}_x \\ \bar{v}_y \end{bmatrix} = \frac{1}{M-1} \sum_m \begin{bmatrix} x^{(m)} \\ y^{(m)} \\ v_x^{(m)} \\ v_y^{(m)} \end{bmatrix} \quad (5.15)$$

$$\mathbf{C}_{\text{pos}} = \frac{1}{M-1} \sum_m \begin{bmatrix} (x^{(m)} - \bar{x})^2 & (x^{(m)} - \bar{x})(y^{(m)} - \bar{y}) \\ (x^{(m)} - \bar{x})(y^{(m)} - \bar{y}) & (y^{(m)} - \bar{y})^2 \end{bmatrix} \quad (5.16)$$

$$\mathbf{C}_v = \frac{1}{M-1} \sum_m \begin{bmatrix} (v_x^{(m)} - \bar{v}_x)^2 & (v_x^{(m)} - \bar{v}_x)(v_y^{(m)} - \bar{v}_y) \\ (v_x^{(m)} - \bar{v}_x)(v_y^{(m)} - \bar{v}_y) & (v_y^{(m)} - \bar{v}_y)^2 \end{bmatrix} \quad (5.17)$$

$$\mathbf{C}_\chi = \begin{bmatrix} \mathbf{C}_{\text{pos}} & 0 \\ 0 & \mathbf{C}_v \end{bmatrix} \quad (5.18)$$

Trackverwaltung

Die Trackverwaltung zur Objektverfolgung und anschließenden Detektion (TBD) wird über eine Zustandsmaschine realisiert. In Abbildung 5.7 ist diese dargestellt.

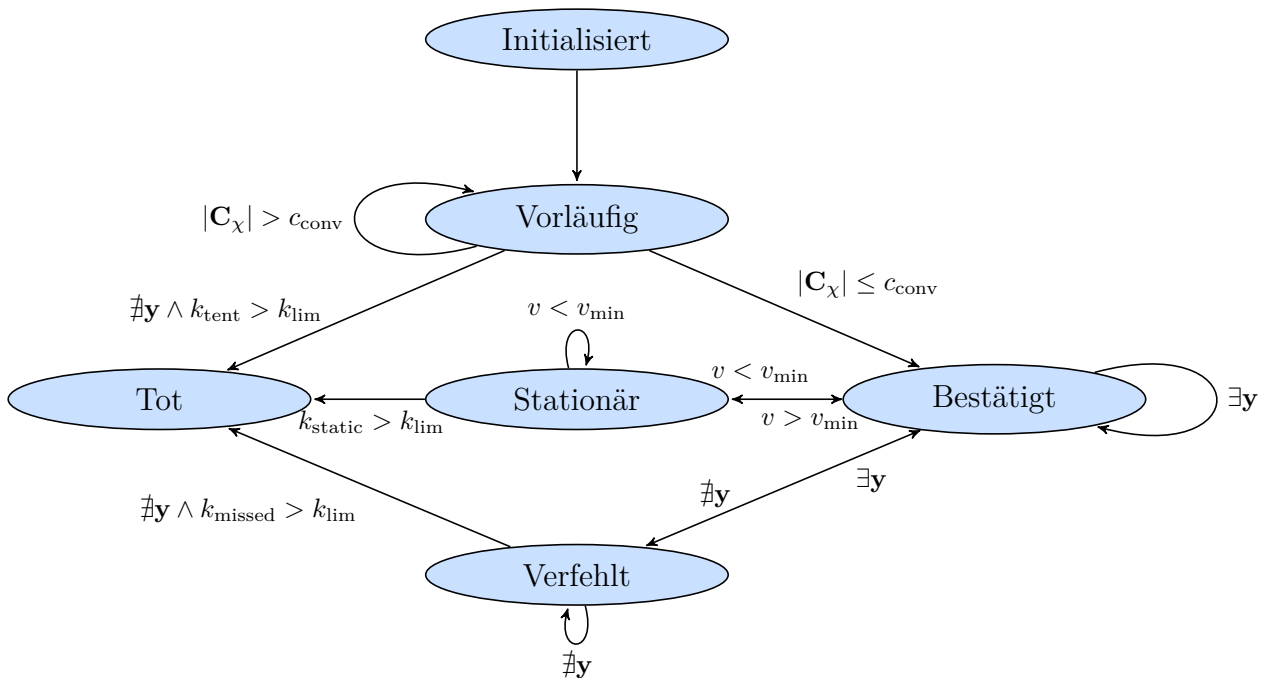


Abbildung 5.7 – Trackverwaltung für die Verfolgung bewegter Objekte.

Die Zustandsmaschine beginnt im Initialisierungszustand. Für das Partikelfilter werden Stichproben lediglich aus inkonsistenten Zellen von \mathbf{K}_{dyn} gezogen. Jede inkonsistente Zelle n_{dyn} wird initial durch $\frac{|\mathcal{X}|}{|\{n_i: n_i \in \mathbf{K}_{\text{dyn}}, p(n_i) > 0,5\}|}$ Partikel repräsentiert, also die Anzahl der Partikel geteilt durch die Anzahl der inkonsistenten belegten Zellen in der dynamischen Belegungskarte.

Abbildung 5.8a zeigt die Initialisierung des Partikelfilters durch die erste dynamische Karte $\mathbf{K}_{\text{dyn}}^0$. Für das CV-Modell wird v gleichverteilt aus dem Intervall $[-a\Delta k, a\Delta k]$ gezogen. Für den Velodyne Laserscanner liegt Δk bei 0,1 s.

Nach der Initialisierung wechselt der Zustandsautomat in den Zustand „Vorläufig“. In diesem Schritt verteilen sich die Partikel ihrer zufälligen Initialisierung entsprechend. Die Abbildungen 5.8b und 5.8c geben Beispiele. Treten keine Messungen mehr auf, so wechselt der Zustand nach „Tot“. Wenn zwar Messungen auftreten, der Prozessfehler aber groß ist, verbleibt der Track im Zustand „Vorläufig“. Sinkt das Volumen der Kovarianzmatrix \mathbf{C}_χ unter eine Konstante c_{conv} , so wechselt der Zustand auf „Bestätigt“. Dies gilt als Detektion für ein bewegtes Objekt und als Bestätigung für den Track. In den Abbildungen 5.8d und 5.8e ist ein bestätigtes Objekt dargestellt, das durch den Partikelfilter verfolgt wird. Wird lediglich ein einzelnes Objekt verfolgt, so verlässt der Zustandsautomat diesen Zustand nicht mehr; vorausgesetzt es sind Messungen vorhanden.

So lange der Track im Zustand „Vorläufig“ ist, wird ein Zähler k_{tent} inkrementiert. Konvergiert der Track nicht, bevor eine Schwelle k_{lim} überschritten ist, so wird der Track entfernt. Dies dient dazu, Rauschen zu eliminieren.

In Abbildung 5.8f ist die zugehörige statische Karte abgebildet, in der alle Artefakte, die durch den vorausfahrenden PKW entstehen würden, nicht mehr vorhanden sind. In Abbildung 5.9 ist das zugehörige Bild der Referenzkamera dargestellt.

Treten während der Objektverfolgung keine inkonsistenten Zellen mehr auf, so wechselt der Filter in den Zustand „Verfehlt“. In diesem Zustand werden keine Innovationen mehr berücksichtigt, der Zustand wird lediglich prädiert. Dieser Zustand dient dazu, Objekte korrekt zu verfolgen die kurzzeitig verdeckt sind, den Sichtbereich verlassen oder aus anderen Gründen keine Messungen liefern. Die Gewichtung der Partikel wird nicht geändert; der Resampling-schritt wird demnach auch nicht durchgeführt. Die Prozessfehlerkovarianzmatrix wächst und ein Zähler k_{missed} wird erhöht. Treten nach kurzer Zeit wieder Messungen auf, so kann der Partikelfilter die Verfolgung wieder aufnehmen. Treten jedoch keine neuen Messungen mehr auf, wird mit dem Zähler k_{missed} eine definierte Schwelle überschritten und der Zustand wechselt auf „Tot“.

Verkehrsteilnehmer bewegen sich in der Regel nicht durchgängig, sondern bleiben unter anderem an Ampeln oder Kreuzungen stehen. Um dieses Verhalten zu modellieren, gibt es einen weiteren Zustand: „Stationär“. Dieser wird erreicht, wenn die geschätzte Geschwindigkeit v des Objektes unter eine Schwelle v_{min} fällt. In diesem Zustand wird der Zähler k_{static} inkrementiert.

Diese Modellierung hat zwei Vorteile: Zum einen entstehen keine statischen Zellen, die sich an der Position des temporär stehen gebliebenen Objektes bilden würden. Zum anderen befindet sich der Filter bereits im Zustand „Bestätigt“ und die Verfolgung kann ohne Verzögerung wiederaufgenommen werden, sobald das Objekt seine Bewegung wieder aufnimmt. Überschreitet k_{static} jedoch eine definierte Schwelle, so wird der Track entfernt und die zugehörigen Zellen werden wieder als statisch in die Karte eingetragen.

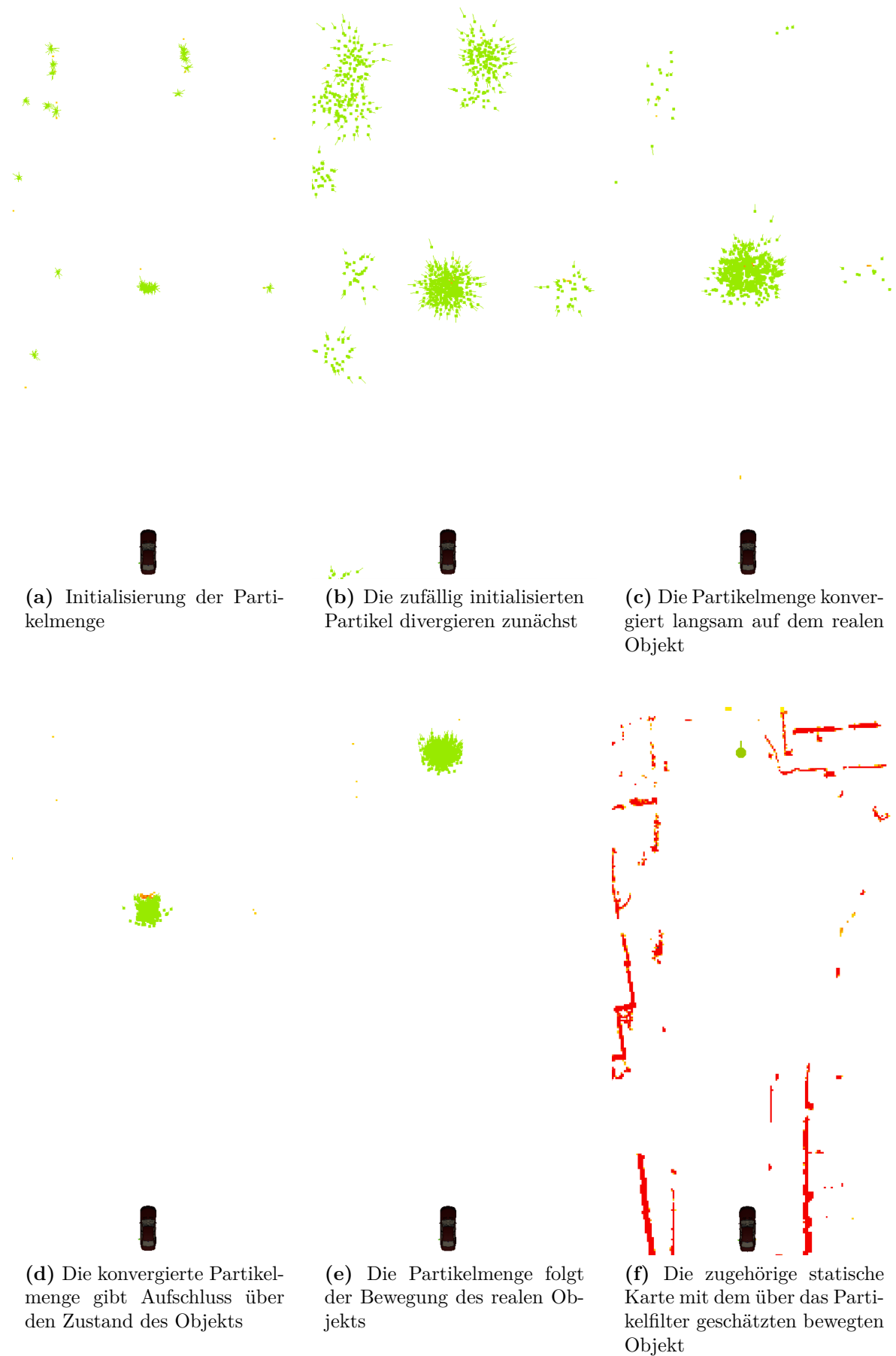


Abbildung 5.8 – Exemplarische Veranschaulichung der Zustände der Trackverwaltung.



Abbildung 5.9 – Bild der Referenzkamera zu der in Abbildung 5.8 betrachteten Sequenz.

5.4.5 Kalman-Filter

Im Vergleich zur Verfolgung mit dem Partikelfilter wird für ein einzelnes Objekt auch die Zustandsschätzung mit einem Kalman-Filter betrachtet. Wie in Abschnitt 5.2.1 zur objektbasierten Verfolgung beschrieben, wird dazu eine Segmentierung der Messpunkte des Laserscanners benötigt. Hier wird, wie für die Initialisierung des Partikelfilters, der DBSCAN-Algorithmus [39] eingesetzt. Die Mittelpunkte der Häufungen werden als Messung für die Objektposition aufgefasst.

Prozessrauschen

Für die maximale Beschleunigung eines Fahrzeuges wird $a_{\max} = 9,81 \frac{m}{s^2}$ angenommen. Wird diese als 3σ -Intervall interpretiert, so gilt für die Standardabweichung der Beschleunigung $\sigma_a = 3,27 \frac{m}{s^2}$.

Für den Velodyne Laserscanner gilt $\Delta k = 0,1 s$ und damit $\sigma_v = 0,327 \frac{m}{s}$. Da die Bewegungsrichtung des Objektes nicht explizit modelliert wird, wird approximativ sowohl σ_{v_x} als auch σ_{v_y} auf $0,327 \frac{m}{s}$ gesetzt.

Die Kovarianzmatrix für das Prozessrauschen ergibt sich somit zu:

$$\mathbf{Q} = \text{diag} \left(0, 0, \sigma_{v_x}^2, \sigma_{v_y}^2 \right) \quad (5.19)$$

Messung

Die Messung ergibt sich nach dem Segmentierungsschritt als Mittelpunkt der ermittelten Häufung inkonsistenter Zellen:

$$\mathbf{y} = \begin{bmatrix} x & y \end{bmatrix}^T \quad (5.20)$$

Messrauschen

Das Messrauschen ist ebenfalls durch die Segmentierung bestimmt: Die Messpunkte des Laserscanners liegen in der Regel auf der Außenkontur von Objekten, wohingegen als Referenzpunkt die Objektmitte angenommen wird. Bei der Evaluierung wird ein PKW betrachtet. Wenn alle Messpunkte auf der Rückfront des PKW liegen und dessen Länge 5 m beträgt, so entsteht ein maximaler Fehler von 2,5 m. Wird dieser als 3σ -Intervall angenommen, so ergeben sich stark vereinfachend die Standardabweichungen σ_x und σ_y zu 0,833 m.

Die Kovarianzmatrix für das Messrauschen \mathbf{R} wird demnach festgelegt auf:

$$\mathbf{R} = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix} \quad (5.21)$$

Messmatrix

Die Messmatrix, die für den Innovationsschritt des Kalman-Filters erforderlich ist, ergibt sich direkt aus dem Zustandsvektor und dem Messvektor:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (5.22)$$

5.4.6 Aktualisierung der statischen Karte

In Abschnitt 5.3.3 wurde geschildert, dass trotz der aufwändigen Separierung der statischen Karte $\mathbf{K}_{\text{static}}$ von der dynamischen Karte \mathbf{K}_{dyn} noch Artefakte von bewegten Objekten in der statischen Karte auftreten. Ein bestätigtes bewegtes Objekt kann dazu genutzt werden, dieses Problem zu lösen und die statische Karte zu aktualisieren. Es wird angenommen, dass sich ein dynamisches Objekt stets über einen Freiraum bewegt. Das bedeutet, dass fälschlicherweise

belegte Bereiche, die durch Rauschen verursacht werden und über die sich ein Objekt bewegt, in der statischen Karte gelöscht werden können.

Aus der Prozessfehlerkovarianz des Partikelfilters wird die Ellipse berechnet, die dem 1σ -Intervall entspricht.

In dem Bereich, der innerhalb dieser Ellipse liegt, werden belegte Zellen gelöscht. Dies dient der Robustheit der statischen Karte. Werden jedoch Zellen aus der statischen Karte aufgrund einer Falschdetektion gelöscht, tritt folgendes Problem auf: Statische Zellen aus der Messkarte werden dann als bewegt identifiziert, was eine Verstärkung des Effektes zur Folge hat. Dieses Problem wurde bereits in Abschnitt 5.3.4 beschrieben. Falschdetektionen durch Rauschen müssen daher vermieden werden, beispielsweise durch eine Anpassung der Konstante c_{conv} .

5.5 Auswertung

Zur Evaluierung der vorgestellten Ansätze zur Detektion und Verfolgung bewegter Objekte ist sowohl für die geschätzte Position als auch für die geschätzte Geschwindigkeit der verfolgten Objekte eine Referenzmessung notwendig.

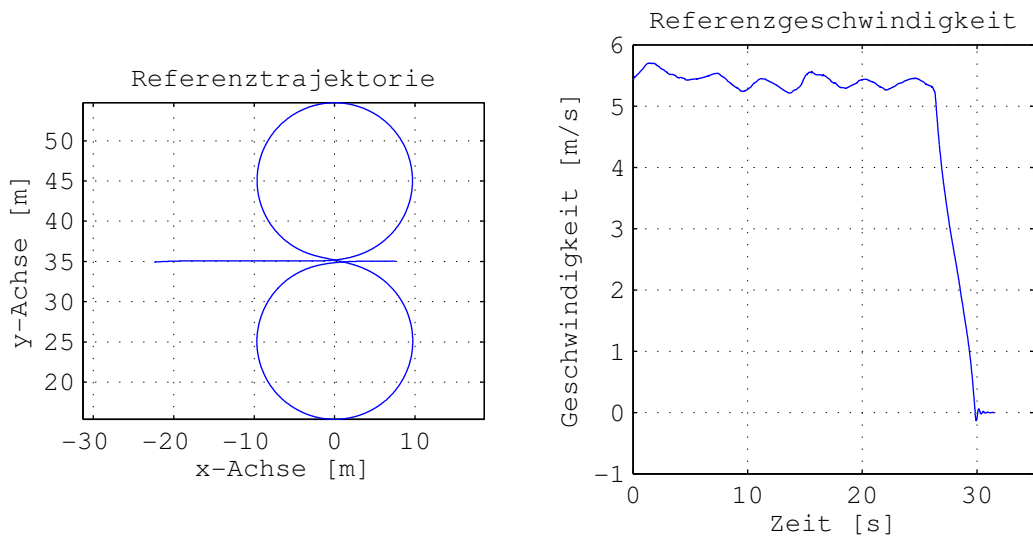
5.5.1 Szenario

Mit den automatisch fahrenden Fahrzeugen der Daimler AG [16] wurde solch eine Referenzmessung durchgeführt. Der in Abschnitt 6.1.4 beschriebene Versuchsträger wurde mit einem Anthony Best Dynamics (ABD)-Fahrroboter ausgestattet. Dieser erlaubt es, vorgegebene Trajektorien hochgenau abzufahren und aufzuzeichnen. Ein weiterer stationärer Versuchsträger wurde mit einem Velodyne Laserscanner ausgestattet um das Szenario aufzuzeichnen. Abbildung 5.10 zeigt das Roboterfahrzeug auf einem Testgelände.

An dem Tag der Messung hat es geregnet, was zu einem erhöhten Messrauschen führt. Zudem liefert ein schwarzes Fahrzeug wenige Messungen: Aufgrund der schwarzen Lackierung wird das Laserlicht absorbiert. Als Trajektorie wurde ein schwieriges Szenario aufgezeichnet: Der Versuchsträger fährt die in Abbildung 5.11a dargestellte Acht ab, bei der die Perspektive auf den Versuchsträger ständig wechselt und mehrere Richtungsänderungen vorgenommen werden. Die Referenz für die Geschwindigkeit des Roboters ist in Abbildung 5.11b dargestellt.



Abbildung 5.10 – Roboterfahrzeug, mit dem vorgegebene Trajektorien sehr genau abgefahren und aufgezeichnet werden können.



(a) Trajektorie des Fahrroboters. Das beobachtende Fahrzeug befindet sich an Position (0,0).

(b) Geschwindigkeit des Fahrroboters.

Abbildung 5.11 – Referenzfahrt des Roboterfahrzeuges.

5.5.2 Auswertung der Positionsschätzung

In Abbildung 5.12 sind die geschätzten Trajektorien der Objektverfolgungsansätze dargestellt. Die dynamische Belegungskarte K_{dyn} hat eine Auflösung von 0,2 m. Beide Ansätze sind in der Lage die Trajektorie nachzuvollziehen, trotz des einfachen CV-Bewegungsmodells. Das Partikelfilter arbeitet mit $M = 1000$ Partikeln und der Normalverteilung als Kerndichtefunktion.

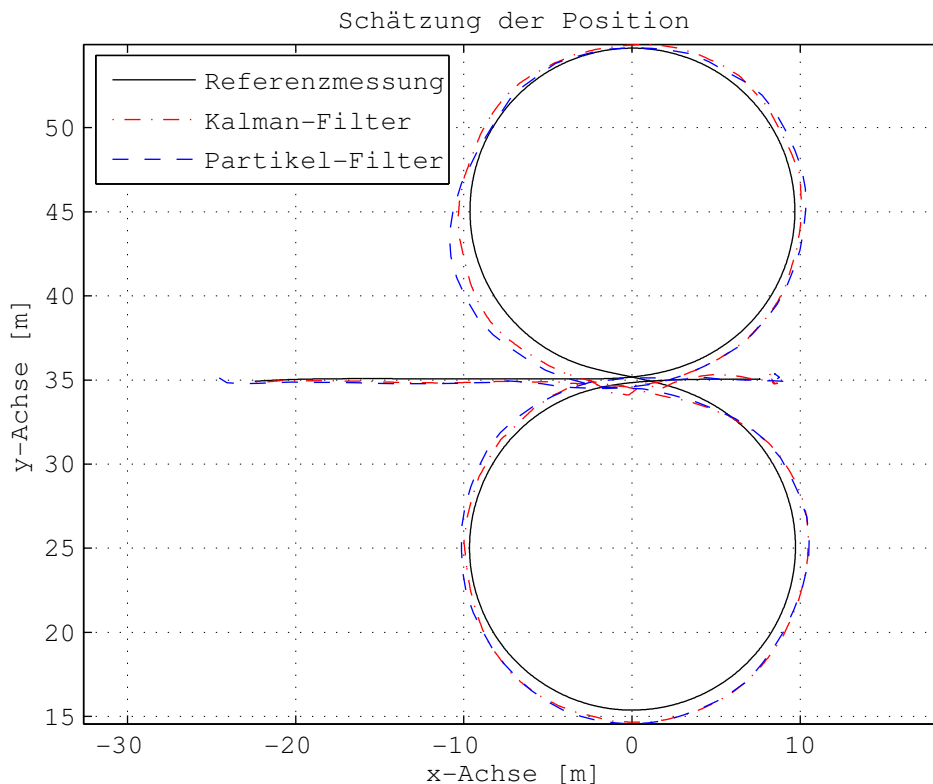


Abbildung 5.12 – Trajektorien der Objektverfolgung mit Partikelfilter und Kalman-Filter im Vergleich zur Referenz.

In Abbildung 5.13 sind die Fehler der Positionsschätzung der Objektverfolgungsansätze dargestellt. Das Kalman-Filter schätzt die Position des Fahrzeuges zu Beginn genauer, später lässt sich im Vergleich zum Partikelfilter kein eindeutiger Unterschied mehr ausmachen. Weiterhin wird der Epanechnikov-Kern als Kerndichtefunktion betrachtet.

Der RMSE des Kalman-Filters beträgt 1,14 m. Der des Partikelfilters mit $M = 1000$ und der normalverteilten Kerndichtefunktion 1,15 m. Deutlich schlechter ist der Partikelfilter mit $M = 1000$ und dem vereinfachten Epanechnikov-Kern: Hier beträgt der RMSE 1,63 m.

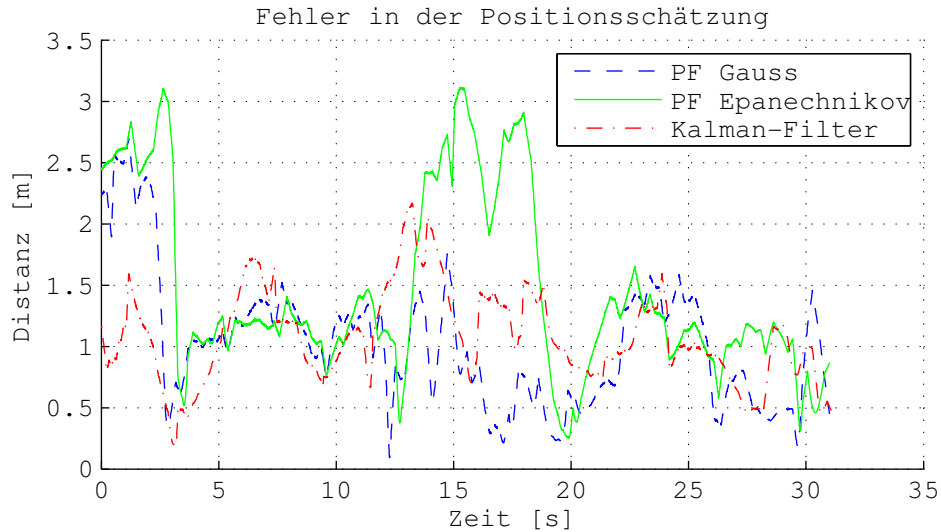


Abbildung 5.13 – Fehler der geschätzten Position der Objektverfolgung mit Partikelfilter und Kalman-Filter zur Referenz.

5.5.3 Auswertung der Geschwindigkeitsschätzung

In Abbildung 5.14 sind die geschätzten Geschwindigkeiten der Objektverfolgungsansätze dargestellt. Die Schwankungen in der Schätzung sind darauf zurückzuführen, dass die Perspektive auf das verfolgte Fahrzeug wechselt und die inkonsistenten Zellen von der Rückseite des Fahrzeugs in kurzer Zeit zur Vorderseite wandern.

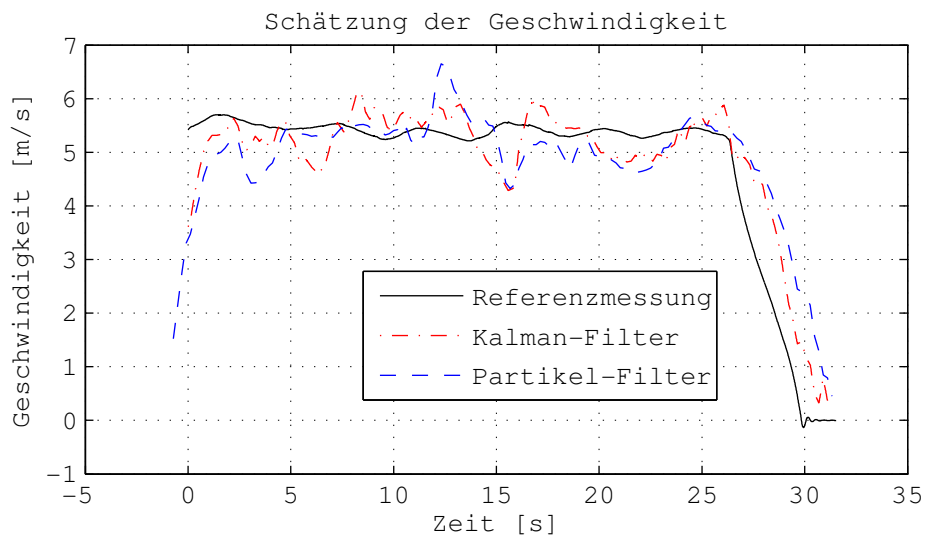


Abbildung 5.14 – Geschätzte Geschwindigkeit der Objektverfolgung mit Partikelfilter und Kalman-Filter im Vergleich zur Referenz.

In Abbildung 5.15 sind die geschätzten Fehler der Geschwindigkeitsschätzung der Objektverfolgungsansätze dargestellt. Das Partikelfilter mit $M = 1000$ und der Normalverteilung als

Kerndichtefunktion hat einen RMSE von $0,57 \frac{m}{s}$, das mit dem vereinfachten Epanechnikov-Kern hat einen RMSE von $0,56 \frac{m}{s}$. Das Kalman-Filter schneidet mit einem RMSE von $0,49 \frac{m}{s}$ etwas besser ab. Bei $t = 13s$ und $t = 30s$ gibt es jeweils einen deutlichen Anstieg des Fehlers: Bei $t = 13s$ rutschen die Partikel vom Fahrzeugheck über die Fahrzeugseite zur Fahrzeugfront, wodurch eine zu hohe Geschwindigkeit geschätzt wird. Bei $t = 30s$ hingegen verletzt der plötzliche Stillstand des Fahrzeuges nach der Verzögerung das Bewegungsmodell und führt so zu einem erhöhten Fehler.

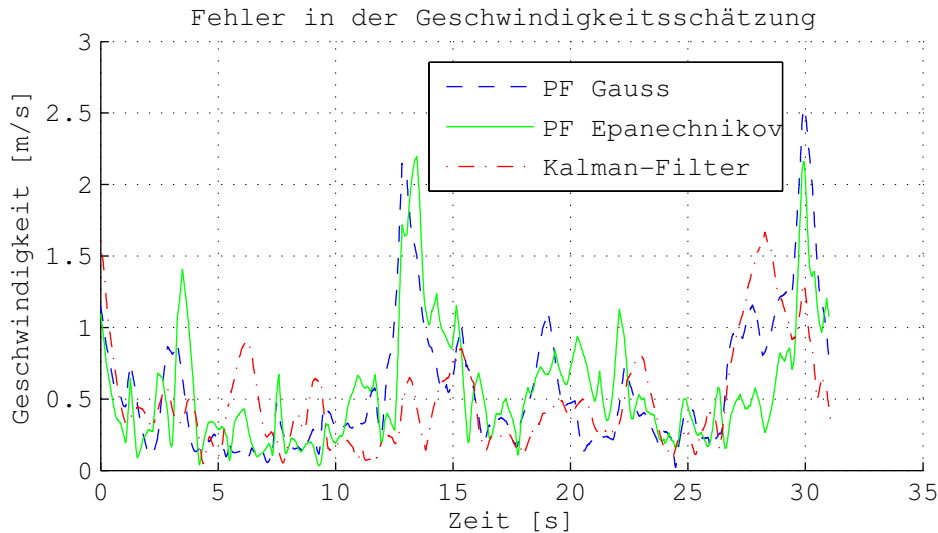


Abbildung 5.15 – Fehler der geschätzten Geschwindigkeit der Objektverfolgung mit Partikelfilter und Kalman-Filter zur Referenz.

5.5.4 Verfolgung mehrerer Objekte

Für Sequenzen mit mehreren dynamischen Objekten stehen keine Referenzdaten zur Verfügung. Die Verfolgung mehrerer Objekte wird daher an einer innerstädtischen Sequenz ohne Referenzdaten exemplarisch betrachtet. Da für jedes hypothetische Objekt ein Partikelfilter aufgesetzt wird, wird die Anzahl der Partikel pro Filter auf $M = 500$ reduziert.

Abbildung 5.16a zeigt den ersten Ausschnitt aus der Sequenz. Das Sensorfahrzeug steht an einer Ampel und wartet. Von hier sind Fußgänger und mehrere abbiegende Fahrzeuge zu beobachten. Es sind vier bestätigte Objekte zu sehen. Das dem Sensorfahrzeug nächstliegende Objekt entspricht einem Kind auf einem Skateboard, es ist ganz rechts am Rand des Bildes der Referenzkamera (Abbildung 5.16b) zu sehen. Das Objekt mit dem größten Abstand entspricht einem auf den Referenzbildern kaum zu erkennenden Fußgänger, der die Straße von links nach rechts überquert. Die beiden weiteren bestätigten Objekte in der Mitte entsprechen zwei PKW, die von rechts kommen und nach links beziehungsweise rechts abbiegen.

Abbildung 5.16b zeigt dieselbe Szene wenige Augenblicke später. Nun sind fünf bestätigte Objekte vorhanden: Der Fußgänger ist auf der anderen Straßenseite angekommen, der Skateboardfahrer nähert sich dem Sensorfahrzeug an. Die beiden PKW befinden sich im Abbiegevorgang. Es ist hervorzuheben, dass der hintere der beiden PKW gut verfolgt werden kann, obwohl er zu großen Teilen verdeckt ist. Als weiteres Fahrzeug kommt von rechts ein LKW, der auf dem Dokumentationsbild noch nicht zu sehen ist.

In Abbildung 5.16c ist der LKW bereits fast vollständig abgebogen. Insgesamt sind sechs bestätigte Objekte vorhanden. Der LKW fährt dem schwarzen Fahrzeug aus Abbildung 5.16b hinterher und ihm folgen zwei weitere PKW. Der LKW und die zwei ihm folgenden PKW sind im Dokumentationsbild in Abbildung 5.16c sehr gut zu erkennen. Ebenso sind alle vier Objekte gut in der Draufsicht identifizierbar. Der Skateboardfahrer ist nochmals näher gekommen. Das Objekt im Hintergrund ist eine Falschdetektion: Die fehlende Modellierung von Verdeckungen führt zu einer scheinbaren Bewegung auf der Hauswand. Diese wird in der Annahme, dass ein bewegtes Objekt vorhanden ist, gelöscht. Zur Vervollständigung des Ansatzes ist daher zusätzlich die Modellierung und Behandlung von Verdeckungen zu lösen.

5.6 Zusammenfassung und Ausblick

Zusammenfassung

In diesem Kapitel wurden Methoden zur Detektion und Verfolgung bewegter Objekte in Belegungskarten vorgestellt. Die Detektion und Verfolgung arbeiten dabei nahezu modellfrei, um den Anspruch, möglichst viele unterschiedliche Objekte verfolgen zu können, zu erfüllen.

Erstmalig in der Literatur wurden dabei bewegte Zellen in einer hierarchischen Karte detektiert. Die Detektion erfolgte durch Mengenoperationen auf Belegungskarten und führt zu einer Separierung von statischer Karte und dynamischer Karte. Die dennoch auftretenden Artefakte von dynamischen Objekten in der statischen Karte wurden über eine Objektverfolgung durch ein Partikelfilter eliminiert, um eine robuste statische Umgebungsrepräsentation für die Freiraumanalyse zu erreichen.

Zur Verfolgung von dynamischen Objekten wurden zwei Ansätze verglichen: Ein Partikelfilter mit einer angepassten Gewichtungsfunktion und das Kalman-Filter mit einer vorangehenden Segmentierung und Referenzpunktbestimmung.

Das Verfahren wurde mit zwei Roboterfahrzeugen der Daimler AG evaluiert. Es wurde ein Szenario mit äußerst schwierigen Bedingungen für einen Laserscanner zur Evaluierung gewählt: Bei Regen wird ein schwarz lackiertes Fahrzeug verfolgt, welches ständig die Fahrtrich-

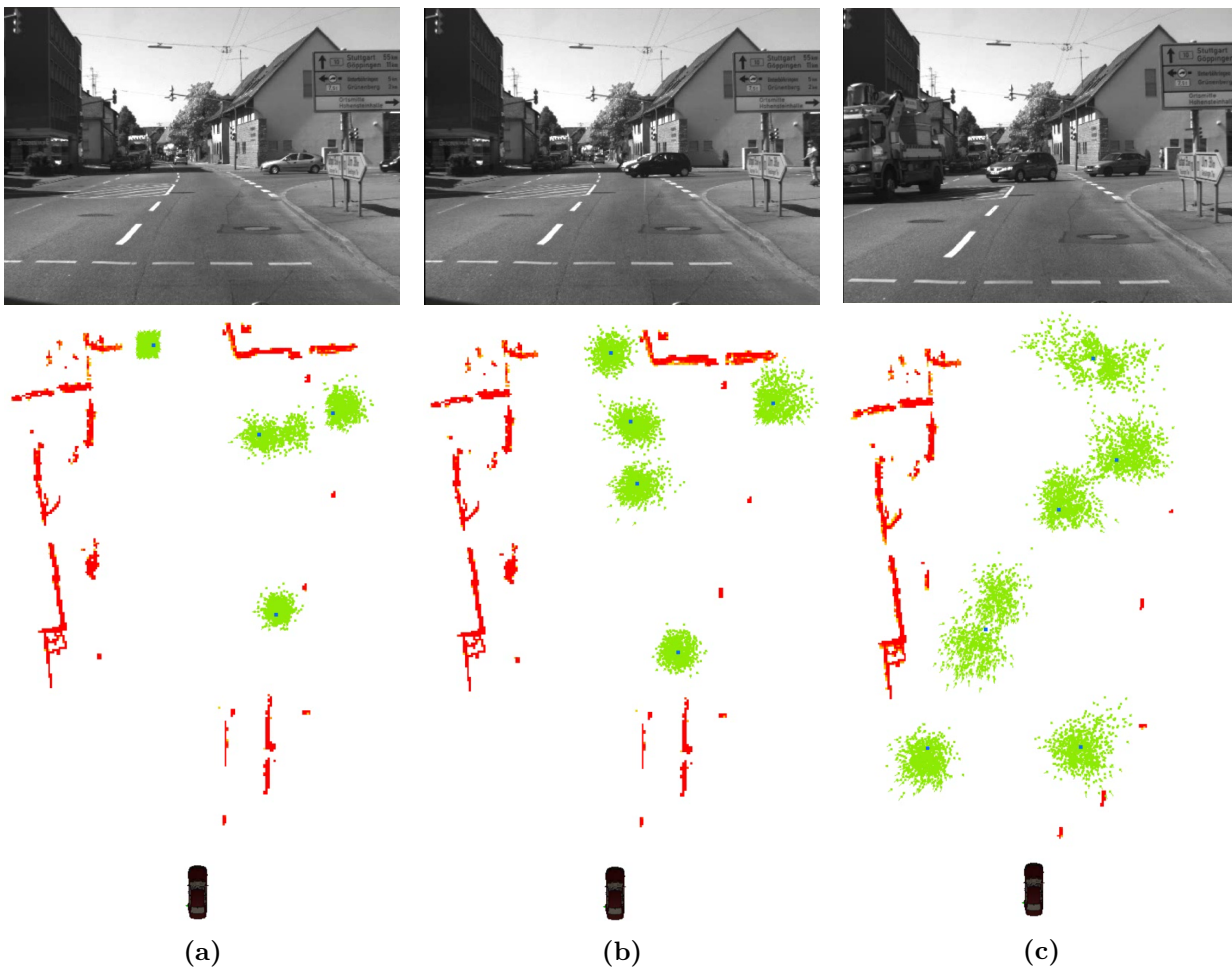


Abbildung 5.16 – Verfolgung bewegter Objekte. Oben ist jeweils das Bild der Dokumentationskamera abgebildet, unten die zugehörige Draufsicht auf die statische Belegungskarte K_{static} und die Zustandsschätzung durch mehrere Partikelfilter. Die Belegungskarte ist in rot dargestellt, die Partikel in grün. Bestätigte Objekte werden als blaue Punkte dargestellt. Der Ansatz ist in der Lage, mehrere Objekte simultan zu verfolgen.

tung ändert.

Im Hinblick auf den Vergleich des Kalman-Filters mit dem Partikelfilter sei darauf hingewiesen, dass das Kalman-Filter auf der Segmentierung der detektierten inkonsistenten Zellen beruht. Die Parametrierung für die Segmentierung ist stark abhängig von dem betrachteten Szenario, um eine robuste Verfolgung zu ermöglichen. Beim Partikelfilter wird die Segmentierung lediglich für die Initialisierung der Anzahl der zu verfolgenden Objekte genutzt. Das eingesetzte Partikelfilter ist im Gegensatz zum eingesetzten Kalman-Filter demnach nicht für Segmentierungsfehler anfällig.

Bei dem evaluierten Szenario zeigt sich mit dem Partikelfilter ein Fehler von 1,15 m in der Positionsbestimmung und ein Fehler von $0,57 \frac{m}{s}$ in der Geschwindigkeitsschätzung. Das Kalman-Filter erzielt zum Vergleich für ein einzelnes Objekt einen Fehler von 1,14 m in der Positions-

bestimmung und einen Fehler von $0,49 \frac{m}{s}$ in der Geschwindigkeitsschätzung.

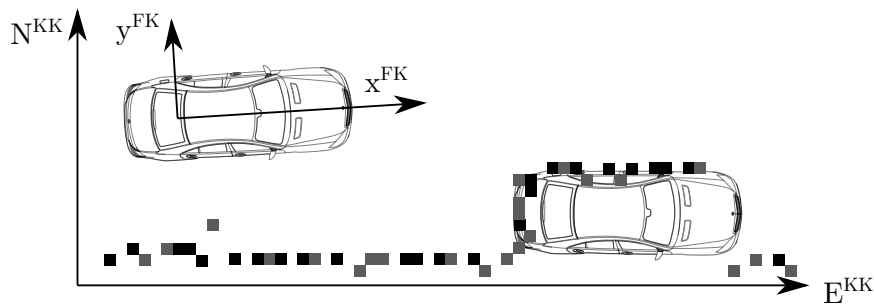
Der hohe Fehler in der Positionsschätzung ist darauf zurückzuführen, dass als gemeinsames möglichst generisches Objektmodell für alle Objekte das Punktmodell gewählt wurde. Dieses passt nicht ideal auf PKW. Zudem sind bei Szenarien mit einfacheren Manövern unter günstigen Witterungsbedingungen bessere Ergebnisse zu erwarten.

Ausblick

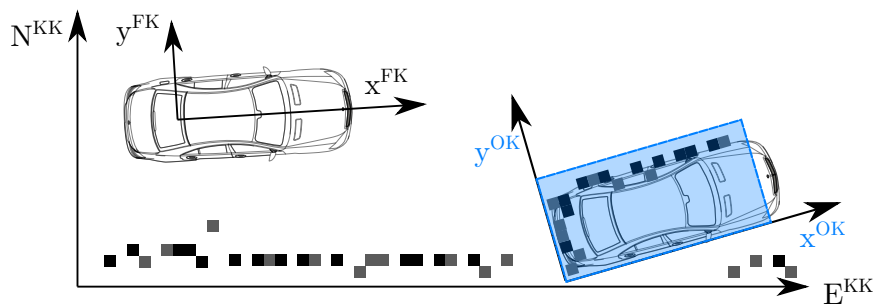
- Für die Evaluierung der Verfolgung mehrerer Objekte stand leider keine Referenzplattform zur Verfügung. In zukünftigen Arbeiten wäre eine Evaluierung der Verfolgung mehrerer Objekte von Interesse, insbesondere könnten hiermit unterschiedliche Ansätze zur Objektverfolgung realitätsnah verglichen werden.
- In dieser Arbeit wurden Verdeckungen von statischen Objekten durch dynamische Objekte nicht betrachtet. Statische Objekte erscheinen demnach durch Verdeckungen teilweise als dynamisch und führen sowohl in der Zustandsschätzung als auch in der statischen Karte zu Fehlern. Eine Modellierung von Verdeckungen ist daher unerlässlich.
- Eine besondere Schwierigkeit stellt die Objektverfolgung dar, wenn die Klasse des zu verfolgenden Objektes nicht bekannt ist. So können keine spezifischen Bewegungsmodelle und Modelle zu den geometrischen Dimensionen eingesetzt werden. Nach der Bestätigung für ein Objekt sollte demnach in zukünftigen Arbeiten eine Objektklassifikation mit einer Anpassung des Objekt- und Bewegungsmodells untersucht werden. Ein Grund für den großen Fehler von 1,1 m in der Positionsbestimmung ist, dass die Referenzposition jeweils zur Hinterachse angegeben ist, wohingegen die Messungen des Laserscanners auf der Objektaußenkontur liegen. Bereits ein einfaches rechteckiges Objektmodell würde diesen Fehler deutlich reduzieren.
- In diesem Kapitel wurde das CV-Modell mit konstanter Geschwindigkeit zur Verfolgung von Verkehrsteilnehmern untersucht. In zukünftigen Arbeiten könnte neben einer Objektklassifikation und spezifischen Bewegungsmodellen weiterhin die Leistungsfähigkeit einer Verfolgung durch mehrere Modelle parallel in einem IMM-Filter untersucht werden.
- Zur Evaluierung mit Referenzsensorik standen lediglich Messungen des Velodyne Laserscanners zur Verfügung. Die Verfahren sollten weiterhin mit Sensoren evaluiert werden, die verdeckt verbaut werden können, wie beispielsweise dem Ibeo LUX (Abschnitt 2.4.2).
- In dieser Arbeit wurden bewegte Objekte durch Mengenoperationen auf Belegungskarten detektiert. Moras [87] beschreibt die Detektion bewegter Objekte mit Hilfe der DST,

jedoch keine Objektverfolgung. Die Arbeit von Moras könnte mit der dreidimensionalen hierarchischen Karte und der in diesem Kapitel vorgestellten Variante des Partikelfilters kombiniert werden. Eine besondere Herausforderung stellt hier die Unterscheidung dar, ob es sich bei inkonsistenten Zellen um ein bewegtes Objekt handelt oder ob lediglich die Diskretisierung der Karte zu gering ist, wie in Abschnitt 3.6.2 beschrieben.

- Die detektierten dynamischen Zellen wurden durch den in diesem Kapitel beschriebenen Ansatz lediglich zur Objektverfolgung genutzt. Die dynamischen Zellen könnten jedoch in objektlokalen Karten [30] akkumuliert werden, um Objektkonturen über die Zeit genauer und vollständig zu beschreiben. Abbildung 5.17 zeigt ein Beispiel des Nutzens von objektlokalen Karten. Ein Objekt wird nach der Detektion einer Bewegung in einer objektlokalen Karte mit dem Koordinatensystem OK geführt. Eine Herausforderung wäre die Akkumulierung der dreidimensionalen Geometrie von Verkehrsteilnehmern in objektlokalen Belegungskarten.



(a) Statisches Szenario mit einem parkenden Fahrzeug.



(b) Verfolgung eines Objektes mit einer objektlokalen Bewegungskarte.

Abbildung 5.17 – Ein parkendes Fahrzeug, das sich länger nicht bewegt hat, wird der statischen Karte zugeordnet (a). Fährt das Fahrzeug los, so können die Belegungswahrscheinlichkeiten aus der statischen Karte nach der Detektion der Bewegung in einer objektlokalen Karte weitergeführt werden, um dessen Abmaße genauer schätzen zu können (b).

6

Weitere Applikationen

In den vorigen Kapiteln wurden bereits zwei Applikationen beschrieben: Die Schätzung der Pose und der Dynamik des Sensorfahrzeuges in Kapitel 4 und die Verfolgung von Verkehrsteilnehmern in Kapitel 5. Beide Applikationen basieren auf der dreidimensionalen Umgebungsmodellierung, die in Kapitel 3 vorgestellt wurde.

Weitere Anwendungen, die sich mit der vorgeschlagenen Belegungskarte realisieren lassen sind Fahrbahnverlaufsschätzung, Freiraumanalyse, Detektion und Vermessung von Parklücken sowie Höhenschätzung. In diesem Kapitel werden zwei dieser Anwendungen herausgegriffen und näher untersucht und ausgewertet: Die Detektion und Vermessung von Parklücken sowie die Schätzung der Höhe von Brücken.

Kapitelübersicht

In Abschnitt 6.1 wird eine Höhenschätzung basierend auf der dreidimensionalen Belegungskarte aus Kapitel 3 präsentiert. Die Belegungskarte wird dabei unter anderem mit einem einzeiligen Laserscanner aufgebaut. Abschnitt 6.2 stellt die Parklückendetektion und -vermessung vor. Hierfür kommen als Sensoren Nahbereichsradare, Laserscanner und eine Stereokamera zum Einsatz. Beide Anwendungen, sowohl die Höhenschätzung als auch die Parklückenvermessung, werden anhand mehrerer Messfahrten mit realen Sensordaten evaluiert. Das Kapitel endet nach Abschnitt 6.3 mit einer Zusammenfassung und einem Ausblick.

6.1 Höhenschätzung

In diesem Abschnitt wird die Höhenschätzung von überhängenden Strukturen, wie beispielsweise Brücken, mit realen Laserscannermessdaten vorgestellt. Damit wird eine Anwendung beschrieben, die ausschließlich auf einer echt dreidimensionalen Datenstruktur arbeiten kann.

6.1.1 Einleitung

Für ein System zur Längsführung, wie den Abstandsregeltempomaten (ACC), ist es wichtig, ob das eigene Fahrzeug unter Schilderbrücken, Brücken oder Tunneleinfahrten durchfahren kann. Ist die Umgebungserfassung nicht in der Lage, einem Objekt die korrekte Höhe zuzuordnen zu können, so stellt das System im schlechtesten Fall auch für eine Brücke eine mögliche Kollision fest und führt eine Notbremsung durch.

6.1.2 Stand der Technik

Diewald [24] beschreibt eine radarinterferenzbasierte Höhenschätzung. Die Genauigkeit der Höhenschätzung reicht aus, um ein Ziel als „Brücke“ beziehungsweise als „keine Brücke“ zu klassifizieren.

Pfeiffer [98] erweitert den Stixel-Ansatz, der einer abstrakten Repräsentation eines Tiefenbildes einer Stereokamera entspricht, um mehrere Ebenen. Einzelne Stixel erhalten daher eine Höhe, die zur Schätzung der Höhe einer Brücke eingesetzt werden kann.

6.1.3 Verfahren

In Abbildung 6.1 wird ein einfacher Algorithmus zur Höhenschätzung skizziert. In der dreidimensionalen statischen Belegungskarte $\mathbf{K}_{\text{static}}$ werden zwei Schritte durchgeführt: Im ersten Schritt wird vor dem Fahrzeug ein Hindernis gesucht. Die Höhe des Hindernisses spielt dabei keine Rolle. Im zweiten Schritt wird an der Position des Hindernisses von der Fahrbahnoberfläche aus nach oben nach belegten Zellen gesucht. Die Höhe, an der zum ersten Mal belegte Zellen auftreten, wird als Höhe des Objektes über der Fahrbahn interpretiert.

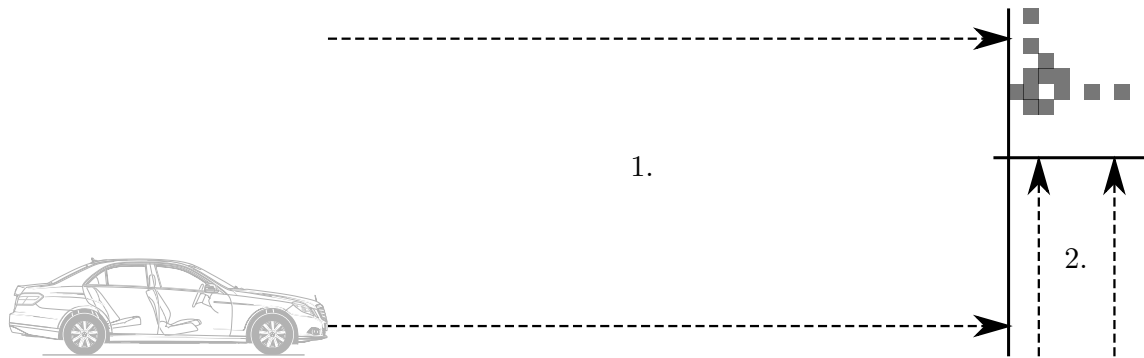


Abbildung 6.1 – Skizzierung des Algorithmus zur Höhenschätzung.

6.1.4 Eingesetzter Versuchsträger: E-Klasse.

Als Versuchsträger für die Höhenschätzung wird eine modifizierte E-Klasse [18] eingesetzt, bei der alle Sensoren verdeckt verbaut sind. Abbildung 6.2a zeigt ein Bild der E-Klasse auf dem Gelände des Forschungszentrums der Daimler AG in Ulm.



(a) Versuchsträger auf dem Gelände des Forschungszentrums in Ulm



(b) Zwei SICK LD-MRS im unteren Bereich der Stoßstange

Abbildung 6.2 – Versuchsträger E-Klasse Baureihe 212

Sensorkonfiguration

Das Sensorsetup besteht aus vier Serien-Nahbereichsradaren sowie zwei baugleichen Nahbereichsradaren, die zusätzlich angebracht wurden, um den Seitenbereich des Fahrzeuges beobachten zu können. Weiterhin sind zwei Sick LD-MRS, die baugleich zum Ibeo LUX sind, eingebaut (siehe Abschnitt 2.4.2). Abbildung 6.2b zeigt ein Bild der Laserscanner im

Versuchsträger. Die Einbaulagen und die resultierenden Sichtbereiche der Sensoren sind in Abbildung 6.3 in der Draufsicht dargestellt.

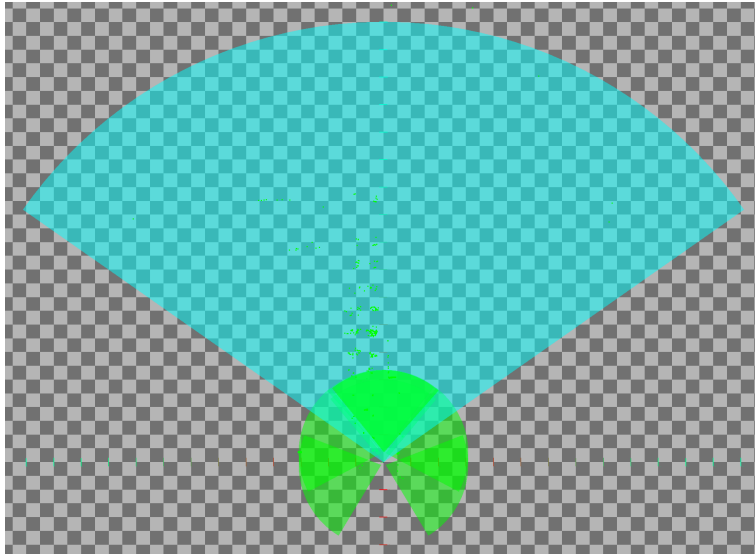


Abbildung 6.3 – Sensorsetup der E-Klasse. In Türkis ist der Sichtbereich des Sick LD-MRS eingezeichnet und in Grün die sechs Nahbereichsradare. Die Rasterung dient der Veranschaulichung, eine Zelle ist 5x5 m groß.

Kalibrierung

In dieser Arbeit wird nur der rechte der beiden Laserscanner verwendet. Zur Bestimmung der Einbaulage wird der Laserscanner zum Fahrzeug kalibriert. Die Halterung ermöglicht eine Justage in drei Freiheitsgraden. Es wird davon ausgegangen, dass die Angaben über die Elevation der einzelnen Messebenen relativ zum Gehäuse aus dem Datenblatt exakt sind, so dass es genügt, das Gehäuse des Sensors gegen das Fahrzeug zu kalibrieren. Das Gehäuse wird mit einem Laserabstandsmessgerät kalibriert, das mit einer Genauigkeit von $\pm 1,5$ mm arbeitet. Abbildung 6.4 zeigt das Vorgehen bei der Kalibrierung des Gehäuses des Laserscanners. Mit dem Laserabstandsmessgerät wird in Verlängerung der Blickrichtung des Sensors der Abstand d_1 zur Wand gemessen, weiterhin wird der Abstand d_2 ermittelt. Der Nickwinkel β ergibt sich somit zu $\beta = \tan^{-1} \frac{d_2}{d_1}$.

Der Laserscanner wurde speziell für Brückendetektionen um $3,2^\circ$ nach oben ausgerichtet, um Brücken sicher detektieren zu können. Der Laserscanner hat ein vertikales Sichtfeld von $\pm 1,6^\circ$. Eine fünf Meter hohe Brücke kann damit im Idealfall theoretisch in einem Abstand zwischen 56 m und 168,3 m detektiert werden. Bei einer Einbaulage mit einem Nickwinkel von 0° kann eine Brücke lediglich in Abständen größer als 168,3 m detektiert werden. Um eine universelle Einsatzfähigkeit von Laserscannern für mehrere Funktionen zu erreichen, sollten

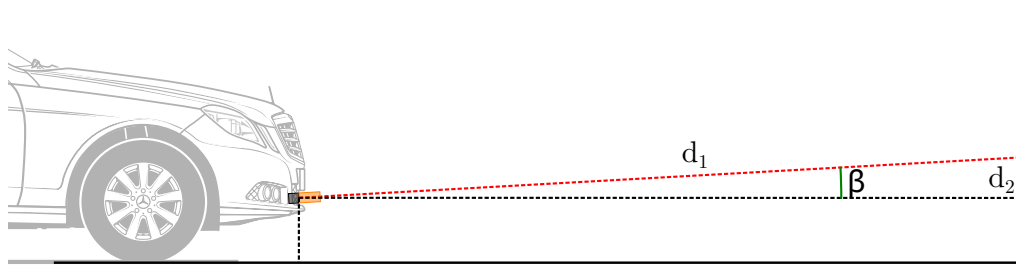


Abbildung 6.4 – Kalibrierung des Nickwinkels des Laserscanners.

daher zukünftig Sensoren mit einem erhöhten vertikalen Sichtbereich eingesetzt werden, wie zum Beispiel der prototypische Ibeo LUX 8L mit einem vertikalen Sichtbereich von $6,4^\circ$.

6.1.5 Ergebnisse

Die Auswertung erfolgt exemplarisch an zwei überhängenden Gebäuden und an zwei Brücken, um die Machbarkeit der Höhenschätzung über Belegungskarten zu demonstrieren. Die Belegungskarte wird dreidimensional aufgebaut mit einer konstanten Zellgröße von $(0,2\text{ m})^3$. Der Versuchsträger nähert sich in den Sequenzen jeweils der Brücke.

In Sequenz 1 handelt es sich um eine Brücke in Neu-Ulm. Die Referenzhöhe wurde mit einem hochgenauen Laserabstandsmessgerät ermittelt und beträgt $5,0\text{ m}$. Abbildung 6.5b zeigt das Bild der Dokumentationskamera. Die Höhe der Brücke kann erstmals in einer Distanz von 82 m geschätzt werden. Abbildung 6.5a zeigt das zugehörige Diagramm des Algorithmus zur Höhenschätzung. Die durchfahrbare Höhe wird während der Annäherung an die Brücke mit einem mittleren Fehler von $0,25\text{ m}$ geschätzt.

In Sequenz 2 handelt es sich um eine $4,5\text{ m}$ hohe Brücke in Neu-Ulm, siehe Abbildung 6.6b. Die Höhe der Brücke kann erstmals in einer Distanz von 98 m geschätzt werden. Abbildung 6.6a zeigt den Graphen für die geschätzte Höhe. Die Höhe wird mit einem mittleren Fehler von $0,52\text{ m}$ geschätzt.

In Sequenz 3 handelt es sich um die überhängende Struktur eines Gebäudes auf dem Gelände des Daimler Forschungszentrums in Ulm. Die wahre Höhe beträgt $3,761\text{ m}$. Abbildung 6.7b zeigt das Bild der Dokumentationskamera. In Abbildung 6.7a ist der zugehörige Graph für die Höhenschätzung dargestellt. Die Höhe des Gebäudes kann erstmals in einer Distanz von 77 m geschätzt werden, der mittlere Fehler beträgt hier $0,462\text{ m}$.

Bei der vierten und letzten Sequenz handelt es sich ebenfalls um ein Gebäude auf dem Gelände des Daimler Forschungszentrums, siehe Abbildung 6.8b. Die wahre Höhe beträgt hier

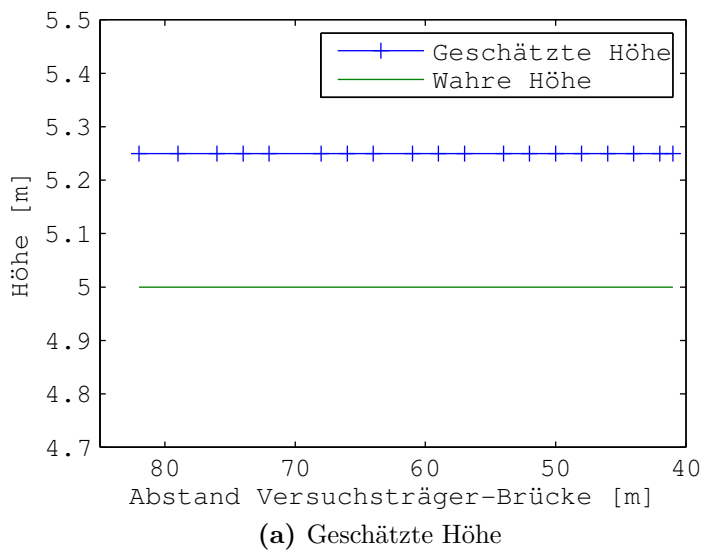


Abbildung 6.5 – Sequenz Nr. 1: Die Höhe wird während der Annäherung an die Brücke mit einem mittleren Fehler von 0,25 m geschätzt.

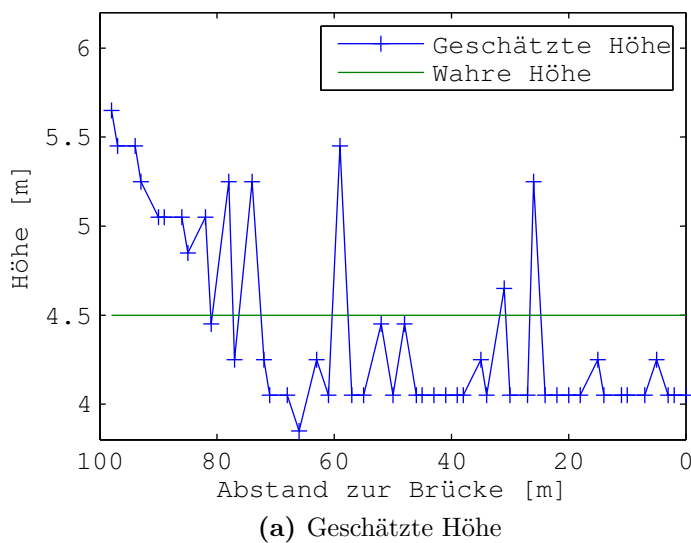


Abbildung 6.6 – Sequenz Nr. 2: Die Höhe wird mit einem mittleren Fehler von 0,52 m geschätzt.

4,5 m. Die Höhe des Gebäudes kann erstmals in einer Distanz von 92 m geschätzt werden. Abbildung 6.8a zeigt den zugehörigen Graphen des Algorithmus zur Höhenschätzung, der mittlere Fehler der Höhenschätzung beträgt 0,51 m.

Über die vier Beispielszenarien beträgt der mittlere Fehler in der Höhenschätzung 0,48 m. Zur Verringerung des Fehlers können folgende Maßnahmen ergriffen werden:

- Einbeziehen von Unebenheiten und Neigungen der Straßenoberfläche

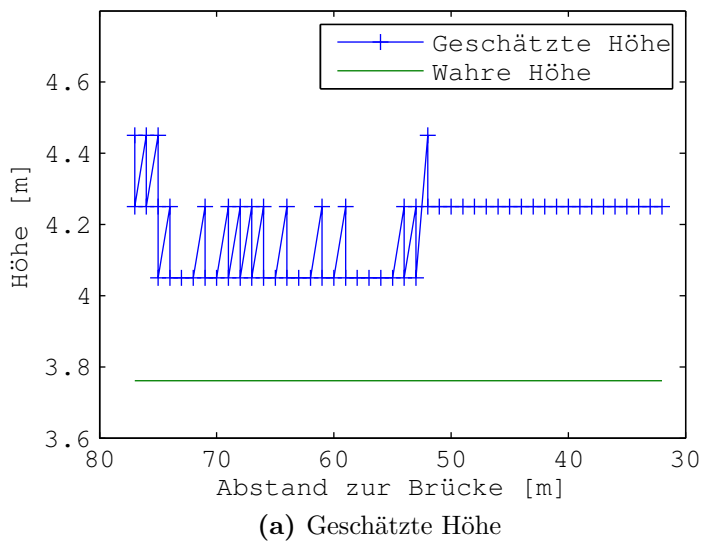


Abbildung 6.7 – Sequenz Nr. 3: Die Höhe wird mit einem mittleren Fehler von 0,46 m geschätzt.

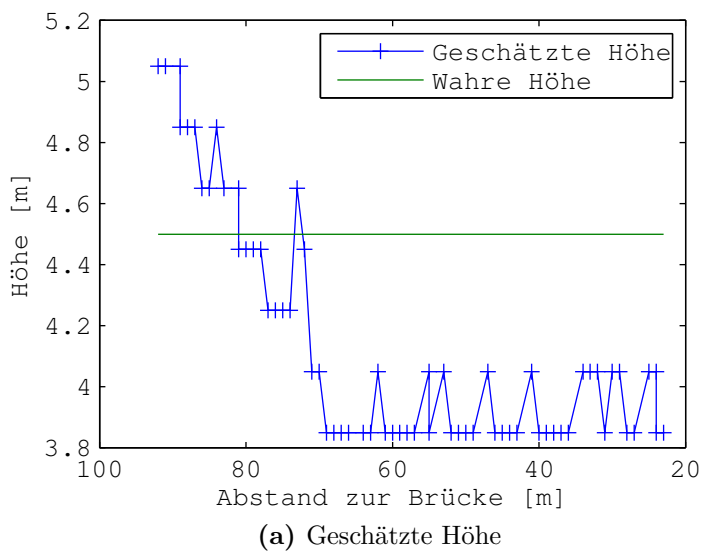


Abbildung 6.8 – Sequenz Nr. 4: Die Höhe wird mit einem mittleren Fehler von 0,51 m geschätzt.

- Erhöhung der Auflösung der Belegungskarte
- Definition einer Funktion zur Detaillierungssteuerung für Brücken, analog zu der in Abschnitt 3.8.1 beschriebenen Vorgehensweise.
- Exaktes Kalibrieren des Laserscanners über dessen Strahlkeulen wie in [90] beschrieben

6.1.6 Variante: Höhengschätzung mit einzeiligem Laserscanner

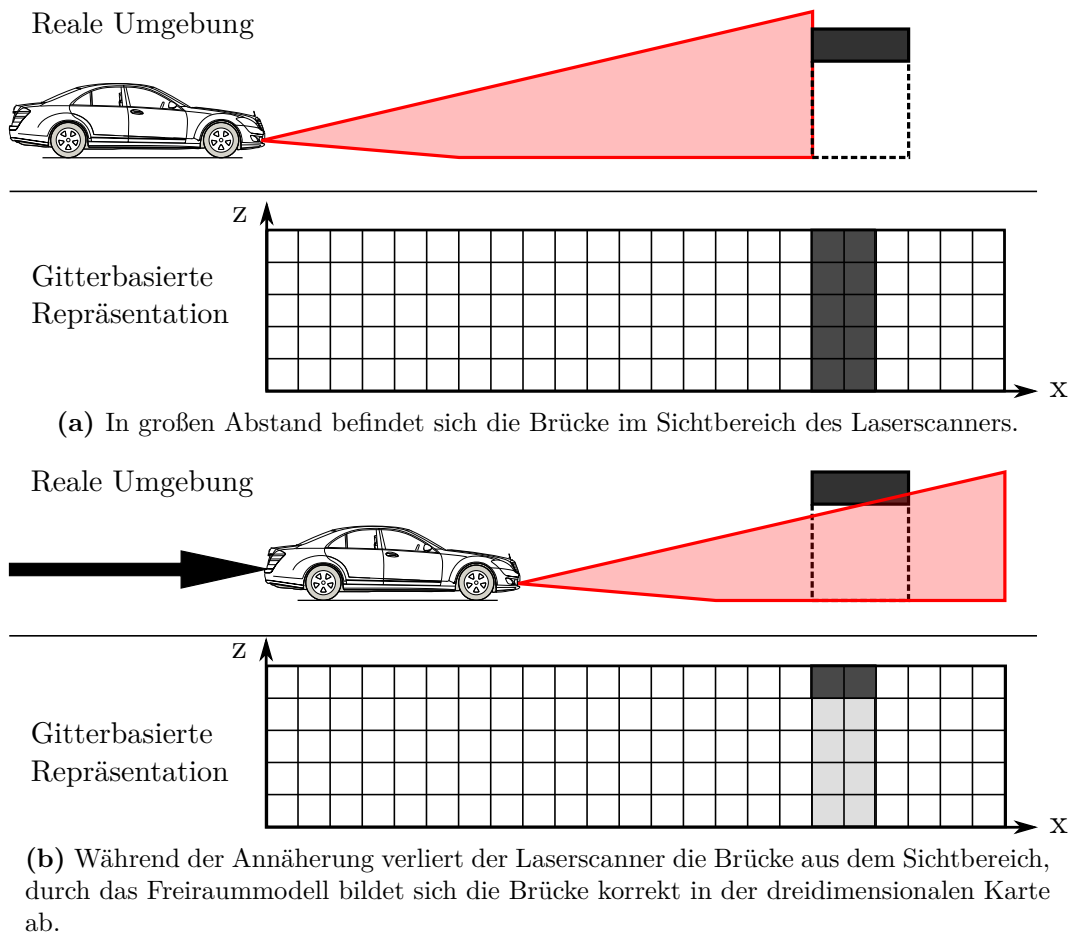


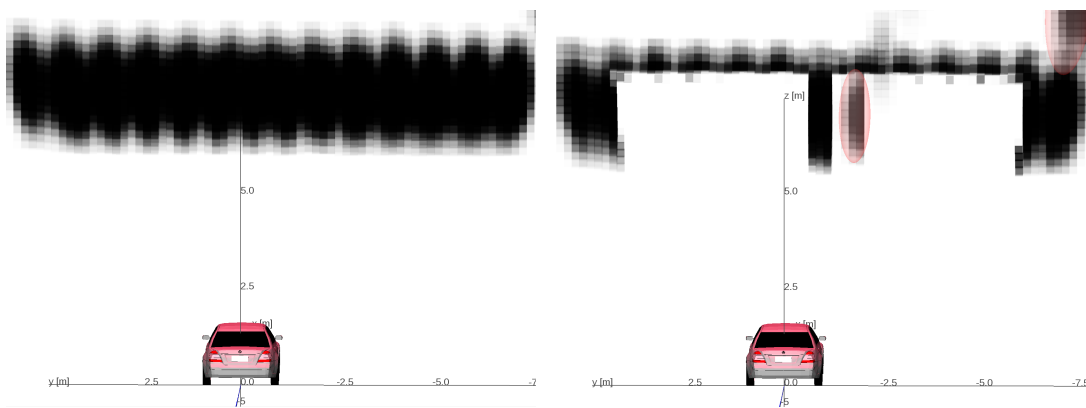
Abbildung 6.9 – Skizze der Höhengschätzung mit einem einzeiligen Laserscanner.

Selbst mit einem einzeiligen Laserscanner wie dem Hella IDIS, siehe Abschnitt 2.4.2, ist eine Höhengschätzung in dreidimensionalen Belegungskarten möglich. Abbildung 6.9 skizziert das Verfahren: Zunächst misst der einzeilige Laserscanner die Brücke an, siehe Abbildung 6.9a. Während der Annäherung an die Brücke verliert der Laserscanner die Brücke aus dem Sichtbereich. Das Freiraummodell des Laserscanners aktualisiert die Belegungskarte, damit wird der befahrbare Freiraum unter der Brücke in der dreidimensionalen Belegungskarte abgebildet, siehe Abbildung 6.9b.

Das Verfahren soll an einem realen Beispiel demonstriert werden. Der Versuchsträger fährt auf eine brückenähnliche überhängende Struktur zu. In Abbildung 6.10a ist das Referenzbild dargestellt. Es werden zwei unterschiedliche Situationen dieser Sequenz dargestellt. Solange die Distanz zur überhängenden Struktur über 50 m groß beträgt, befindet sich das Objekt im Sichtbereich des Laserscanners, damit werden in die Belegungskarte Messungen integriert. Die integrierten Messungen erscheinen aufgrund des inversen Sensormodells als Wand mit hoher Belegungswahrscheinlichkeit, siehe Abbildung 6.10b.



(a) Bild der Referenzkamera



(b) Dreidimensionale Karte mit Messungen der überhängenden Struktur. In einer großen Distanz wird die Struktur angemessen; das inverse Sensormodell führt zur gezeigten Belegungskarte.

(c) Dreidimensionale Belegungskarte mit Freiraummodellierung. In dem hier reduzierten Abstand zum Gebäude sorgen Freiraummessungen dafür, dass das Gebäude sich herausbildet.

Abbildung 6.10 – Modellierung einer überhängenden Struktur mit dem einzeiligen Laserscanner Hella IDIS. Die überhängende Struktur entwickelt sich in der dreidimensionalen Belegungskarte während der Annäherung des Versuchsträgers an das Objekt.

Sobald der Abstand unter 50 m fällt, verlässt das Objekt den Sichtbereich des Laserscanners. Die Freiraummessungen des Laserscanners, wie in Abschnitt 3.7.4 beschrieben, sorgen dafür, dass die Belegungswahrscheinlichkeit bereits belegter Zellen wieder sinkt. Alle Zellen, die außerhalb des neuen Sichtbereichs des Laserscanners liegen, bleiben jedoch unberührt. In einem Abstand von 30 m haben die Freiraummessungen den Bereich komplett gelöscht. Damit bleibt ein realistisches Modell der realen überhängenden Struktur, siehe Abbildung 6.10c. Die Zellgrößen sind in diesem Beispiel $(0,2\text{ m})^3$ groß.

6.2 Detektion und Vermessung von Parklücken

In diesem Abschnitt wird die Detektion und Vermessung von Parklücken vorgestellt und ausgewertet. Das vorgestellte Verfahren basiert auf der in Kapitel 3 vorgestellten Belegungskarte. Als Sensoren kommen Radarsensoren, ein Laserscanner und eine Stereokamera zum Einsatz.

6.2.1 Einleitung

Ford führte 2011 eine Studie [46] durch, die ergab, dass 32% der europäischen Autofahrer mehr als einen Versuch beim Einparken benötigen. Viele Fahrer empfinden beim Einparken zudem deutlichen Stress. Dies erklärt die hohe Nachfrage von Fahrern nach Parkhilfeassistenzsystemen. EurotaxGlass's and Harris Interactive führte 2011 eine Studie [40] zu der Verbaurrate von Parkhilfeassistenzsystemen durch. Europäische Fahrer, die in den letzten zwölf Monaten einen Neuwagen gekauft hatten, wurden befragt mit welchen Assistenzsystemen der neue Wagen ausgestattet ist. Parkhilfeassistenzsysteme wurden in 48% der Fälle genannt, bei Premiumfahrzeugen waren es sogar 59%.

Bei einem Parkhilfeassistenten handelt es sich um ein Komfortsystem: Das Fahrzeug soll automatisch in eine Parklücke fahren oder dem Fahrer soll über ein entsprechendes Lenkmoment eine Trajektorie vorgeschlagen werden. Hierfür sind die Detektion und die exakte Vermessung der Parklücke notwendig.

Ziel der in diesem Abschnitt vorgestellten Parklückendetektion und -vermessung ist es, die Parklücke möglichst früh zu detektieren und exakt zu vermessen. Im Unterschied zu heutigen Systemen soll der Fahrer daher nicht zuerst an einer Lücke vorbeifahren müssen, damit diese von der Sensorik vermessen werden kann. So kann das Fahrzeug frühzeitig abgebremst werden und der Einparkwunsch rechtzeitig durch Blinken dem nachfolgenden Verkehr signalisiert werden.

6.2.2 Stand der Technik

Scheunert [104] stellt eine Methode zur Parklückendetektion vor, basierend auf einer zweidimensionalen Belegungskarte und einer PMD (engl. Photonic Mixer Device)-Kamera. In jeder Zelle werden neben der geschätzten Belegungswahrscheinlichkeit die Rohdatenpunkte abgelegt, um einzelnen Zellen der Karte eine der folgenden Klassen zuzuweisen: Fahrbahn, niedriges Hindernis oder hohes Hindernis. Basierend auf diesen Daten wird die Bordsteinkante ermittelt. Aus den Zellen der Klasse „hohe Hindernisse“ wird anschließend ein Distanzprofil bezüglich der Bordsteinkante erstellt. Bei der Vermessung von 22 Parklücken wurde ein

mittlerer Fehler von 0,252 m ermittelt.

Suhr [120] detektiert Querparklücken über optischen Fluss. Als Sensor wird eine Rückfahrkamera eingesetzt. Dazu wird beim Rückwärtsfahren die Position der benachbarten parkenden Fahrzeuge geschätzt. Bei Testfahrten wurden 139 von 154 Querparklücken erkannt. Dies entspricht einer Erfolgsquote von 90,3%. Die Schwächen der Methode zeigen sich besonders bei ungünstigen Lichtverhältnissen und bei großen Entfernungen zur Parklücke.

Jung [67] detektiert Querparklücken mit einem Laserscanner, der am Heck angebracht ist. Eine Parklücke wird dabei nicht bei der Vorbeifahrt detektiert, sondern erst wenn das Fahrzeug sich bereits zum Einparken positioniert hat. Vermessen werden die Parklücken jedoch nicht. Von 112 Parklücken wurden 110 als solche detektiert, was einer Detektionsrate von 98,2% entspricht.

Abad [2] präsentiert ein Verfahren, das sowohl für Quer- als auch Längsparklücken geeignet ist. Die zwei Fahrzeuge, zwischen die eingeparkt werden soll, werden über senkrechte Rechtecke modelliert. Mit Ultraschallsensoren wird der laterale Abstand zu den parkenden Fahrzeugen bestimmt, während mit einer Rückfahrkamera und optischem Fluss die Eckpunkte der Fahrzeuge bestimmt werden. Aus diesen wird über ein Histogramm die Länge der Parklücke bestimmt. Die Abweichung der Parklückenlänge beträgt im Mittel 0,16 m.

6.2.3 Ansatz zur Parklückendetektion

Weiss [139] beschreibt die Fahrbahnverlaufsschätzung basierend auf einer zweidimensionalen Belegungskarte. Dazu werden von der Fahrzeuglängsachse aus Segmente nach links und rechts verschoben, bis diese auf belegte Zellen treffen. Dieser Ansatz wird für die Freiraumanalyse auf drei Dimensionen erweitert, um auf Basis der dreidimensionalen Belegungskarte K_{static} (5.2) arbeiten zu können. Auf Basis des Freiraums werden Parklücken anschließend detektiert und vermessen. Teilergebnisse des im Folgenden beschriebenen Ansatzes wurden bereits in [155] veröffentlicht. Zu dem Verfahren wurde ein Patent angemeldet [151].

Freiraumanalyse

Als Freiraum wird der Bereich um das Fahrzeug definiert, in dem keine belegten Zellen liegen. Zur Vereinfachung wird der Freiraum lediglich in einem gewissen Bereich bestimmt. Zunächst wird deshalb ein Intervall auf der Fahrzeuglängsachse definiert, innerhalb dessen der Freiraum bestimmt werden soll. Eine Schrittweite s bestimmt, wie groß die diskreten Schritte sind, in denen die Belegungskarte abgetastet wird.

Zur Bestimmung des Freiraums werden rechteckige Segmente S_j ausgehend von der x-z-Ebene des Fahrzeuges senkrecht nach außen verschoben, bis belegte Zellen in der statischen Belegungskarte $\mathbf{K}_{\text{static}}$ erreicht werden. Die Kontur des Freiraums wird demnach durch die Menge aller Segmente

$$\mathcal{P} = \{S_j\} \quad (6.1)$$

im Fahrzeugkoordinatensystem beschrieben. Abbildung 6.11 zeigt eine Beispielsituation, in der Segmente (dargestellt in Rot) in Richtung der Kontur der Parklücke verschoben werden.

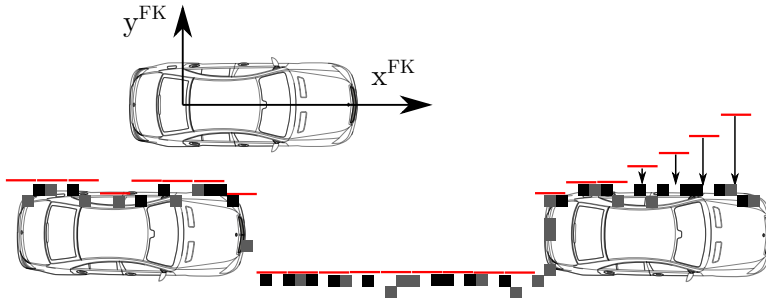


Abbildung 6.11 – Skizze eines parallelen Parkplatzes. Der Algorithmus zur Freiraumerkennung tastet die Kontur belegter Zellen der Belegungskarte über Segmente ab.

Die Position eines Segmentes ist für den Fall des rechten Fahrbahnrandes korrekt bestimmt, wenn der Bereich links des Segmentes frei und der Bereich rechts des Segmentes belegt ist. Um die konkrete Position eines Segmentes S_j zu bestimmen, wird ein Quader Q zu Hilfe genommen, der dem Segment entlang der y-Achse eine Ausdehnung verleiht. In diesem Quader kann die Belegungskarte anschließend auf Belegung überprüft werden.

Das Segment S_j ist im Fahrzeugkoordinatensystem durch die zwei Punkte $\mathbf{p}_{j,a}^{\text{FK}}$ und $\mathbf{p}_{j,b}^{\text{FK}}$ definiert. Das Segment erhält eine räumliche Ausdehnung, indem $\mathbf{p}_{j,a}^{\text{FK}}$ entlang der positiven y-Achse in Fahrzeugkoordinaten um einen konstanten Betrag verschoben wird. Der resultierende Quader wird mit $Q_{S_j}^{\text{FK}}$ bezeichnet. Dieser wird anschließend im Kartenkoordinatensystem betrachtet. Da der Quader dort nicht mehr achsenparallel ausgerichtet ist, wird stattdessen der kleinste achsenparallele umschließende Quader $Q_{S_j}^{\text{KK}}$ um $Q_{S_j}^{\text{FK}}$ betrachtet. In diesem werden alle Zellen n_i durch den in Abschnitt 3.5.4 beschriebenen Zugriff auf die Datenstruktur bestimmt. Die Mittelpunkte der Zellen werden über die Transformationsmatrix $\mathbf{T}^{\text{FK} \rightarrow \text{KK}^{-1}}$ zurück ins Fahrzeugkoordinatensystem transformiert, wo die Prüfung, ob ein Punkt innerhalb $Q_{S_j}^{\text{FK}}$ liegt, effizient erfolgen kann. Als Wahrscheinlichkeit, dass im Quader $Q_{S_j}^{\text{FK}}$ eine Belegung auftritt, wird das Maximum der Belegungswahrscheinlichkeiten aller Zellen bestimmt.

Ein Segment S_j startet dabei jeweils bei $y = 0$. Der Abstand von $\mathbf{p}_{j,a}^{\text{FK}}$ und $\mathbf{p}_{j,b}^{\text{FK}}$ auf der x-Achse beträgt exakt s . Übersteigt die Belegungswahrscheinlichkeit des zugehörigen Quaders $Q_{S_j}^{\text{FK}}$ eine gewisse Schwelle nicht, so wurde keine Belegung gefunden. In diesem Fall wird das Segment um einen konstanten Betrag z weiter an den Fahrbahnrand verschoben, bis

eine Belegung auftritt. An dieser Position wird das Segment festgehalten. Trifft das Segment innerhalb eines gewissen Abschnittes auf keine Belegung, so wird die Positionsbestimmung abgebrochen und ein definierter Maximalwert von beispielsweise 20 m gesetzt.

In [156] wurden Alternativen zur linearen Verschiebung des Quaders um einen konstanten Betrag z untersucht. Als am effizientesten hat sich folgende Vorgehensweise erwiesen: Die Suche nach belegten Bereichen wird zunächst mit einer großen Schrittweite von beispielsweise $z = 1$ m durchgeführt. Sobald eine Belegung auftritt, wird in diesem Bereich erneut mit einer Schrittweite von $z = 0,05$ m die genaue Position des Segmentes bestimmt.

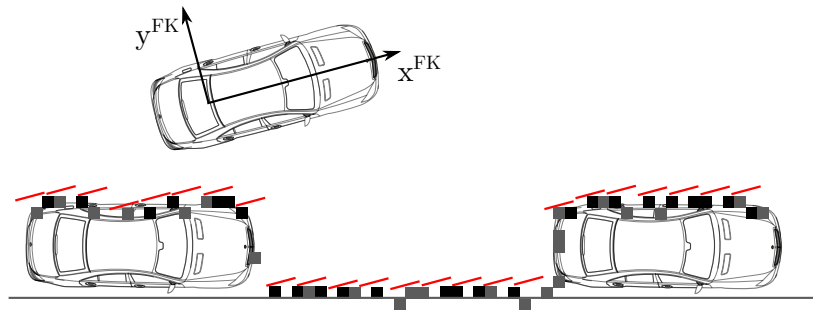


Abbildung 6.12 – Skizze eines parallelen Parkplatzes. Es kann nicht davon ausgegangen werden, dass die Parklücke parallel zur Fahrtrichtung des eigenen Fahrzeuges ausgerichtet ist. Infolge dessen geht durch die diskrete Abtastung der Umgebung durch die Segmente Genauigkeit verloren.

Wie in Abbildung 6.12 angedeutet ist, kann nicht davon ausgegangen werden, dass das eigene Fahrzeug parallel zu den parkenden Fahrzeugen ausgerichtet ist. Daher wird im nächsten Schritt die Orientierung der Parklücke bestimmt. Diese wird genutzt um ein Koordinatensystem zu bestimmen, welches parallel zur Parklückenorientierung liegt: Das Parklückenkoordinatensystem PK. Die Bestimmung der Orientierung erfolgt effizient auf der bereits bestimmten Kontur des Freiraums \mathcal{P} . Als Verfahren kommen ein Ransac-Algorithmus und ein Kleinster-Quadrate-Schätzer zum Einsatz, die dazu genutzt werden, eine Gerade l in die Segmentmenge einzupassen. Als Datenpunkte kommen jeweils die Mittelpunkte m der Segmente zum Einsatz, wobei für den Mittelpunkt des j -ten Segments gilt: $m_j = \frac{\mathbf{p}_{j,a}^{\text{KK}} + \mathbf{p}_{j,b}^{\text{KK}}}{2}$. Durch die Wahl der Mittelpunkte von Segmenten entsteht für die Positionierung der Geraden ein systematischer Fehler. Dies wird beispielsweise in Abbildung 6.12 deutlich: Die Segmente stoßen mit einem Endpunkt an belegte Zellen, die Mittelpunkte der Segmente haben jedoch noch einen gewissen Abstand zu den belegten Zellen. Dies ist im Rahmen einer konservativen Schätzung des freien Bereiches jedoch durchaus erwünscht.

Vorbereitend zur Parklückendetektion soll zudem der Abstand zu parkenden Fahrzeugen ermittelt werden. Bei einer langen Reihe parkender Fahrzeuge kann es jedoch vorkommen, dass die Mehrheit der Segmente auf den Fahrzeugseitenkonturen liegt oder aber auf dem Bordstein. Die durch den Ransac-Algorithmus bestimmte Gerade könnte daher zwischen den

Fahrzeugseitenkonturen und der Bordsteinkante hin- und herspringen. Daher wird das Verfahren erweitert, um zwei Geraden l_1 und l_2 einzupassen. Zwei Segmente liefern wiederum ein Modell für eine Gerade. Ein zusätzliches drittes Segment definiert eine zur ersten Geraden parallele Gerade. In Abbildung 6.13 sind beide Geraden veranschaulicht.

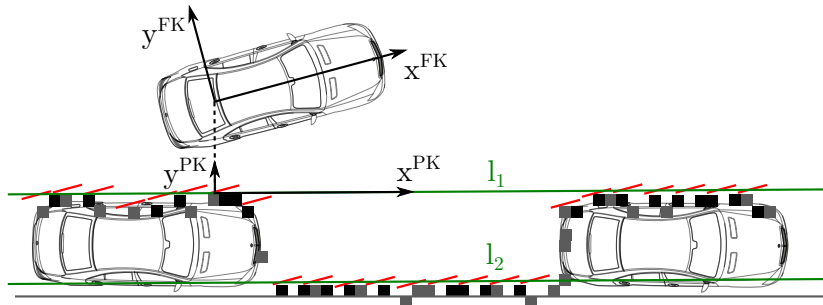


Abbildung 6.13 – Skizze eines parallelen Parkplatzes. Der Algorithmus zur Bestimmung der Orientierung der Parklücke passt zwei parallele Geraden l_1 und l_2 in die Segmente ein.

Die Gerade mit dem kleineren Abstand l_1 zum Fahrzeug wird als Modell für die Fahrzeugaußenkonturen interpretiert, wogegen die Gerade mit dem größeren Abstand als Bordstein interpretiert wird. Die Gerade entlang der Fahrzeugaußenkonturen l_1 wird als x-Achse des Koordinatensystems PK definiert.

Parklückendetektion

Auf Grundlage der Orientierung der parkenden Fahrzeuge wird eine potenzielle Parklücke gesucht. Entlang der x-Achse des Koordinatensystems PK wird ein Quader Q_P so lange über belegte Zellen bewegt, bis ein Freiraum gefunden wird. Abbildung 6.14 zeigt das Parklückenkoordinatensystem PK und den Quader.

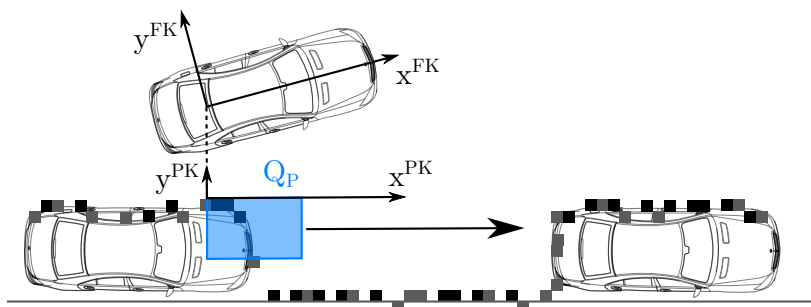


Abbildung 6.14 – Skizze eines parallelen Parkplatzes. Zur Detektion der Parklücke wird der Quader Q_P entlang der x-Achse von PK bewegt.

Anschließend wird der Quader so weit ausgedehnt, bis er seitlich als auch längs an belegte Zellen stößt. Hierfür sind zwei unterschiedliche Strategien möglich: Der Quader kann zunächst

entlang der x -Achse ausgedehnt werden, womit längere Parklücken gefunden werden, als wenn der Quader zunächst entlang der y -Achse ausgedehnt wird. Die passende Strategie ist je nach Anwendung zu wählen.

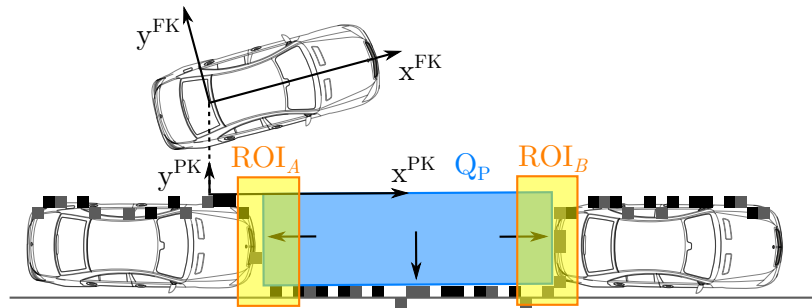


Abbildung 6.15 – Skizze eines parallelen Parkplatzes. Ist der Freiraum gefunden wird der Quader im zweiten Schritt ausgedehnt. Hat der Quader seine maximale Ausdehnung im Freiraum erreicht, werden durch dessen Enden die Bereiche ROI_A und ROI_B definiert.

Um die vordere und die hintere Parklückenbegrenzung wird jeweils ein Bereich von Interesse ROI (engl. Region of Interest) definiert: ROI_A und ROI_B . Abbildung 6.15 zeigt den ausgedehnten Quader und die Bereiche ROI_A und ROI_B .

Die Parklückenkonturbestimmung erfolgt nicht wie die Freiraumanalyse in Bezug auf das Fahrzeugkoordinatensystem FK, sondern in Bezug auf das Parklückenkoordinatensystem PK. Zur Ermittlung der Parklückenkontur wird zunächst eine große konstante Schrittweite s gewählt, um eine hohe Performanz zu erreichen. Die große Segmentbreite hat jedoch den Nachteil, dass bei der Konturbestimmung große Diskretisierungsfehler entstehen. Für die Ermittlung der Parklückenkontur werden daher die Segmente in den Bereichen ROI_A und ROI_B anschließend noch einmal feiner unterteilt. Diese dynamische Anpassung der Schrittweite führt zu einer hohen Performanz des Algorithmus, als auch zu einer präzisen Parklückenvermessung. Die Umgebung wird damit an den Ecken einer Parklücke in hoher Auflösung dargestellt, während der Rest in größerer Schrittweite abgetastet wird.

Abbildung 6.16 zeigt ein Beispiel eines detektierten Parkplatzes. Die konstante Segmentbreite s betrug hier zunächst 1,5 m und ist noch in den breiten Segmenten zu sehen. Anschließend erfolgte eine dynamische Verfeinerung, die in einer Segmentbreite von 0,08 m resultiert.

Ermitteln der Abmaße einer Parklücke

Für ein Parkhilfeassistenzsystem sind neben der Umgebungserfassung weitere Komponenten, wie die Trajektorienberechnung und -ausführung notwendig. Es wird eine einfache parametrische Darstellung des Parkplatzes gesucht, um eine abstrakte Schnittstelle von der Umgebungserfassung zur Situationsanalyse zu bieten. Abbildung 6.17 zeigt die gewählten

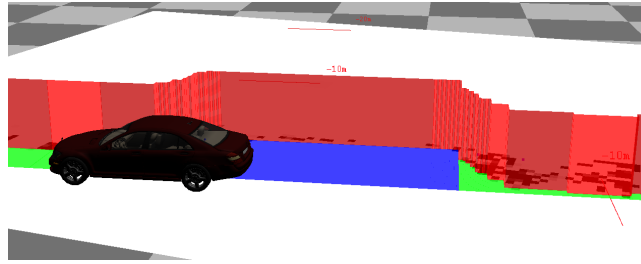


Abbildung 6.16 – Beispiel eines detektierten Parkplatzes. Die Kontursegmente sind alternierend hell- und dunkelrot eingefärbt. Die Ecken des Parkplatzes wurden mit feinen Segmenten der Länge 0,08 m abgetastet. Alle anderen Segmente sind deutlich größer, um eine hohe Performanz des Algorithmus zu gewährleisten. Die zu Grunde liegende Belegungskarte ist im Hintergrund sichtbar.

Parameter: Die vier Eckpunkte des Quaders Q_P in kartesischen Fahrzeugkoordinaten: P_1 bis P_4 .

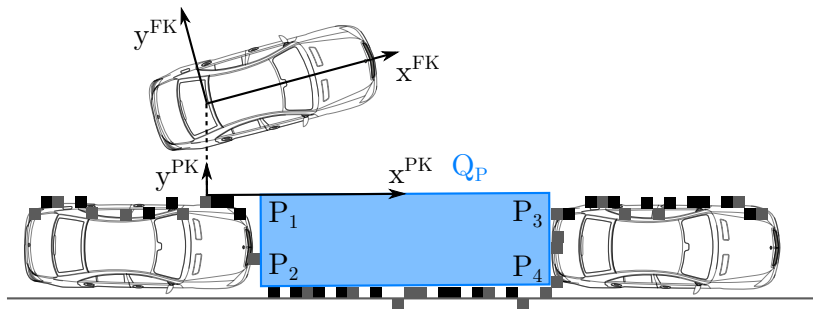


Abbildung 6.17 – Skizze mit den abstrakten Parametern P_1 bis P_4 einer detektierten Parklücke für die Situationsanalyse.

Um zu bestimmen, ob die Parklücke groß genug ist, werden zwei weitere Parameter bestimmt: Die Parklückenlänge l und die Parklückenbreite w . Um diese zu berechnen werden lediglich die Eckpunkte der Parklücke bezüglich des Parklückenkoordinatensystems PK betrachtet:

$$l = |P_4^{\text{PK}}.x - P_1^{\text{PK}}.x| \quad (6.2)$$

$$w = |P_4^{\text{PK}}.y - P_1^{\text{PK}}.y| \quad (6.3)$$

Dabei bezeichnet $P^{\text{PK}}.x$ die x-Koordinate eines Eckpunktes in Parklückenkoordinaten und $P^{\text{PK}}.y$ entsprechend die y-Koordinate.

Präzise Umgebungsrepräsentation im Innenraum von Gebäuden

In Tiefgaragen und Parkhäusern treten vereinzelt Elemente auf, die von der Decke hängen oder an der Wand befestigt sind und somit die Größe der Parkplätze einschränken. Beispi-



Abbildung 6.18 – Beispiel für eingeschränkte Parklücken in einem Parkhaus.

le für solche Elemente sind Belüftungsanlagen oder Rohre. Abbildung 6.18 zeigt ein reales Beispiel einer solchen Situation.

Ein eingeschränkter Parkplatz erscheint in einer zweidimensionalen Repräsentation in der Regel teilweise als belegt, obwohl das Fahrzeug unter gewissen Voraussetzungen tatsächlich hineinpasst. Der in Abbildung 6.19 gezeigte PKW passt beispielsweise vorwärts in die Parklücke, rückwärts jedoch nicht.

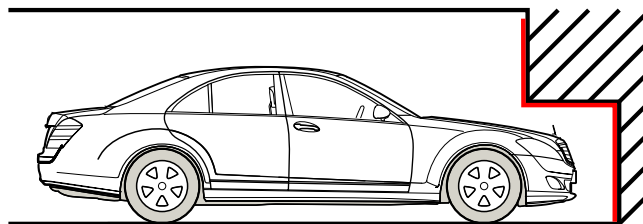


Abbildung 6.19 – Dreidimensionale Struktur einer eingeschränkten Parklücke in einem Parkhaus.

Durch eine dreidimensionale Beschreibung des eingeschränkten Parkplatzes, zusammen mit der dreidimensionalen Kontur des eigenen Fahrzeuges, kann dieser Sachverhalt korrekt dargestellt werden. Die Segmente S können genutzt werden, um die Umgebung in mehreren Höhengritten diskret abzubilden. Dazu wird die in Abschnitt 6.2.3 beschriebene Freiraumanalyse in mehreren diskreten Höhengritten ausgeführt. Die rote Linie in Abbildung 6.19 zeigt schließlich eine Darstellung der Situation, die mit dem vorgestellten Ansatz beschrieben werden kann.

6.2.4 Adaption des Detaillierungsgrades für die Belegungskarte

Bei der Parklückendetektion auf Basis einer Belegungskarte ist es von Vorteil, wenn diese in einer hohen Auflösung vorliegt. Wenn die grobe Parklückenkontur bekannt ist, genügt es, die Kontur der Parklücke hochaufgelöst abzubilden. Es muss nicht die gesamte Szene in hoher Auflösung repräsentiert werden.

Die dynamische Detaillierung der Karte, wie in Abschnitt 3.8.1 beschrieben, bietet die Möglichkeit, lediglich entlang der Kontur der Parklücke eine hohe Auflösung zu verwenden. Alle anderen Bereiche, die für eine Parkfunktion nicht relevant sind, werden in geringerer Auflösung repräsentiert.

Der Detaillierungsgrad für eine Parklücke PL wird durch die Funktion LOD_{PL} bestimmt. Die Konstante c_{PL} beschreibt den Abstand zur Kontur der Parklücke, in dem die Belegungskarte verfeinert wird.

$$\text{LOD}_{\text{PL}}(\mathbf{p}) = \begin{cases} c_{\text{max}}, & \min |\text{dist}(\mathbf{p}, S_i)| < c_{\text{PL}} \quad \forall S_i \in \mathcal{P} \\ c_{\text{max}} - 2, & \text{sonst} \end{cases} \quad (6.4)$$

$\forall \mathbf{p} \in \mathbb{R}^3$

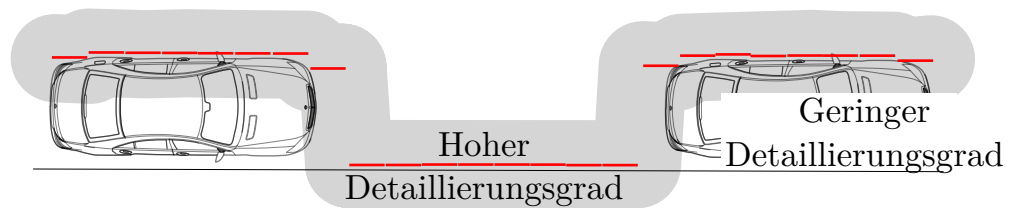
Die Menge \mathcal{P} enthält dabei eine Menge von Punkten S_i auf der Parklückenkontur. Die Bestimmung der Parklückenkontur wird in Kapitel 6.2.3 beschrieben.

Abbildung 6.20a zeigt eine skizzierte Parklücke. Die Parklückenkontur wird über die Menge der roten Kontursegmente \mathcal{P} dargestellt. Die graue Markierung definiert den Bereich, in dem die Distanz eines Segmentes S_i unter eine Distanz von 1 m fällt. Der Detaillierungsgrad wird in diesem Bereich zu c_{max} gewählt, wohingegen andere Bereiche gröber dargestellt werden.

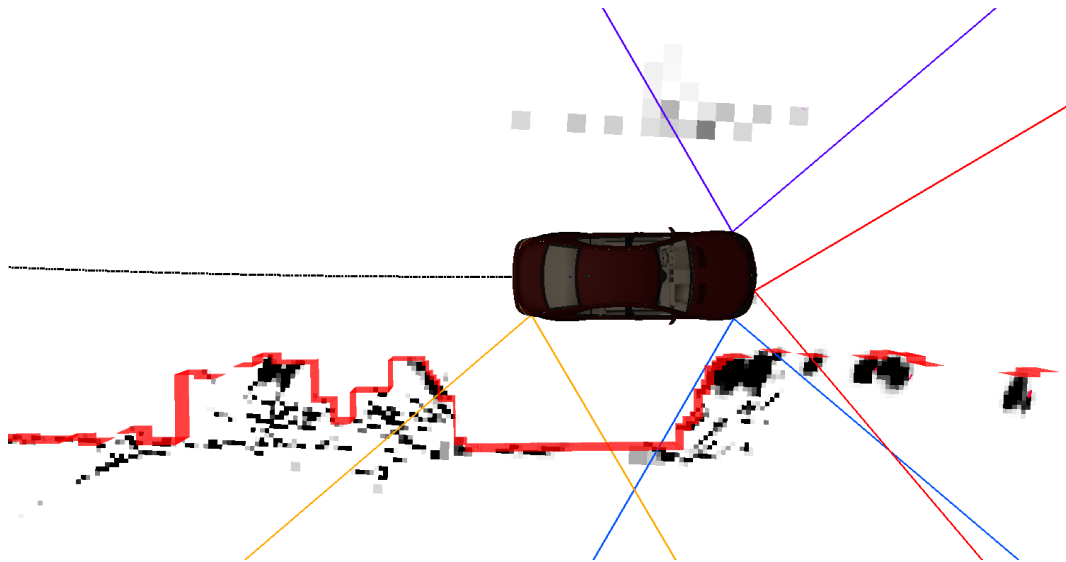
Abbildung 6.20b zeigt die Ergebnisse einer realen Messung, die durch Nahbereichsradare entstanden ist. Die minimale Seitenlänge einer Zelle ist zu 0,1 m gewählt. Durch die Anwendung von Gleichung (6.4) wird der Bereich nahe der detektierten Parklückenkontur durch $(0,1 \text{ m})^3$ große Zellen hoch aufgelöst. Andere Bereiche werden durch deutlich gröbere Würfel der Größe $(0,4 \text{ m})^3$ repräsentiert.

6.2.5 Eingesetzter Versuchsträger: A-Klasse

Ein prototypischer Versuchsträger, der mit einigen zusätzlichen Sensoren ausgerüstet ist, wurde auf Basis einer A-Klasse [17] aufgebaut. Der Versuchsträger ist in Abbildung 6.21 zu



(a) Skizzierung der Detaillierungssteuerung für Parklücken anhand eines Beispiels.



(b) Hier wird Funktion zur Detaillierungssteuerung, Gleichung (6.4), auf reale Messdaten angewandt. Für jede neu zu integrierende Messung wird überprüft, ob die Hierarchietiefe des Octrees erhöht werden muss. So werden Messungen im Bereich der Parklückenkontur mit hoher Auflösung in der Karte repräsentiert. Zu Beginn, wenn noch keine Parklücke detektiert wurde, werden alle Messungen in niedriger Auflösung eingetragen. Zusätzlich werden die Einbaulagen der Nahbereichs-Radarsensoren über deren Sichtbereiche farbige dargestellt: Die violette Markierung kennzeichnet den Sichtbereich nach links, die rote Markierung nach vorn, die blaue Markierung vorne nach rechts und die gelbe Markierung den Sichtbereich hinten nach rechts.

Abbildung 6.20 – Steuerung der Auflösung in der Belegungskarte anhand der Funktion zur Detaillierungssteuerung, wie in Gleichung (6.4) beschrieben. Für eine Parkfunktion werden alle Zellen in einem definierten Abstand zur detektierten Lücke in hoher Auflösung dargestellt, wohingegen andere Bereiche in geringer Auflösung dargestellt werden.

sehen.

Das prototypische Sensorsetup beinhaltet acht Nahbereichsradare (siehe Abschnitt 2.4.1), einen Ibeo LUX Laserscanner (siehe Abschnitt 2.4.2) und eine Stereokamera (siehe Abschnitt 2.4.3). Die Einbaulagen und die resultierenden Sichtbereiche der Sensoren sind in Abbildung 6.22 in der Draufsicht dargestellt. Die Kalibrierung der Sensoren erfolgte analog zu der in Abschnitt 6.1.4 beschriebenen Kalibrierung.



Abbildung 6.21 – Versuchsträger zur Evaluierung der Parklückenvermessung.

Bei der eingesetzten Stereokamera handelt es sich um ein prototypisches System. Die Basisbreite beträgt 0,26 m. Der Öffnungswinkel der Stereokamera beträgt in etwa 57°. Auf dem CAN-Bus werden Stixel mit 25 Hz bereitgestellt, eine Messung enthält dabei jeweils 50 Stixel.

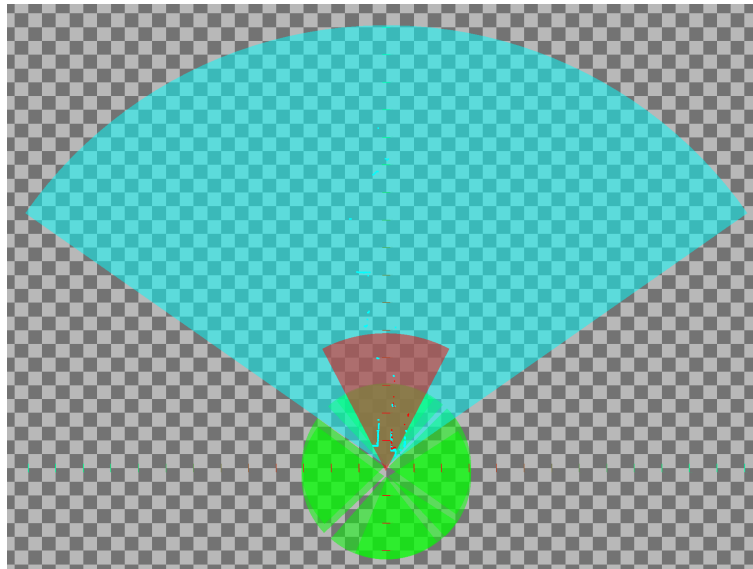


Abbildung 6.22 – Sensorsetup der A-Klasse. In Türkis ist der Sichtbereich des Ibeo LUX eingezeichnet, in Grün die acht Nahbereichsradare und in Rot die Stereokamera. Die Rasterung dient der Veranschaulichung. Eine Zelle ist 5 m x 5 m groß.

6.2.6 Ergebnisse

In diesem Abschnitt wird der vorgestellte Algorithmus anhand realer gemessener Parklücken evaluiert. Zum Parken wird die zweidimensionale Belegungskarte, die mit einem Quadtree realisiert wurde, eingesetzt.

Für sowohl die Nahbereichsradare, den Laserscanner als auch die Stereokamera erfolgt die Auswertung mittels unterschiedlicher Szenarien, die im realen Straßenverkehr aufgenommen

wurden, separat. Für jede Sensortechnologie wird zunächst eine exemplarische Sequenz vorgestellt, anschließend erfolgt eine Auswertung über mehrere Sequenzen. Zur Analyse der Performanz des Algorithmus werden unterschiedliche Zellgrößen betrachtet. Die Ergebnisse im folgenden Abschnitt sind auf demselben Rechner entstanden wie in Abschnitt 3.9.2 beschrieben.

Laufzeit

Abbildung 6.23 zeigt die Laufzeit der Parklückendetektion und -vermessung mit dem Ibeo LUX, abhängig von der eingesetzten Zellgröße der Belegungskarte und unter Verwendung der in Abschnitt 6.2.4 beschriebenen dynamischen Detaillierungsfunktion. Mit 0,04 m Auflösung der Karte K_{static} beträgt die maximale Rechenzeit über eine Sequenz, in der nacheinander mehrere Parklücken detektiert und vermessen werden, 273,3 ms. Bei 0,075 m Auflösung sinkt die maximale Laufzeit bereits auf 132,8 ms. Bei 0,1 m Auflösung beträgt die maximale Rechenzeit 91,7 ms, damit kann sowohl die abstrakte Parklücke als auch die exakte Parklückenkontur mit 10 Hz an die Situationsanalyse weitergereicht werden. Bei 0,4 m Auflösung schließlich stehen die Ergebnisse der Parklückendetektion und -vermessung mit 25 Hz zur Verfügung.

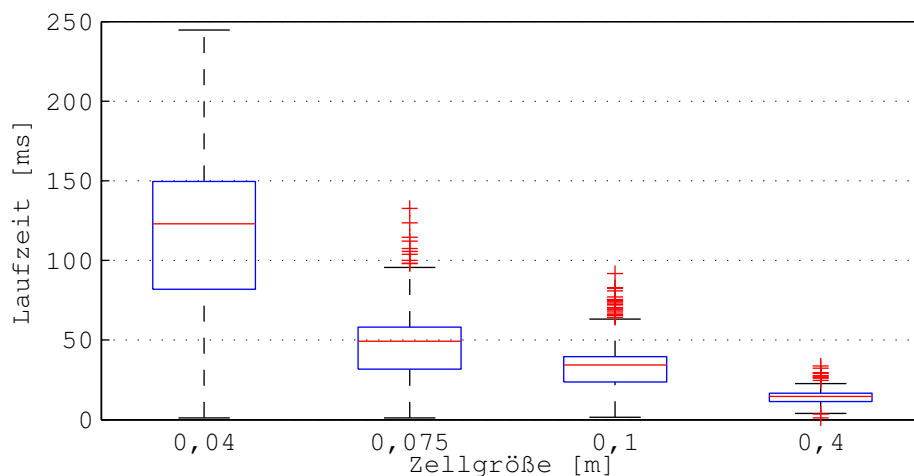


Abbildung 6.23 – Laufzeit der Parklückendetektion und -vermessung mit dem Ibeo LUX, abhängig von der eingesetzten Zellgröße unter Verwendung der Detaillierungsfunktion (6.4).

Ibeo LUX Laserscanner

Abbildung 6.24 zeigt exemplarisch die Längenschätzung einer detektierten Parklücke einer typischen Sequenz mit dem Ibeo LUX. Die Parklücke wird erstmalig in einer Distanz von 54,18 m detektiert, hier wird die Länge jedoch noch nicht richtig geschätzt. Betrachtet man

die geschätzten Längen jedoch erst unter einer Distanz von 35 m, so beträgt der mittlere Fehler nur 0,08 m.

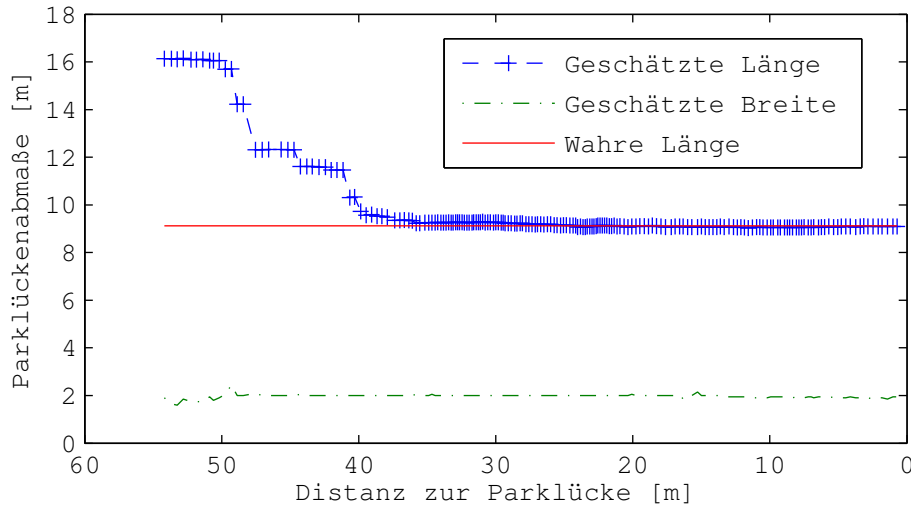


Abbildung 6.24 – Exemplarische Sequenz einer detektierten Parklücke mit dem Ibeo LUX.

Mit der A-Klasse wurden in Böblingen sechs parallele Parklücken am rechten Fahrbahnrand aufgezeichnet. Der Versuchsträger nähert sich in den Sequenzen jeweils einer Parklücke an. Die wahren Längen der Parklücken wurden mit einem hochgenauen Laserabstundmessgerät vermessen. Die Abweichungen in der Längenschätzung sind in Abbildung 6.25 als Box-Whisker-Plots dargestellt, aufgeschlüsselt nach Sequenz. Über alle Sequenzen beträgt der mittlere Fehler 0,12 m.

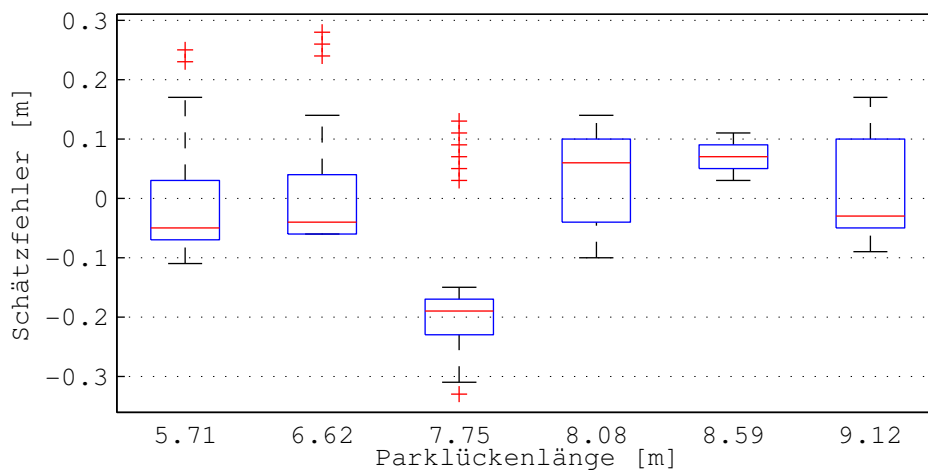


Abbildung 6.25 – Längenschätzung mit dem Ibeo LUX in der A-Klasse. Dargestellt sind Abweichungen in der Längenschätzung vom wahren Wert über 6 Sequenzen. Es werden nur Schätzungen in einem Abstand kleiner 35 m zur Parklücke betrachtet. Der mittlere Fehler über alle Sequenzen beträgt 0,12 m.

Nahbereichsradare

Abbildung 6.26 zeigt eine exemplarische Sequenz einer detektierten parallelen Parklücke mit den Nahbereichsradaren der A-Klasse. Die Parklücke wird erstmalig in einer Distanz von 12,1 m detektiert. Der mittlere Fehler beträgt in dieser Sequenz 0,38 m bei Schätzungen unter einer Distanz unter 10 m. Die geschätzte Länge nimmt nach dem Passieren der Lücke etwas ab: Die Winkelfehler der seitlichen Radarsensoren sorgen für eine scheinbare Verkürzung der Parklücke.

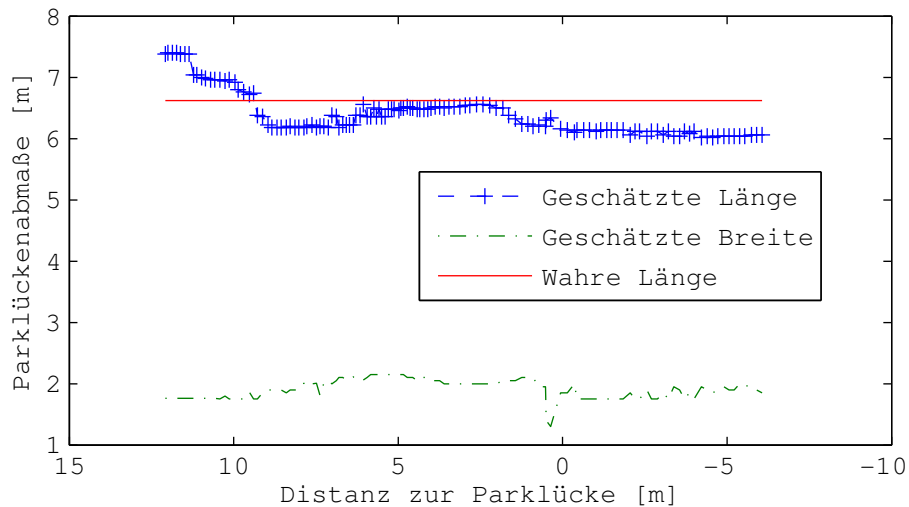


Abbildung 6.26 – Exemplarische Sequenz einer detektierten Parklücke mit den Nahbereichsradaren.

Die Abweichungen in der Längenschätzung mit den Nahbereichsradaren in Böblingen, aufgeschlüsselt nach Sequenz, sind in Abbildung 6.27 als Box-Whisker-Plots dargestellt. Über alle Sequenzen beträgt der mittlere Fehler 0,28 m.

Mit der E-Klasse wurden in Neu-Ulm 13 Sequenzen mit parallelen Parklücken im realen Straßenverkehr aufgezeichnet und ausgewertet. Abbildung 6.28 zeigt die Fehler in der Längenschätzung nach Sequenz als Box-Whisker-Plot. Der mittlere Fehler über alle Sequenzen beträgt hier 0,36 m.

Ebenfalls mit der E-Klasse wurden vier Querparklücken aufgezeichnet und ausgewertet. Das Verfahren zur Parklückendetektion und -vermessung funktioniert analog zu den parallelen Parklücken. Abbildung 6.29 zeigt eine exemplarische Sequenz einer detektierten Querparklücke mit den Nahbereichsradaren. Die Parklücke wird erstmalig in einer Distanz von 5,2 m detektiert. Die geschätzte Parklückenbreite schwankt um den wahren Wert. Nachdem das Sensorfahrzeug die Parklücke passiert hat, wird die Schätzung schlechter: Die hohen Winkelfehler des hinteren Nahbereichsradars lassen die Parklücke kürzer erscheinen, als sie ist. Der mittlere Fehler beträgt in dieser Sequenz 0,18 m. Abbildung 6.30 zeigt die Fehler in

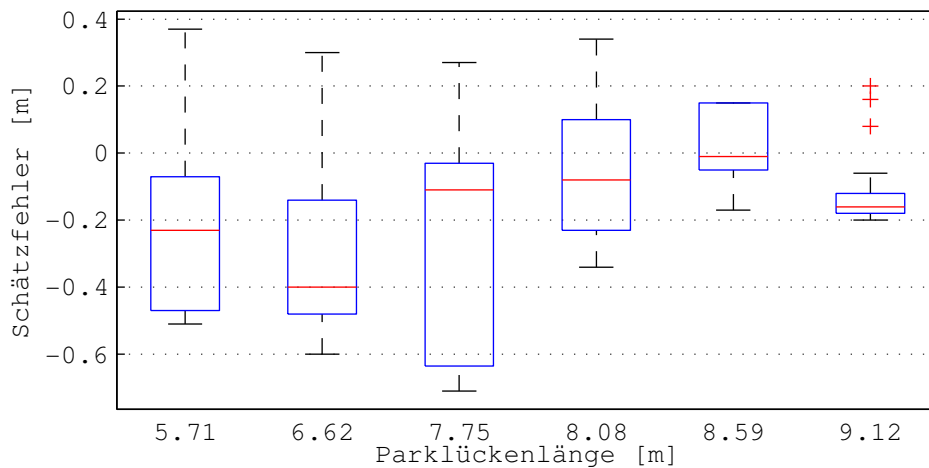


Abbildung 6.27 – Längenschätzung mit den Nahbereichsradaren in der A-Klasse. Dargestellt sind Abweichungen in der Längenschätzung vom wahren Wert über 6 Sequenzen. Es werden nur die geschätzten Längen in einem Abstand kleiner 10 m zur Parklücke betrachtet. Der mittlere Fehler über alle Sequenzen beträgt 0,28 m.

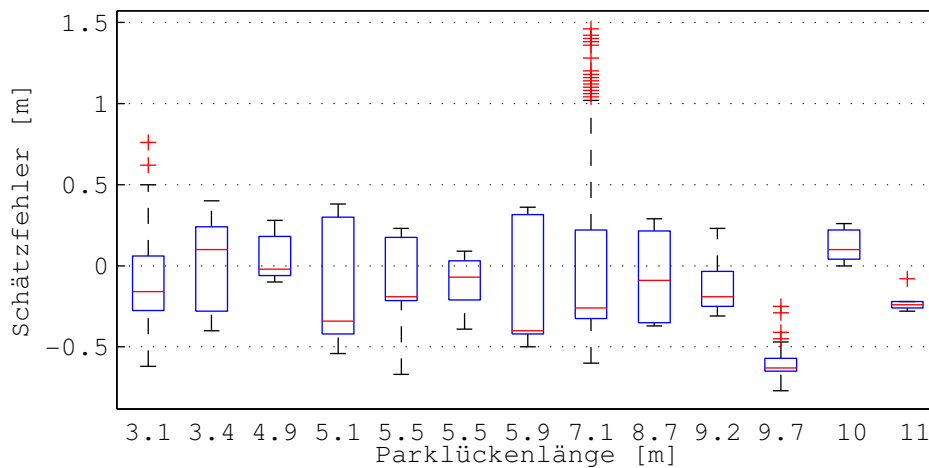


Abbildung 6.28 – Längenschätzung mit den Nahbereichsradaren in der E-Klasse. Dargestellt sind Abweichungen in der Längenschätzung vom wahren Wert über 13 Sequenzen. Es werden nur die geschätzten Längen in einem Abstand kleiner 10 m zur Parklücke betrachtet. Der mittlere Fehler über alle Sequenzen beträgt 0,36 m.

der Breitenschätzung je nach Sequenz als Box-Whisker-Plot. Der mittlere Fehler über alle Sequenzen mit Querparklücken beträgt 0,36 m.

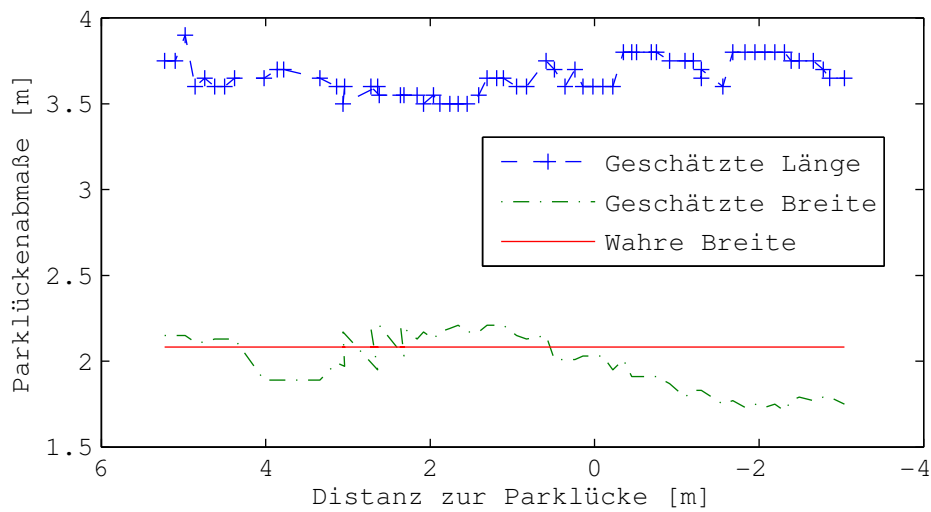


Abbildung 6.29 – Exemplarische Sequenz einer detektierten Querparklücke mit den Nahbereichsradaren.

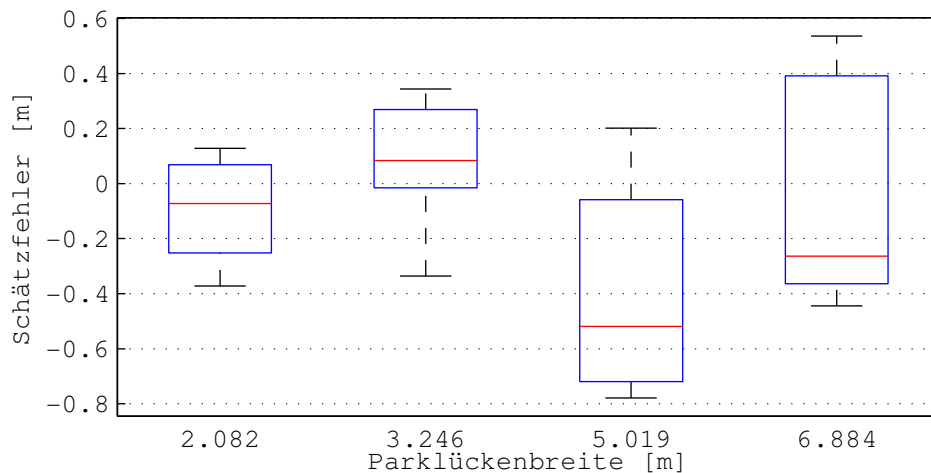


Abbildung 6.30 – Breitenschätzung mit den Nahbereichsradaren in der E-Klasse. Dargestellt sind Abweichungen in der Breitenschätzung vom wahren Wert über vier Sequenzen. Es werden nur die geschätzten Breiten in einem Abstand kleiner 10 m zur Parklücke betrachtet. Der mittlere Fehler über alle Sequenzen beträgt 0,36 m.

Stereokamera

Abbildung 6.31 zeigt eine exemplarische Sequenz einer detektierten Parklücke mit der Stereokamera im Versuchsträger. Die Parklücke wird erstmalig in einer Distanz von 19 m detektiert. Der mittlere Fehler beträgt in dieser Sequenz 0,14 m bei Schätzungen in einer Distanz unter 15 m.

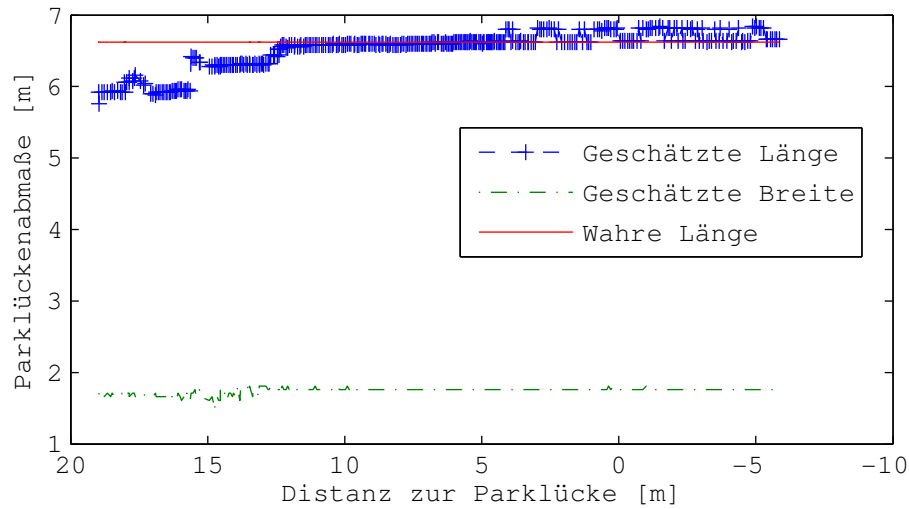


Abbildung 6.31 – Exemplarische Sequenz einer detektierten Parklücke mit der Stereokamera.

Für die Stereokamera im Versuchsträger existieren lediglich zwei Sequenzen. Abbildung 6.32 zeigt die Fehler in der Längenschätzung nach Sequenz als Box-Whisker-Plot. Der mittlere Fehler über beide Sequenzen beträgt 0,26 m.

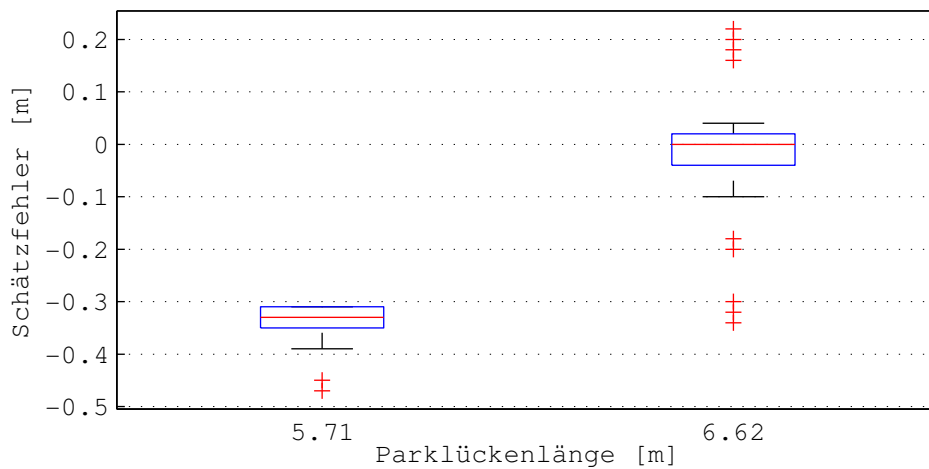


Abbildung 6.32 – Längenschätzung mit der Stereokamera im Versuchsträger. Dargestellt sind Abweichungen in der Längenschätzung vom wahren Wert über 2 Sequenzen. Es werden nur die geschätzten Längen in einem Abstand kleiner 15 m zur Parklücke betrachtet. Der mittlere Fehler über die Sequenzen beträgt 0,26 m.

Vergleich

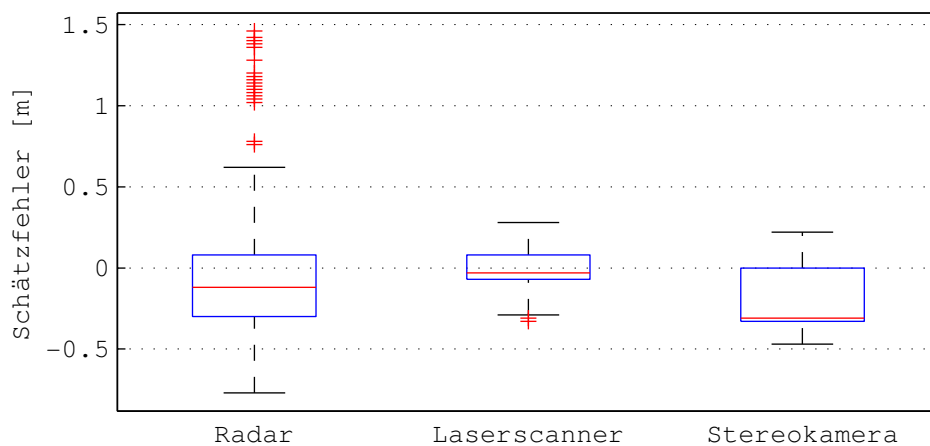
Zum direkten Vergleich werden die Sequenzen je nach Sensorik zusammengefasst. Mit dem Laserscanner sind es sechs Sequenzen, für unterschiedliche Konfigurationen von Nahbereichs-

Tabelle 6.1 – Mittlerer Fehler in der Längenschätzung paralleler Parklücken nach Sensorik.

Sensor	Sequenzen	RMSE [m]
Nahbereichsradare	19	0,32
Stereokamera (Stixel)	2	0,26
Ibeo LUX Laserscanner	6	0,12

radaren 19 Sequenzen und für die Stereokamera zwei Sequenzen. Es ist zu bemerken, dass die Länge von Parklücken mit dem vorgestellten Ansatz, unabhängig von der Sensorik, tendenziell eher etwas zu kurz als zu lang geschätzt wird. Im Sinne eines Parkhilfeassistenten ist eine konservative Schätzung der Parklückenlänge zu bevorzugen, um eine Kollision in jedem Falle zu vermeiden.

Abbildung 6.33 zeigt die Fehler in der Längenschätzung. Tabelle 6.1 schlüsselt die mittleren Fehler je nach Sensorik auf. Mit dem Laserscanner lassen sich Parklücken am genauesten vermessen. Mit den Nahbereichsradaren treten höhere Abweichungen auf, wodurch die Mindestparklückenlänge etwas zunimmt. Für die Stereokamera existieren leider zu wenige Sequenzen, so dass hier keine allgemeine Aussage getroffen werden kann. Das Potenzial zur genauen Parklückenvermessung ist jedoch vorhanden.

**Abbildung 6.33** – Abweichungen der Längenschätzung über alle Sequenzen mit parallelen Parklücken der unterschiedlichen Sensoren.

6.3 Zusammenfassung und Ausblick

Zusammenfassung

In diesem Kapitel wurden zwei neuartige Funktionen vorgestellt. Die Schätzung der Höhe von überhängenden Strukturen basierend auf einem Laserscanner, die im Besonderen von einer dreidimensionalen Belegungskarte profitiert sowie die Detektion und Vermessung von Parklücken, basierend auf unterschiedlichen Sensoren. Zur Realisierung und Evaluierung dieser Funktionen wurde auf zwei weitere Versuchsträger zurückgegriffen. Dies hat einerseits den praktischen Hintergrund, dass diese Versuchsträger mit unterschiedlichen Sensoren und Sensorkonfigurationen versehen sind. Andererseits demonstriert dies die Allgemeingültigkeit der Modelle und Algorithmen.

Bei der Höhenschätzung wurde mit einer dreidimensionalen Belegungskarte mit einer Auflösung von 0,2 m die Höhe von Brücken mit einer Genauigkeit von 0,48 m geschätzt. Diese Anwendung stellt lediglich exemplarisch die Möglichkeiten einer dreidimensionalen Karte dar. Die Möglichkeiten sind hier noch nicht ausgereizt.

Das Verfahren zur Parklückendetektion ist zur Detektion von sowohl Parallel- als auch Querparklücken geeignet und arbeitet mit unterschiedlichen Sensoren. Mit dem Laserscanner werden Parklücken bereits in Entfernungen von bis zu 50 m detektiert, ab einer Distanz von 35 m werden diese in der Länge mit einem mittleren Fehler von 0,12 m vermessen. Mit den Nahbereichsradaren werden diese in einer Distanz von 12 m detektiert, und ab einer Distanz von 10 m mit einem mittleren Fehler von 0,32 m in der Länge vermessen. Für die Stereokamera lässt sich aufgrund der zu geringen Stichprobe der aufgezeichneten Parklücken keine Aussage treffen, das Potenzial zur sehr genauen Parklückenvermessung ist jedoch vorhanden. Das Verfahren zur Detektion und Vermessung von Parklücken wurde zum Patent angemeldet [151].

Ausblick

- Zur Erhöhung der Genauigkeit der Höhenschätzung sollten Unebenheiten in der Fahrbahnoberfläche berücksichtigt werden.
- Neben den vorgestellten Sensoren für die Parklückenvermessung wäre eine Untersuchung des beschriebenen Verfahrens mit Ultraschallsensoren von Interesse.

7

Zusammenfassung und Ausblick

Zusammenfassung

Fahrerassistenz- und Sicherheitssysteme übernehmen immer umfangreichere Aufgaben in modernen Fahrzeugen und werden damit zunehmend komplexer. Ein wichtiger Faktor für die Zuverlässigkeit von solchen Systemen ist eine robuste, umfassende und hochgenaue Umgebungserfassung. Hohe Sicherheitsstandards erfordern komplementäre Sensorik, ein hoher Sichtbereich bedingt in der Regel ebenfalls mehrere Sensoren. Um ein konsistentes Gesamtbild der Umgebung zu liefern, werden die Daten mehrerer Sensoren häufig zu einem gemeinsamen Umgebungsmodell fusioniert.

Zweidimensionale Umgebungsmodelle haben den Nachteil, dass Brücken oftmals nicht als unterfahrbar oder Bordsteine nicht als überfahrbar erkannt werden. Aktuelle Forschungsarbeiten behandeln daher zunehmend dreidimensionale Repräsentationen in Bezug auf Fahrerassistenzsysteme [98, 11].

Je mehr Funktionen mit einer gemeinsamen Umgebungsrepräsentation bedient werden können und je unabhängiger diese von speziellen Sensoren ist, desto eher kann diese universell und baureihenübergreifend eingesetzt werden.

Aus diesen Anforderungen resultieren hohe Rechenzeit- und Speicheranforderungen für die Umgebungserfassung und Sensordatenfusion. Um diese bewältigen zu können, wurde in dieser Arbeit ein System zur Umgebungserfassung beschrieben, bei dem Anwendungen relevante Bereiche definieren können, in denen die Umgebungsmodellierung beeinflusst wird. So können erstmalig anwendungsrelevante Gebiete der Umgebung in hoher Auflösung repräsentiert werden, während weniger wichtige Bereiche in geringer Auflösung dargestellt werden. Realisiert wurde das System über eine zwei- und dreidimensionale hierarchische Belegungskarte. Dieses Verfahren wurde in [148] patentiert. In Abschnitt 3.6.2 wurde ein Verfahren beschrieben, um basierend auf der Unsicherheit des Belegungszustandes in der Belegungskarte die dynamische Detaillierung zu steuern. Die Unsicherheit im Belegungszustand wird mittels DST geschätzt. Das Verfahren wurde zum Patent angemeldet [152]. In Abbildung 1.1 wurde die Architektur des Systems zur Umgebungsmodellierung vorgestellt. Diese wurde exemplarisch umgesetzt und der prototypische Einsatz wurde mit unterschiedlichen Applikationen nachgewiesen.

In Abschnitt 3.9 wurden der Rechenzeit- und der Speicherplatzbedarf der Implementierungen ausgewertet. Die zweidimensionale Variante ist mit einer konstanten Auflösung von 0,1 m echtzeitfähig. Durch den Einsatz der Funktionen zur Detaillierungssteuerung wird auch eine Auflösung von 0,05 m echtzeitfähig. Die dreidimensionale Implementierung ist aufwändiger, hier ist eine Auflösung von 0,2 m echtzeitfähig. Unter Verwendung der Detaillierungssteuerung kann die Auflösung auf 0,1 m erhöht werden.

Gegenüber einer Implementierung mit einem dichten zweidimensionalen Datenfeld (engl. Array), welches die gesamte Karte abdeckt, gewinnt man mit der hierarchischen Implementierung in der ausgewerteten Testsequenz 54,9% Speicherplatz. Mit einer speziell für die Parklückendetektion ausgelegten Detaillierungssteuerung werden sogar 89,9% Speicherplatz weniger benötigt. Für die dreidimensionale Karte sind die Einsparungen noch größer: Gegenüber eines dichten dreidimensionalen Datenfelds benötigt die hierarchische Variante 83,2% weniger Speicherplatz, mit dem Fokus auf Parklücken können sogar 96,9% Speicherplatz eingespart werden.

In Kapitel 4 wurde eine rasterbasierte Selbstlokalisierung vorgestellt. In der zweidimensionalen Variante läuft die Selbstlokalisierung in Echtzeit. Für die in Abschnitt 4.6.3 beschriebene herausfordernde Sequenz lässt sich die Genauigkeit der Posenschätzung im Vergleich zur Odometrie deutlich verbessern. Die Odometrie liefert in diesem lokal begrenzten Szenario einen mittleren Fehler von 2,24 m zur Referenz. Mit der rasterbasierten Selbstlokalisierung bei 0,2 m Auflösung wird eine deutliche Verbesserung auf einen mittleren Fehler von 0,64 m erzielt. Auf der Landstraße stehen erwartungsgemäß deutlich weniger Objekte zur Selbstlokalisierung zur Verfügung. Der Fehler der Odometrie nach einer gefahrenen Strecke von 100 m beträgt 3,35 m, dieser konnte mit der Selbstlokalisierung auf 0,85 m reduziert werden.

Die dreidimensionale Selbstlokalisierung ist, durch die drei zusätzlichen Freiheitsgrade und die komplexere Karte, noch nicht echtzeitfähig. Im städtischen Szenario liefert die Odometrie einen mittleren Fehler von 2,24 m, welcher sich durch die rasterbasierte Selbstlokalisierung auf 1,5 m verbessern lässt. Auf der Landstraße während den ersten 100 m Strecke beträgt der mittlere Fehler der Odometrie 4,86 m der durch die Selbstlokalisierung immerhin auf 2,17 m reduziert werden kann.

Ein neues Verfahren, um einen dreidimensionalen Kartenabgleich durchzuführen, wurde in Abschnitt 4.7 vorgestellt. Das Verfahren profitiert von der dreidimensionalen Belegungskarte der Umgebung. Dafür wird eine digitale Karte mit Höheninformationen vorausgesetzt, wie sie beispielsweise über das ADASIS-Protokoll zur Verfügung gestellt werden kann. Das Verfahren wurde patentiert [150].

Das Erscheinungsbild von Verkehrsteilnehmern ist äußerst vielfältig: Die Kontur von PKW oder LKW kann durch ein quaderförmiges Modell gut angenähert werden, für Zweiradfahrer oder Fußgänger passt dieses Modell jedoch weniger. Weiterhin sind Positionsänderungen von motorisierten Fahrzeugen über Bewegungsmodelle gut vorhersagbar, Fußgänger hingegen können jederzeit spontan ihre Richtung ändern. Die Detektion und Verfolgung bewegter Objekte in Kapitel 5 arbeitet daher nahezu modellfrei, um den Anspruch zu erfüllen, möglichst viele unterschiedliche Objekte ohne vorherige Klassifikation verfolgen zu können.

Erstmalig in der Literatur wurden dabei bewegte Zellen in einer hierarchischen Karte detektiert. Die Detektion erfolgte durch Mengenoperationen auf Belegungskarten und führt zu einer Separierung von statischer und dynamischer Karte. Durch eine Zustandsschätzung dynamischer Objekte über ein Partikelfilter konnten auch durch Rauschen auftretende Artefakte weitestgehend eliminiert werden, um eine robuste statische Umgebungsrepräsentation für die Freiraumanalyse zu erzielen. Das Verfahren wurde mit zwei Roboterfahrzeugen der Daimler AG evaluiert. Es wurde ein Szenario mit äußerst schwierigen Bedingungen für einen Laserscanner zur Evaluierung gewählt: Bei Regen wird ein schwarz lackiertes Fahrzeug verfolgt, das ständig die Fahrtrichtung ändert. Als Verfahren wurde ein Partikelfilter mit einer angepassten Gewichtungsfunktion angewandt, wofür keine anwendungsabhängige und fehleranfällige Segmentierung der Messpunkte des Laserscanners notwendig ist.

Bei dem evaluierten Szenario zeigt sich ein mittlerer Fehler von 1,15 m in der Positionsbestimmung und ein Fehler von $0,57 \frac{\text{m}}{\text{s}}$ in der Geschwindigkeitsschätzung. Der hohe Fehler in der Positionsschätzung ist darauf zurückzuführen, dass weder das Objektmodell noch das Bewegungsmodell für einen PKW berücksichtigt wurde. In künftigen Arbeiten könnte daher eine Klassifikation von Verkehrsteilnehmern und das Anwenden von passenden Modellen zu einer Verringerung des Fehlers genutzt werden.

In Kapitel 6 wurden zwei weitere neuartige Funktionen vorgestellt: Die Schätzung der Höhe von überhängenden Strukturen basierend auf einem Laserscanner sowie die Detektion und

Vermessung von Parklücken.

Die Höhenschätzung arbeitet mit unterschiedlichen Laserscannern und profitiert im Besonderen von der dreidimensionalen Belegungskarte. Mit einer Kartenauflösung von 0,2 m wurde die Höhe von Brücken mit einem mittleren Fehler von 0,48 m geschätzt. Diese Anwendung stellt prinzipiell die Potenzial einer dreidimensionalen Karte dar, die Möglichkeiten sind damit aber noch nicht ausgereizt.

Das Ziel der bearbeiteten Parklückendetektion war es, parallele Parklücken möglichst früh während des Heranfahrens zu detektieren und zu vermessen. Dabei konnte auf unterschiedliche Sensoren zurückgegriffen werden. Mit einem Laserscanner wurden Parklücken bereits in Entfernungen von bis zu 50 m detektiert, ab einer Distanz von 35 m wurden diese in der Länge auch sehr genau vermessen: Der mittlere Fehler beträgt lediglich 0,12 m. Mit Nahbereichsradaren wurden Parklücken in einer Distanz von 12 m detektiert und ab einer Distanz von 10 m mit einer Genauigkeit von 0,32 m in der Länge vermessen. Das Verfahren war ebenso in der Lage, Querparklücken zu detektieren und zu vermessen. Weiterhin wurde die Parklückendetektion und -vermessung mit einer Stereokamera, basierend auf der Stixel-Repräsentation, betrachtet. Dieses Verfahren besitzt ebenso das Potenzial zur sehr genauen Vermessung von Parklücken, sollte jedoch noch in größerem Umfang evaluiert werden. Das Verfahren zur Detektion und Vermessung von Parklücken wurde zum Patent angemeldet [151].

Ausblick

An einigen Stellen der Untersuchungen wurden mögliche Erweiterungsfelder identifiziert, die im Rahmen dieser Arbeit jedoch nicht betrachtet wurden. In den Zusammenfassungen der einzelnen Kapitel wurde bereits auf spezielle weiterführende Themen eingegangen, die in diesem Abschnitt noch einmal zusammengefasst werden.

Die Applikationen, die im Rahmen dieser Arbeit präsentiert wurden, befinden sich noch in einem frühen Stadium. So ist die Höhenschätzung in der vorgestellten Form noch nicht einsatzfähig, demonstriert jedoch eindrucksvoll die Möglichkeiten einer vollständigen dreidimensionalen Umgebungsrepräsentation. Mitunter sollten Unebenheiten in der Fahrbahnoberfläche zur Erhöhung der Genauigkeit berücksichtigt werden. Im Rahmen der Evaluierung der Parkfunktion stellte sich heraus, dass die getroffenen Annahmen in Verbindung mit dem einheitlichen inversen Sensormodell für die hochgenaue Vermessung von Parklücken noch nicht zu einem optimalen Ergebnis führen. Durch die Betrachtung der statistischen Abhängigkeiten der Messungen könnte die Vermessung verbessert werden.

Die Belegungskarte wurde im Rahmen dieser Arbeit lediglich in Bezug auf ein einzelnes Sensorfahrzeug betrachtet. Im Kontext der Fahrzeug-Fahrzeug (V2V)-Kommunikation sind

fahrzeugübergreifende Belegungskarten denkbar. Dies würde einer umfassenderen Freiraumanalyse für einzelne Verkehrsteilnehmer dienen, die ohne Kommunikation und ohne digitale Straßennetzkarte in einer Kreuzung nicht feststellen können, welchen Verlauf die Fahrbahn hinter einem Gebäude nimmt. Des Weiteren würden durch eine fahrzeugübergreifende Belegungskarte Freiräume frühzeitig allen Verkehrsteilnehmern bereitgestellt. Dies hat den Vorteil, dass mögliche Parklücken oder die Durchfahrtshöhe von Brücken von jedem Fahrzeug einzeln evaluiert werden können, da sie von der jeweiligen Fahrzeugaußenkontur abhängen.

Der Belegungszustand wurde in dieser Arbeit durch die Wahrscheinlichkeitstheorie geschätzt. Eine Schätzung über die Evidenztheorie nach Dempster und Shafer, siehe dazu Abschnitt 3.6.2, hätte möglicherweise zwei Vorteile: Zum einen könnten inkonsistente Zellen dazu genutzt werden, die dynamische Zellverfeinerung zu steuern [152]. Zum anderen könnte die aufwändige Detektion bewegter Objekte durch Mengenoperationen auf Belegungskarten in Abschnitt 5.3 durch die DST gelöst werden, wie von Moras [87] gezeigt. Eine besondere Herausforderung stellt die Unterscheidung dar, ob es sich bei inkonsistenten Zellen um ein bewegtes Objekt handelt, oder ob lediglich die Auflösung der Karte zu gering ist.

Zur Vervollständigung des Ansatzes zur Selbstlokalisierung ist eine Kombination der vorgestellten rasterbasierten Variante mit einer punktlandmarkenbasierten Lokalisierung, wie von Weiss [139] beschrieben, denkbar. Damit sollte die Genauigkeit in Situationen, in denen geeignete Punktmerkmale zur Lokalisierung vorhanden sind, weiter erhöht werden. Das Hinzunehmen von Kontextwissen, wie zum Beispiel die aktuelle Straßenneigung, könnte die Genauigkeit weiter erhöhen.

Der in Abschnitt 4.7 vorgeschlagene dreidimensionale Kartenabgleich wurde nicht evaluiert. In zwei Dimensionen wird dieser Ansatz bereits erfolgreich angewandt [75, 109]. In drei Dimensionen stehen für den Abgleich mehr Merkmale zur Verfügung, von daher sollte eine Registrierung genauer erfolgen können. Eine Herausforderung stellt hier die Extraktion des dreidimensionalen Fahrbahnverlaufs aus der Belegungskarte dar.

Die Verfolgung bewegter Objekte erfolgt in dieser Arbeit lediglich mit einem einfachen einheitlichen Modell. Um eine größere Genauigkeit zu erhalten, ist eine Klassifikation in unterschiedliche Klassen von Verkehrsteilnehmern notwendig. Dies macht den Einsatz von unterschiedlichen Objekt- und Bewegungsmodellen möglich. Unterschiedliche Bewegungsmodelle könnten alternativ auch parallel mit einem IMM-Filter verfolgt werden. Die Evaluierung der Zustandsschätzung bewegter Objekte sollte in Folgearbeiten weiterhin mit Sensoren erfolgen, die verdeckt verbaut werden können, wie beispielsweise dem Ibeo LUX Laserscanner (siehe in Abschnitt 2.4.2). Ein weiterer Punkt, der bei der Detektion und Verfolgung bewegter Objekte noch nicht gelöst ist, ist die Verdeckung von Hintergrundobjekten durch Verkehrsteilnehmer. Verdeckungen wurden in dieser Arbeit nicht gesondert behandelt und führen gelegentlich zu Falschdetektionen. In zukünftigen Arbeiten sollten Verdeckungen modelliert und behandelt

werden, wie beispielsweise in [106] vorgestellt, um solche Fehler reduzieren zu können.

Die detektierten dynamischen Zellen werden durch den in Kapitel 5 beschriebenen Ansatz zur Objektverfolgung genutzt. Die dynamischen Zellen könnten zusätzlich in objektlokalen Karten [30] akkumuliert werden, um Objektkonturen über die Zeit genauer und vollständig zu beschreiben. Eine Herausforderung wäre die Akkumulierung der dreidimensionalen Geometrie von Verkehrsteilnehmern in objektlokalen Belegungskarten. Diese hätten zum Vorteil, dass ein bislang parkendes Fahrzeug, das losfährt, sofort mit seiner bereits bekannten Kontur als bewegtes Objekt geführt werden kann und damit der frei werdende Raum sofort bekannt ist.



Literaturverzeichnis

A.1 Verwendete Literatur

- [1] *GSL – GNU Scientific Library*. <http://www.gnu.org/software/gsl/>. Version: 2011
- [2] ABAD, F.; BENDAHAN, R.; WYBO, S.; BOUGNOUX, S.; VESTRI, C.; KAKINAMI, T.: Parking Space Detection. In: *14th World Congress on Intelligent Transport Systems*. Beijing, Oktober 2007
- [3] ARUN, K. S.; HUANG, T. S.; BLOSTEIN, S. D.: Least-Squares Fitting of Two 3-D Point Sets. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-9* (1987), September, Nr. 5, S. 698–700
- [4] BADINO, Hernan; FRANKE, Uwe; PFEIFFER, David: The Stixel World - A Compact Medium Level Representation of the 3D-World. In: *Proceedings of the 31st DAGM Symposium on Pattern Recognition*. Jena, Germany: Springer-Verlag, Berlin, Heidelberg, 2009. – ISBN 978-3-642-03797-9, S. 51–60
- [5] BENTLEY, Jon L.: Multidimensional binary search trees used for associative searching. In: *Communications of the ACM* 18 (1975), September, Nr. 9, S. 509–517. – ISSN 0001-0782. – ACM ID: 361007
- [6] BERGMAN, Niclas: *Recursive Bayesian Estimation Navigation and Tracking Applications*. Linköping, Sweden, Department of Electrical Engineering Linköping University, Diss., 1999

- [7] BESL, Paul J.; MCKAY, Neil D.: A Method for Registration of 3-D Shapes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992), Februar, Nr. 2, S. 239–256
- [8] BLANCO-CLARACO, Jose-Luis: *Contributions to Localization, Mapping and Navigation in Mobile Robotics*. Spanien, Universität Málaga, Diss., November 2009
- [9] BOEHLAU, Christian; LICHTER, Bernd; OTTENHUES, Thomas: New Concept of a Compact LIDAR Scanner for ACC and Safety Applications. In: *SAE World Congress & Exhibition*. Detroit, USA, April 2009
- [10] BOUZOURAA, M. E.; HOFMANN, U.: Fusion of Occupancy Grid Mapping and Model Based Object Tracking for Driver Assistance Systems Using Laser and Radar Sensors. In: *Intelligent Vehicle Symposium*. San Diego, CA, USA, Juni 2010, S. 294–300
- [11] BOUZOURAA, M. E.; HOFMANN, U.: Fusion kartenbasierter Umfeldwahrnehmung mit modellbasierter Objektverfolgung für Fahrerassistenzsysteme. In: *7. Workshop Fahrerassistenzsysteme (FAS2011)*. Walting, Deutschland, März 2011, S. 115–123
- [12] BRAESS, Hans-Hermann; SEIFFERT, Ulrich: *Handbuch Kraftfahrzeugtechnik*. 2. Braunschweig/Wiesbaden: Friedrich Vieweg & Sohn Verlagsgesellschaft mb, 2001. – ISBN 3-528-13114-4
- [13] CASTRO, Renner; LEWINER, Thomas; LOPES, Hélio; TAVARES, Geovan; BORDIGNON, Alex: Statistical optimization of octree searches. In: *Computer Graphics Forum* 27 (2008), September, Nr. 6, S. 1557–1566
- [14] CHOI, Myung G.; JU, Eunjung; CHANG, Jung-Woo; LEE, Jehee; KIM, Young J.: Linkless Octree Using Multi-Level Perfect Hashing. In: *Computer Graphics Forum* 28 (2009), Oktober, Nr. 7, S. 1773–1780
- [15] DAIMLER AG: *Pioniere nachhaltiger Mobilität. Geschäftsbericht 2009*. <http://gb2009.daimler.com/>. Version: 2009
- [16] DAIMLER AG: Automated driving – a new proving method for state-of-the-art safety systems at Mercedes-Benz. In: *Presseinformation* (2010), Mai
- [17] DAIMLER AG: *Die technischen Daten der A-Klasse Limousine*. Oktober 2010
- [18] DAIMLER AG: *Die technischen Daten der E-Klasse Limousine*. 2010
- [19] DAIMLER AG: *Die technischen Daten der S-Klasse Limousine*. 2010
- [20] DAIMLER AG: *Innovation aus Tradition. Geschäftsbericht 2010*. <http://gb2010.daimler.com/>. Version: 2010
- [21] DARMS, Michael; KOMAR, Matthias; LUEKE, Stefan: Map based Road Boundary Estimation. In: *Intelligent Vehicles Symposium (IV 2010)*. San Diego, CA, USA, Juni 2010, S. 609–614
- [22] DEMPSTER, A. P.: Upper and Lower Probabilities Induced by a Multivalued Mapping. In: *Annals of Mathematical Statistics* 38 (1967), Nr. 2, S. 325–339

- [23] DIETMAYER, Klaus: Evidenztheorie: Ein Vergleich zwischen Bayes- und Dempster-Shafer-Methoden. In: *Informationsfusion in der Mess- und Sensortechnik*. Karlsruhe: Universitätsverlag Karlsruhe, 2006, S. 39–49
- [24] DIEWALD, F.; KLAPPSTEIN, J.; SARHOLZ, F.; DICKMANN, J.; DIETMAYER, K.: Radar-interference-based bridge identification for collision avoidance systems. In: *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, Juni 2011. – ISBN 978–1–4577–0890–9, S. 113–118
- [25] DOUCET, Arnaud; GODSILL, Simon; ANDRIEU, Christophe: On sequential Monte Carlo sampling methods for Bayesian filtering. In: *Statistics and Computing* 10 (2000), Nr. 3, S. 197–208
- [26] DOUCET, Arnaud; JOHANSEN, Adam M.: A Tutorial on Particle Filtering and Smoothing: Fifteen years later / Department of Statistics, University of British Columbia. 2008. – Forschungsbericht
- [27] DRAKEN, Wolfgang: Funktion und Nutzung des SAPOS® -Deutschland-Netzes. In: *Flächenmanagement und Bodenordnung (FuB)* (2005), Januar, Nr. 2005/1, S. 21–32
- [28] DRYANOVSKI, I.; MORRIS, W.; XIAO, Jizhong: Multi-volume occupancy grids: An efficient probabilistic 3D mapping model for micro aerial vehicles. In: *Intelligent Robots and Systems (IROS)*. Taipei, China, Oktober 2010, S. 1553–1559
- [29] EBERHARDT, Henning; KLUMPP, Vesa; HANEBECK, Uwe D.: Density Trees for Efficient Nonlinear State Estimation. In: *Proceedings of the 13th International Conference on Information Fusion*. Edinburgh, United Kingdom, Juli 2010, S. 1–8
- [30] EFFERTZ, Jan: *Autonome Fahrzeugführung in urbaner Umgebung durch Kombination objekt- und kartenbasierter Umfeldmodelle*. Braunschweig, Universität Braunschweig, Diss., 2009
- [31] EINHORN, Erik; SCHRÖTER, Christof; GROSS, Horst-Michael: Building 2D and 3D Adaptive-Resolution Occupancy Maps using ND-Trees. In: *Proceedings of the 55th International Scientific Colloquium*. Ilmenau, Germany, 2010, S. 306–311
- [32] EINHORN, Erik; SCHRÖTER, Christof; GROSS, Horst-Michael: Finding the Adequate Resolution for Grid Mapping - Cell Sizes Locally Adapting On-the-Fly. In: *International Conference on Robotics and Automation*. Shanghai, China, Mai 2011, S. 1843–1848
- [33] ELEKTROBIT AUTOMOTIVE GMBH: *Datenblatt EB Assist ADTF*. http://www.elektrobit.com/download/1613/datasheet_eb_assist_adtf_dt_screen. Version: 2009
- [34] ELFES, A.: Using occupancy grids for mobile robot perception and navigation. In: *Computer* 22 (1989), Juni, Nr. 6, S. 46–57. – ISSN 0018–9162
- [35] ELFES, Alberto: A Sonar-Based Mapping and Navigation System. In: *Proceedings of the International Conference on Robotics and Automation*. San Francisco, CA: IEEE, April 1986, S. 1151–1156

- [36] ELIAZAR, Austin: *DP-SLAM*. North Carolina, USA, Duke University, Diss., 2005
- [37] EPANECHNIKOV, V. A.: Non-Parametric Estimation of a Multivariate Probability Density. In: *Theory of Probability and its Applications* 14 (1969), Nr. 1, S. 153–158. – ISSN 0040585X
- [38] ERBS, Friedrich; BARTH, Alexander; FRANKE, Uwe: Moving Vehicle Detection by Optimal Segmentation of the Dynamic Stixel World. In: *Intelligent Vehicles Symposium (IV '11)*. Baden-Baden, Germany, Juni 2011, S. 951–956
- [39] ESTER, Martin; KRIEGEL, Hans-Peter; SANDER, Jörg; XU, Xiaowei: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*. Portland, Oregon: AAAI Press, August 1996, S. 226–231
- [40] EUROTAXGLASS'S INTERNATIONAL AG: EurotaxGlass's asks consumers about relevance of Driver Assistance Systems and take rates across Europe. In: *Presseinformation* (2011), August
- [41] EVANS, J.; CHANG, T.; HONG, T. H.; BOSTELMAN, R. V.; BUNCH, R.: Three Dimensional Data Capture in Indoor Environments for Autonomous Navigation / National Institute of Standards and Technology. 2002 (6912). – Forschungsbericht
- [42] FAIRFIELD, Nathaniel; WETTERGREEN, David; KANTOR, George A.: Real-Time SLAM with Octree Evidence Grids for Exploration in Underwater Tunnels. In: *Journal of Field Robotics* 24 (2007), Februar, Nr. 1-2, S. 3–21
- [43] FOESSEL-BUNTING, Alex: Radar Sensor Model for Three-Dimensional Map Building. In: *Mobile Robots XV and Telemanipulator and Telepresence Technologies VII* Bd. 4195. Boston, MA, USA: SPIE, November 2000
- [44] FOESSEL-BUNTING, Alex; BARES, John; WHITTAKER, William: Three-Dimensional Map Building with MMW-Radar. In: *Proceedings of the 3rd International Conference on Field and Service Robotics*. Otaniemi, Espoo, Finland, Juni 2001
- [45] FOLEY, James D.; DAM, Andries v.; FEINER, Steven K.; HUGHES, John F.; PHILLIPS, Richard L.: *Grundlagen der Computergraphik. Einführung, Konzepte, Methoden*. Addison Wesley Verlag, 1994. – ISBN 3893196471
- [46] FORD WERKE GMBH: Studie belegt: Europaweit Stress beim Einparken - Praktische Hilfe bietet der Einpark-Assistent von Ford. In: *Presseinformation* (2011), Juni
- [47] FRISKEN, Sarah F.; PERRY, Ronald N.: Simple and Efficient Traversal Methods for Quadrees and Octrees. In: *Journal of Graphics, GPU and Game Tools* 7 (2002), Nr. 3, S. 1–11
- [48] FUCHS, Henry; KEDEM, Zvi M.; NAYLOR, Bruce: Predetermining visibility priority in 3-D scenes. In: *ACM SIGGRAPH Computer Graphics* 13 (1979), August, Nr. 2, S. 175–181. – ISSN 0097–8930. – ACM ID: 807441

- [49] FUCHS, Henry; KEDEM, Zvi M.; NAYLOR, Bruce F.: On visible surface generation by a priori tree structures. In: *ACM SIGGRAPH Computer Graphics* 14 (1980), Juli, Nr. 3, S. 124–133. – ISSN 0097–8930. – ACM ID: 807481
- [50] FULGENZI, Chiara; SPALANZANI, Anne; LAUGIER, Christian: Dynamic Obstacle Avoidance in uncertain environment combining PVOs and Occupancy Grid. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Rome, Italy, April 2007, S. 1610–1616
- [51] FÜRSTENBERG, Kay C.: *Fahrzeugumfelderfassung und Fußgängerschutz unter Nutzung mehrzeiliger Laserscanner*, Universität Ulm, Fakultät für Ingenieurwissenschaften und Informatik, Diss., 2009
- [52] GENESYS ELEKTRONIK GMBH: *ADMA-G Produktbeschreibung*. August 2009
- [53] GESCH, Dean; EVANS, Gayla; MAUCK, James; HUTCHINSON, John; CARSWELL JR., William J.: *The National Map—Elevation: U.S. Geological Survey Fact Sheet 2009–3053*. 2009. – <http://ned.usgs.gov/Ned/index.asp>
- [54] GOOGLE INC.: *Google Elevation API*. <http://code.google.com/intl/de-DE/apis/maps/documentation/elevation/>. Version: 2010
- [55] GSTREIN, Gregor; SINZ, Wolfgang; EBERLE, Walter; RICHERT, Julien; BULLINGER, Wilfried: Improvement of Airbag Performance through Pre-Trigging. In: *21st International Technical Conference on the Enhanced Safety of Vehicles*. Stuttgart, Germany, Juni 2009
- [56] GUTMANN, Jens-Steffen: *Robuste Navigation autonomer mobiler Systeme*, Universität Freiburg, Diss., Oktober 2000
- [57] HASTINGS, W. K.: Monte Carlo Sampling Methods Using Markov Chains and Their Applications. In: *Biometrika* 57 (1970), April, Nr. 1, S. 97–109
- [58] HIMMELSBACH, Michael: *Topological FastSLAM*. Humboldt Universität zu Berlin, 2006
- [59] HIMMELSBACH, Michael; HUNDELSHAUSEN, Felix von; WÜNSCHE, Hans-Joachim: Fast Segmentation of 3D Point Clouds for Ground Vehicles. In: *Proceedings of IEEE Intelligent Vehicles Symposium*. San Diego, CA, USA, Juni 2010, S. 560–565
- [60] HIMMELSBACH, Michael; MÜLLER, André; LÜTTEL, Thorsten; WÜNSCHE, Hans-Joachim: LIDAR-based 3D Object Perception. In: *Proceedings of 1st International Workshop on Cognition for Technical Systems*. Munich, Germany, Oktober 2008
- [61] HOL, Jeroen D.; SCHON, Thomas B.; GUSTAFSSON, Fredrik: On Resampling Algorithms for Particle Filters. In: *2006 IEEE Nonlinear Statistical Signal Processing Workshop*. Cambridge, UK: IEEE, September 2006. – ISBN 978–1–4244–0581–7, S. 79–82
- [62] HUBER, Daniel; CARMICHAEL, Owen; HEBERT, Martial: 3D Map Reconstruction from Range Data. In: *Proceedings of the IEEE International Conference on Robotics & Automation*. San Francisco, CA, April 2000, S. 891–897

- [63] IBEO AUTOMOBILE SENSOR GMBH: *Betriebsanleitung ibeo LUX Laserscanner*. Februar 2009
- [64] ISARD, Michael; BLAKE, Andrew: CONDENSATION—Conditional Density Propagation for Visual Tracking. In: *International Journal of Computer Vision* 29 (1998), Nr. 1, S. 5–28
- [65] JULIER, Simon J.; UHLMANN, Jeffrey K.: A New Extension of the Kalman Filter to Nonlinear Systems. In: *SPIE AeroSense Symposium*. Orlando, FL, April 1997, S. 182–193
- [66] JUNG, Boyoon; SUKHATME, Gaurav S.: Real-time Motion Tracking from a Mobile Robot. In: *International Journal of Social Robotics* 2 (2009), Dezember, Nr. 1, S. 63–78. – ISSN 1875–4791
- [67] JUNG, Ho G.; CHO, Young H.; YOON, Pal J.; KIM, Jaihie: Scanning Laser Radar-Based Target Position Designation for Parking Aid System. In: *IEEE Transactions on Intelligent Transportation Systems* Bd. 9, 2008 (3), S. 406–424
- [68] KALMAN, R. E.: A New Approach to Linear Filtering and Prediction Problems. In: *Transactions of the ASME—Journal of Basic Engineering* 82 (Series D) (1960), S. 35–45
- [69] KAMMEL, Soeren; PITZER, Benjamin; VACEK, Stefan; SCHROEDER, Joachim; FRESE, Christian; WERLING, Moritz; GOEBL, Matthias: DARPA Urban Challenge Team AnnieWAY Technical System Description / DARPA Urban Challenge Finalist Technical Paper. 2007. – Forschungsbericht
- [70] KAPP, Andreas; GRÖLL, Lutz: Motion-Scan Effect. In: *at - Automatisierungstechnik* 54 (2006), November, Nr. 11, S. 546–555
- [71] KLEINER, Alexander D.: *Mapping and exploration for search and rescue with humans and mobile robots*. Freiburg, Universität Freiburg, Diss., 2007
- [72] KLUGE, Thomas: Laserscanner for Automotive Applications. In: *1st International Workshop on Intelligent Transportation (WIT 2004)*. Hamburg, Deutschland, März 2004
- [73] KONOLIGE, Kurt: Improved Occupancy Grids for Map Building. In: *Autonomous Robots* 4 (1997), April, Nr. 4, S. 351–367
- [74] KONRAD, Marcus; DIETMAYER, K.: Occupancy Grid Mapping using the Dempster-Shafer-Theory. In: *8th International Workshop on Intelligent Transportation (WIT 2011)*. Hamburg, Deutschland, März 2011, S. 167–172
- [75] KONRAD, Marcus; DIETMAYER, Klaus; SZCZOT, Magdalena: Road Course Estimation Using Local Maps and Digital Maps. In: *Advanced Microsystems for Automotive Applications 2010*. Springer Berlin Heidelberg, 2010 (VDI-Buch), S. 433–442
- [76] KONRAD, Marcus; FUCHS, M.; LÖHLEIN, Otto; DIETMAYER, K.: Detektion und Tracking dynamischer Objekte in Occupancy Grids. In: *7. Workshop Fahrerassistenzsysteme (FAS2011)*. Braunschweig, März 2011, S. 105–114

- [77] KONRAD, Marcus; SZCZOT, Magdalena; DIETMAYER, Klaus; SCHÜLE, Florian: Generic Grid Mapping for Road Course Estimation. In: *Intelligent Vehicle Symposium (IV'11)*. Baden-Baden, Germany: IEEE, Juni 2011, S. 845–850
- [78] KÖHLER, Michael: Accurate PreCrash Detection. In: *11th World Congress on Intelligent Transportation Systems*. Nagoya, Japan, Oktober 2004
- [79] LEVINSON, Jesse; THRUN, Sebastian: Robust Vehicle Localization in Urban Environments Using Probabilistic Maps. In: *International Conference on Robotics and Automation*. Anchorage, AK, USA, Mai 2010, S. 4372–4378
- [80] LINZMEIER, Dirk: *Real-Time Detection of Pedestrians from a Moving Vehicle Using Thermopile and Radar Sensor Fusion*. Ulm, Universität Ulm, Diss., 2006
- [81] LONGBIN, Mo; XIAOQUAN, Song; YIYU, Zhou; KANG, Sun Z.; BAR-SHALOM, Y.: Unbiased converted measurements for tracking. In: *IEEE Transactions on Aerospace and Electronic Systems* 34 (1998), Juli, Nr. 3, S. 1023–1027. – ISSN 0018–9251
- [82] MACCORMICK, J.; BLAKE, A.: A probabilistic exclusion principle for tracking multiple objects. In: *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999* Bd. 1. Kerkyra, Greece: IEEE, September 1999. – ISBN 0–7695–0164–8, S. 572–578
- [83] MEAGHER, Donald: Geometric modeling using octree encoding. In: *Computer Graphics and Image Processing* 19 (1982), Juni, Nr. 2, S. 129–147. – ISSN 0146–664X
- [84] MONTEMERLO, Michael: *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem With Unknown Data Association*. Pittsburgh, PA, Carnegie Mellon University, Diss., Juni 2003
- [85] MONTEMERLO, Michael; THRUN, Sebastian: A multi-resolution pyramid for outdoor robot terrain perception. In: *Proceedings of the 19th National Conference on Artificial Intelligence*. San Jose, CA: AAAI Press, Juli 2004 (AAAI'04). – ISBN 0–262–51183–5, S. 464–469. – ACM ID: 1597224
- [86] MOOSMANN, Frank; STILLER, Christoph: Velodyne SLAM. In: *Intelligent Vehicle Symposium (IV'11)*. Baden-Baden, Germany: IEEE, Juni 2011, S. 393–398
- [87] MORAS, Julien; CHERFAOUI, Veronique; BONNIFAIT, Philippe: Moving Objects Detection by Conflict Analysis in Evidential Grids. In: *Intelligent Vehicles Symposium (IV 2011)*. Baden-Baden, Germany: IEEE, Juni 2011, S. 1120–1125
- [88] MORAVEC, Hans P.: Robot Spatial Perception by Stereoscopic Vision and 3D Evidence Grids / The Robotics Institute, Carnegie Mellon University. Pittsburgh, Pennsylvania, September 1996 (CMU-RI-TR-96-34). – Forschungsbericht
- [89] MULLANE, John; ADAMS, Martin; WIJESOMA, Wijerupage S.: Evidential versus Bayesian Estimation for Radar Map Building. In: *9th International Conference on Control, Automation, Robotics and Vision (ICARCV '06)*. Singapore, Dezember 2006, S. 1–8

- [90] MÄHLISCH, M.; SCHWEIGER, R.; RITTER, W.; DIETMAYER, K.: Sensorfusion Using Spatio-Temporal Aligned Video and Lidar for Improved Vehicle Detection. In: *Intelligent Vehicles Symposium (IV 2006)*. Tokyo, Japan: IEEE, Juni 2006, S. 424–429
- [91] MÄHLISCH, Mirko: *Filtersynthese zur simultanen Minimierung von Existenz-, Assoziations- und Zustandsunsicherheiten in der Fahrzeugumfelderfassung mit heterogenen Sensordaten*. Ulm, Universität Ulm, Diss., August 2009
- [92] NELDER, John A.; MEAD, Roger: A Simplex Method for Function Minimization. In: *Computer Journal* 7 (1965), Nr. 4, S. 308–313
- [93] NÜCHTER, Andreas; LINGEMANN, Kai; HERTZBERG, Joachim; SURMANN, Hartmut: 6D SLAM - 3D mapping outdoor environments: Research Articles. In: *Journal of Field Robotics* 24 (2007), Nr. 8-9, S. 699–722
- [94] NÜCHTER, Andreas; SURMANN, Hartmut; LINGEMANN, Kai; HERTZBERG, Joachim; THRUN, S.: 6D SLAM with an Application in Autonomous Mine Mapping. In: *In Proceedings of the IEEE International Conference on Robotics and Automation*. New Orleans, USA, April 2004, S. 1998—2003
- [95] PATHAK, Kaustubh; BIRK, Andreas; POPPINGA, Jann; SCHWERTFEGER, Sören: 3D Forward Sensor Modeling and Application to Occupancy Grid based Sensor Fusion. In: *International Conference on Intelligent Robots and Systems (IROS)*. San Diego, CA, USA, November 2007, S. 2059–2064
- [96] PAYEUR, P.; HEBERT, P.; LAURENDEAU, D.; GOSSELIN, C. M.: Probabilistic Octree Modeling of a 3-D Dynamic Environment. In: *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*. Albuquerque, New Mexico, April 1997, S. 1289–1296
- [97] PETROVSKAYA, Anna; THRUN, Sebastian: Model based vehicle detection and tracking for autonomous urban driving. In: *Autonomous Robots* 26 (2009), April, Nr. 2-3, S. 123–139. – ISSN 0929–5593
- [98] PFEIFFER, David; FRANKE, Uwe: Towards a Global Optimal Multi-Layer Stixel Representation of Dense 3D Data. In: *British Machine Vision Conference 2011*. Dundee, Scotland, August 2011
- [99] PFEIFFER, David; MORALEZ, Sandino; BARTH, Alexander; FRANKE, Uwe: Ground Truth Evaluation of the Stixel Representation Using Laser Scanners. In: *13th International IEEE Conference on Intelligent Transportation Systems*. Madeira, Portugal, September 2010, S. 1091–1097
- [100] PIETZSCH, Sylvia; AYCARD, Olivier; BURLET, Julien; VU, Trung D.; HACKBARTH, Thomas; APPENRODT, Nils; DICKMANN, Jürgen; RADIG, Bernd: Results of a Precrash Application based on Laserscanner and Short Range Radars. In: *Intelligent Vehicles Symposium (IV 2008)*. Eindhoven, Niederlande, Juni 2008, S. 367–372
- [101] RESS, Christian; BALZER, Dirk; BRACHT, Alexander; DUREKOVIC, Sinisa; LÖWENAU, Jan: Adasis Protocol For Advanced In-Vehicle Applications. In: *ITS World Congress*.

- New York, NY, USA, November 2008
- [102] RICHTER, Eric; LINDNER, Philipp; WANIELIK, Gerd; TAKAGI, Kiyokazu; ISOGAI, Akira: Advanced Occupancy Grid Techniques for Lidar based Object Detection and Tracking. In: *Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems*. St. Louis, MO, USA, Oktober 2009, S. 1–5
 - [103] SALMOND, D. J.; BIRCH, H.: A particle filter for track-before-detect. In: *Proceedings of the 2001 American Control Conference* Bd. 5. Arlington, VA, Juni 2001, S. 3755–3760
 - [104] SCHEUNERT, U.; FARDI, B.; MATTERN, N.; WANIELIK, G.; KEPPELER, N.: Free space determination for parking slots using a 3D PMD sensor. In: *2007 IEEE Intelligent Vehicles Symposium*. Istanbul, Turkey: IEEE, Juni 2007. – ISBN 1–4244–1067–3, S. 154–159
 - [105] SCHEUNERT, Ullrich; MATTERN, Norman; LINDNER, Philipp; WANIELIK, Gerd: Generalized Grid Framework for Multi Sensor Data Fusion. In: *11th International Conference on Information Fusion*. Köln, Deutschland, Juli 2008, S. 1–7
 - [106] SCHMID, Matthias R.: *Merkmalsextraktion aus Laserscannerdaten im automobilen Umfeld*. Universität Ulm, September 2008
 - [107] SCHULZ, Dirk; BURGARD, Wolfram; FOX, Dieter; CREMERS, Armin B.: Tracking Multiple Moving Targets with a Mobile Robot using Particle Filters and Statistical Data Association. In: *IEEE International Conference on Robotics and Automation (ICRA)* Bd. 2. Seoul, Korea, Mai 2001, S. 1665–1670
 - [108] SCHWARZ, Daniel; KLOEDEN, Horst; RASSHOFER, Ralph: Ko-TAG - Cooperative Sensor Technology for Traffic Safety Applications. In: *Workshop on Intelligent Transportation (WIT2011)*. Hamburg, Germany, März 2011, S. 31–34
 - [109] SCHÜLE, Florian; SCHWEIGER, Roland; LÖHLEIN, Otto; DIETMAYER, Klaus: Vehicle Positioning on a Digital Map for Road Course Prediction. In: *First International Symposium on Future Active Safety Technology toward zero-traffic-accident*. Tokyo, Japan, September 2011
 - [110] SHAFER, Glenn: *A Mathematical Theory of Evidence*. Princeton University Press, 1976
 - [111] SHNIDMAN, D.A.: Radar detection probabilities and their calculation. In: *IEEE Transactions on Aerospace and Electronic Systems* 31 (1995), Juli, Nr. 3, S. 928–950. – ISSN 00189251
 - [112] SKUTEK, Michael: *Ein PreCrash-System auf Basis multisensorieller Umgebungserfassung*, TU Chemnitz, Diss., Januar 2006
 - [113] SMART MICROWAVE SENSORS GMBH, BRAUNSCHWEIG: *UMRR - LCA BSD Technical Information Sheet*. März 2011
 - [114] SMITH, R.; SELF, M.; CHEESEMAN, P.: Estimating uncertain spatial relationships in robotics. In: *Autonomous robot vehicles*. Springer-Verlag New York, Inc., 1990. – ISBN 0–387–97240–4, S. 167–193. – ACM ID: 93291

- [115] STACHNISS, Cyrill; FRESE, Udo; GRISETTI, Giorgio: *OpenSLAM.org*. 2010. – <http://openslam.org/>
- [116] STACHNISS, Cyrill; HÄHNEL, Dirk; BURGARD, Wolfram: Exploration with Active Loop-Closing for FastSLAM. In: *Intelligent Robots and Systems (IROS 2004)* Bd. 2. Sendai, Japan, Oktober 2004, S. 1505–1520
- [117] STATISTISCHES BUNDESAMT: *Im Blickpunkt Verkehr in Deutschland 2006*. September 2006
- [118] STATISTISCHES BUNDESAMT: *Verkehrsunfälle 2010*. 2010
- [119] STRELLER, Daniel; DIETMAYER, Klaus: Lidar-Sensoren zur Umfelderkennung bei Straßenfahrzeugen. In: *Tagungsband XVI. Messtechnisches Symposium in Kassel*. Kassel, Deutschland: ShakerVerlag, Oktober 2002, S. 163–176
- [120] SUHR, Jae K.; JUNG, Ho G.; BAE, Kwanghyuk; KIM, Jaihie: Automatic free parking space detection by using motion stereo-based 3D reconstruction. In: *Machine Vision and Applications* 21 (2010), Februar, Nr. 2, S. 163–176
- [121] SURMANN, H.; LINGEMANN, K.; NUCHTER, A.; HERTZBERG, J.: Fast Acquiring and Analysis of Three Dimensional Laser Range Data. In: *Proceedings of the Sixth International Fall Workshop Vision, Modeling, and Visualization*. Stuttgart, Germany, November 2001, S. 59–66
- [122] SZCZOT, Magdalena; SERFLING, Matthias; LÖHLEIN, Otto; SCHÜLE, Florian; KONRAD, Marcus; DIETMAYER, Klaus: Global Positioning Using a Digital Map And an Imaging Radar Sensor. In: *Intelligent Vehicle Symposium (IV'10)*. San Diego, CA, USA, Juni 2010, S. 406–411
- [123] TEIZER, J.; BOSCHE, F.; CALDAS, C. H.; HAAS, C. T.: Real-time spatial detection and tracking of resources in a construction environment. In: *Proceedings of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering*. Montreal, Canada, Juni 2006, S. 494–502
- [124] THIEM, J.; MÜHLENBERG, M.: Datafusion of Two Driver Assistance System Sensors. In: *Advanced Microsystems for Automotive Applications 2005*. Berlin, Germany: Springer Berlin Heidelberg, 2005 (VDI-Buch), S. 97–114
- [125] THRUN, S.; MONTEMERLO, M.; DAHLKAMP, H.; STAVENS, D.; ARON, A.; DIEBEL, J.; FONG, P.: Stanley: The Robot that Won the DARPA Grand Challenge. In: *Journal of Field Robotics* 23 (2006), September, Nr. 9, S. 661–692
- [126] THRUN, S.; MONTEMERLO, M.; KOLLER, D.; WEGBREIT, B.; NIETO, J.; NEBOT, E.: FastSLAM: An Efficient Solution to the Simultaneous Localization And Mapping Problem with Unknown Data Association. In: *Journal of Machine Learning Research* 4 (2004), Nr. June, S. 1–48
- [127] THRUN, Sebastian: Robot Mapping: A Survey. 2002 (CMU-CS-02-111). – Forschungsbericht

- [128] THRUN, Sebastian: Learning Occupancy Grids With Forward Sensor Models. In: *Autonomous Robots* 15 (2003), September, Nr. 2, S. 111–127
- [129] THRUN, Sebastian; BURGARD, Wolfram; FOX, Dieter: A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping. In: *IEEE International Conference on Robotics and Automation* Bd. 1. San Francisco, CA, April 2000, S. 321–328
- [130] THRUN, Sebastian; BURGARD, Wolfram; FOX, Dieter: *Probabilistic Robotics*. The MIT Press, 2005
- [131] VELODYNE ACOUSTICS, INC: Velodyne’s HDL-64E: A High Definition Lidar Sensor for 3-D Applications. In: *White Paper* (2007), Oktober. www.velodynehdl.com
- [132] VON HUNDELSHAUSEN, Felix; HIMMELSBACH, Michael; HECKER, Falk; MÜLLER, André; WÜNSCHE, Hans-Joachim: Driving with tentacles: Integral structures for sensing and motion. In: *Journal of Field Robotics* 25 (2008), August, Nr. 9, S. 640–673. – ISSN 1556–4959
- [133] VU, Trung-Dung: *Vehicle Perception: Localization, Mapping with Detection, Classification and Tracking of Moving Objects*. Frankreich, Institut National Polytechnique de Grenoble, Diss., September 2009
- [134] VU, Trung-Dung; AYCARD, Olivier: Laser-based Detection and Tracking Moving Objects using Data-Driven Markov Chain Monte Carlo. In: *2009 IEEE International Conference on Robotics and Automation*. Kobe, Japan, Mai 2009, S. 3800–3806
- [135] VU, Trung-Dung; AYCARD, Olivier; APPENRODT, Nils: Online Localization and Mapping with Moving Object Tracking in Dynamic Outdoor Environments. In: *Proceedings of the 2007 IEEE Intelligent Vehicles Symposium, Istanbul, Turkey*. Istanbul, Turkey: IEEE, Juni 2007, S. 190–195
- [136] WANG, Chieh-Chih: *Simultaneous Localization, Mapping and Moving Object Tracking*, Carnegie Mellon University, Diss., April 2004
- [137] WANG, Miao; TSENG, Yi-Hsing: Automatic 3D Feature Extraction from Structuralized LIDAR Data. In: *26th Asian Association on Remote Sensing (ACRS2005)*. Hanoi, Vietnam, 2005
- [138] WEISS, T.; KAEMPCHEN, N.; DIETMAYER, K.: Precise ego-localization in urban areas using Laserscanner and high accuracy feature maps. In: *Intelligent Vehicles Symposium (IV2005)*. Las Vegas, USA: IEEE, Juni 2005, S. 284–289
- [139] WEISS, T.; SCHIELE, B.; DIETMAYER, K.: Robust Driving Path Detection in Urban and Highway Scenarios Using a Laser Scanner and Online Occupancy Grids. In: *IEEE Intelligent Vehicles Symposium*. Istanbul, Turkey: IEEE, Juni 2007, S. 184–189
- [140] WEISS, Thorsten: *Hochgenaue Positionierung und Kartographie mit Laserscannern für Fahrerassistenzsysteme*. Ulm, Universität Ulm, Diss., 2011

- [141] WEISS, Thorsten; DIETMAYER, Klaus: Applications for Driver Assistant Systems Using Online Maps. In: *Proceedings of 5th International Workshop on Intelligent Transportation (WIT 2008)*. Hamburg, Deutschland, März 2008
- [142] WELCH, Greg; BISHOP, Gary: An Introduction to the Kalman Filter / University of North Carolina at Chapel Hill. Chapel Hill, NC 27599-3175, Juli 2006 (95-041). – Forschungsbericht
- [143] WENDER, Stefan: *Multisensorsystem zur erweiterten Fahrzeugumfelderfassung*, Universität Ulm, Fakultät für Ingenieurwissenschaften und Informatik, Diss., Oktober 2008
- [144] WENGER, Josef: Short range radar – being on the market. In: *Proceedings of the 37th European Microwave Conference*. Munich, Germany, Oktober 2007, S. 1534—1537
- [145] WINNER, Hermann (Hrsg.); HAKULI, Stephan (Hrsg.); WOLF, Gabriele (Hrsg.): *Handbuch Fahrerassistenzsysteme*. Vieweg+Teubner, 2009 (Kraftfahrzeugtechnik)
- [146] WURM, Kai M.; HORNUNG, Armin; BENNEWITZ, Maren; STACHNISS, Cyrill; BURGARD, Wolfram: OctoMap: A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems. In: *Proceedings of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*. Anchorage, AK, USA, Mai 2010

A.2 Eigene Publikationen und Patente

- [147] MUNTZINGER, Marc M.; AEBERHARD, Michael; ZUTHER, Sebastian; MÄHLISCH, Mirko; SCHMID, Matthias R.; DIETMAYER, Klaus: Reliable Automotive Pre-Crash System with Out-of-Sequence Measurement Processing. In: *Intelligent Vehicle Symposium (IV'10)*. San Diego, CA, USA: IEEE, Juni 2010, S. 1022–1027
- [148] MÄHLISCH, M.; SCHMID, Matthias R.: *Verfahren zur Umgebungsrepräsentation eines Fahrzeugs*. Patent DE102010011629A1, Deutschland, Januar 2011
- [149] MÄHLISCH, Mirko; SCHMID, Matthias R.: *Verfahren und Vorrichtung zur Erfassung einer Umgebung eines Fahrzeugs*. Patent DE102009033852A1, Deutschland, Mai 2010
- [150] SCHMID, Matthias R.: *Verfahren zur Ermittlung einer Position eines Fahrzeugs*. Patent DE102010055371A1, Deutschland, August 2011
- [151] SCHMID, Matthias R.; DICKMANN, Jürgen: *Verfahren und Vorrichtung zur Umgebungserfassung eines Fahrzeugs*. Patent DE102011103743A1, Deutschland, März 2012
- [152] SCHMID, Matthias R.; KLAPPSTEIN, Jens: *Verfahren zur Umgebungsrepräsentation eines Fahrzeugs*. Patent DE102011113016A1, Deutschland, März 2012
- [153] SCHMID, Matthias R.; MÄHLISCH, Mirko; DICKMANN, Jürgen; WÜNSCHE, Hans-Joachim: Dynamic Level of Detail 3D Occupancy Grids for Automotive Use. In: *Intelligent Vehicle Symposium (IV'10)*. San Diego, CA, USA: IEEE, Juni 2010, S. 269–274
- [154] SCHMID, Matthias R.; MÄHLISCH, Mirko; DICKMANN, Jürgen; WÜNSCHE, Hans-Joachim: Straight-Feature-Based Self-Localization For Urban Scenarios. In: *8th International Workshop on Intelligent Transportation (WIT 2011)*. Hamburg, Germany, März 2011, S. 63–68
- [155] SCHMID, Matthias R.; ATEŞ, Savas; DICKMANN, Jürgen; VON HUNDELSHAUSEN, Felix; WÜNSCHE, Hans-Joachim: Parking Space Detection with Hierarchic Dynamic Occupancy Grids. In: *Intelligent Vehicle Symposium (IV'11)*. Baden-Baden, Germany: IEEE, Juni 2011, S. 254–259

A.3 Betreute Abschlussarbeiten

- [156] ATEŞ, Savas: *Detektion und Vermessung von Parklücken in dreidimensionalen Belegungskarten*. Universität Ulm, Diplomarbeit, Oktober 2011
- [157] KUZIKOV, Dmitriy: *Merkmalsbasierte Selbstlokalisierung in 2D und 3D für Fahrerassistenzsysteme*. Hochschule Karlsruhe, Masterarbeit, August 2011
- [158] WIYOGO, Yongkie: *Grid-Based Detection and Tracking of Moving Objects in Dynamic Environments*. Technische Universität Darmstadt, Masterarbeit, September 2011