

Karlsruher Schriften
zur Anthropomatik

Band 49



Thomas Emter

**Integrierte Multi-Sensor-Fusion für die
simultane Lokalisierung und Kartenerstellung
für mobile Robotersysteme**



Scientific
Publishing

Thomas Emter

**Integrierte Multi-Sensor-Fusion für die
simultane Lokalisierung und Kartenerstellung
für mobile Robotersysteme**

Karlsruher Schriften zur Anthropomatik

Band 49

Herausgeber: Prof. Dr.-Ing. habil. Jürgen Beyerer

Eine Übersicht aller bisher in dieser Schriftenreihe
erschienenen Bände finden Sie am Ende des Buchs.

Integrierte Multi-Sensor-Fusion für die simultane Lokalisierung und Kartenerstellung für mobile Robotersysteme

von
Thomas Emter

Karlsruher Institut für Technologie
Institut für Anthropomatik und Robotik

Integrierte Multi-Sensor-Fusion für die simultane Lokalisierung
und Kartenerstellung für mobile Robotersysteme

Zur Erlangung des akademischen Grades eines Doktors der
Ingenieurwissenschaften von der KIT-Fakultät für Informatik des
Karlsruher Instituts für Technologie (KIT) genehmigte Dissertation

von Thomas Emter

Tag der mündlichen Prüfung: 23. Juli 2020
Referent: Prof. Dr.-Ing. habil. Jürgen Beyerer
Korreferent: Prof. Dr.-Ing. Michael Heizmann

Impressum



Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark
of Karlsruhe Institute of Technology.
Reprint using the book cover is not allowed.

www.ksp.kit.edu



*This document – excluding the cover, pictures and graphs – is licensed
under a Creative Commons Attribution-Share Alike 4.0 International License
(CC BY-SA 4.0): <https://creativecommons.org/licenses/by-sa/4.0/deed.en>*



*The cover page is licensed under a Creative Commons
Attribution-No Derivatives 4.0 International License (CC BY-ND 4.0):
<https://creativecommons.org/licenses/by-nd/4.0/deed.en>*

Print on Demand 2021 – Gedruckt auf FSC-zertifiziertem Papier

ISSN 1863-6489
ISBN 978-3-7315-1074-1
DOI 10.5445/KSP/1000128137

Kurzfassung

Für die autonome Navigation mobiler Robotersysteme sind gewisse Fähigkeiten notwendig. Diese können in die Grundfunktionalitäten Lokalisierung, Kartierung und Bewegungsplanung unterteilt werden. Um komplexere Aufgabenstellungen zu ermöglichen, können diese Funktionalitäten miteinander kombiniert werden. Die simultane Lokalisierung und Kartenerstellung (engl. *simultaneous localization and mapping* – SLAM) gilt dabei als eine besondere Herausforderung, da die aus den fehlerbehafteten Sensordaten resultierenden Unsicherheiten in der Lokalisierung und der erzeugten Karte voneinander abhängig sind.

Durch die Fusion mehrerer Sensoren kann eine Reduktion der Unsicherheiten erzielt werden und folglich eine genauere Lokalisierung erreicht werden. Mit SLAM-Algorithmen wird durch entsprechende Berücksichtigung der Abhängigkeiten zwischen der Lokalisierung und der bisher erstellten Karte eine konsistente und präzise Kartierung ermöglicht.

In der vorliegenden Arbeit werden Methoden für die Kombination und Integration mehrerer Sensoren für die robuste und präzise Lokalisierung und Kartenerstellung in heterogenen Außenumgebungen vorgestellt, welche sowohl in urbanen als auch in unstrukturierten Bereichen einen möglichst robusten Einsatz erlauben sollen.

Es wird die Fragestellung behandelt, wie zusätzliche Sensoren für SLAM eingesetzt werden können, um die Lokalisierung und Kartenerstellung robuster zu machen und die Präzision zu erhöhen. Eine robustere

Kartierung wird sowohl durch eine zuverlässigere Wiedererkennung der Landmarken und damit bereits besuchter Orte erfolgen als auch durch die Kombination von dichten und landmarkenbasierten Kartierungsverfahren. Weiterhin ist Teil der Untersuchungen, wie sich durch die Kombination von SLAM mit Multi-Sensor-Fusion zur Integration zusätzlicher absoluter Sensoren ein zuverlässigeres Schließen von Schleifen erzielen lässt. Außerdem wird die Integration weiterer relativer Sensoren, d. h. Sensoren, deren Messungen sich auf die Differenz zwischen zwei Zuständen beziehen, betrachtet. Diese können nicht ohne Weiteres mit einfachen Filterverfahren integriert werden, weshalb in der Arbeit erweiterte Filterverfahren behandelt werden, um die probabilistisch korrekte Integrierbarkeit von zusätzlichen relativ messenden Sensoren zu ermöglichen.

Für die zuverlässigere Wiedererkennung von Landmarken wurde ein erweitertes Landmarkenmodell mit zusätzlichen ortsunabhängigen und visuellen Merkmalen entwickelt. Durch eine probabilistische Modellierung kann das erweiterte Modell neben der Wiedererkennung auch in dem zugrundeliegenden Partikelfilter-SLAM-Algorithmus in die Berechnung der Partikelgewichte eingehen und damit die Robustheit weiter verbessern.

Ebenfalls auf Basis des Partikelfilters wurde ein modularer SLAM-Algorithmus entwickelt, welcher über ein erweitertes Kalman-Filter pro Partikel die Möglichkeit bietet, konsistent weitere Sensoren oder Lokalisierungs- bzw. Kartierungsmethoden zu integrieren. Damit konnte zum einen ein hybrider Ansatz realisiert werden, bei dem gleichzeitig eine dichte 2D-Rasterkarte und eine Landmarkenkarte aufgebaut werden, um deren gegenseitige Vorteile für eine robustere Kartierung und Lokalisierung in gemischten Umgebungen in ergänzender Weise zu nutzen. Zum anderen erlaubt der modulare SLAM-Algorithmus die Integration absoluter Sensoren, welche den Vorteil mit sich bringen, dass die Unsicherheit global begrenzt werden kann, was das Schließen von Schleifen vereinfacht und letztendlich das Erstellen konsistenter Karten unterstützt.

Für die Integration zusätzlicher relativer Sensoren wurden Methoden mit stochastischem Klonen untersucht und eine neuartige Methode des impliziten Klonens in Kombination mit Smoothing entwickelt. Beim stochastischen Klonen wird die Zustandsschätzung mit einem Klon der Zustandsschätzung, auf die sich eine relative Messung bezieht, und die Systemkovarianzmatrix erweitert. Beim impliziten Klonen kann die augmentierte Systemkovarianzmatrix bei Bedarf rekonstruiert werden und muss nicht durch alle Filterschritte propagiert werden. Durch das Smoothing werden Informationen von zwischenzeitlich erfassten Messungen anderer Sensoren auf den Klon zurückpropagiert, was die robuste Integration mehrerer relativer und absoluter Messungen ermöglicht.

Für die Fusion der Sensordaten zum möglichst genauen Zeitpunkt wurde ein Zeitstempelschätzer entwickelt, um bei der Sensoranbindung und -datenverarbeitung mit Rechnern ohne speziellem Echtzeitbetriebssystem die schwankenden Latenzen zu kompensieren. Des Weiteren wurde ein paralleles Verarbeitungsverfahren entwickelt, welches es erlaubt, die Lokalisierungsschätzung mit einer möglichst konstant hohen Rate auszugeben und dennoch die Information von Sensoren zu integrieren, deren Verarbeitung im Vergleich zu den anderen Sensoren länger dauert und damit bei sequentieller Verarbeitung zu Verzögerungen führen würde.

In einer vergleichenden Evaluation konnte gezeigt werden, dass ein erweitertes Landmarkenmodell mit ortsunabhängigen Merkmalen zu einer zuverlässigeren Datenzuordnung beitragen kann. Durch die verbesserte Datenzuordnung wurden bei dem SLAM-Algorithmus weniger Partikel für das erfolgreiche Schließen der Schleifen benötigt. Weiterhin konnte gezeigt werden, dass bei der Verwendung von mehreren Partikeln auch die Genauigkeit der Kartierung durch eine verbesserte Datenzuordnung gesteigert werden kann.

Mit dem modularen SLAM-Algorithmus konnte mit einer hybriden Konfiguration von 2D-Rasterkarte und Landmarkenkarte gegenüber den Einzelverfahren eine verbesserte Kartierung und damit auch Lokalisierung erzielt werden. Auch durch die Hinzunahme eines absoluten Sensors

konnte die Robustheit des modularen SLAM-Algorithmus hinsichtlich des Schließens großer Schleifen effektiv gesteigert werden.

Die Methode des impliziten Klonens mit Smoothing hat bei der Integration zusätzlicher relativer Sensoren in Evaluationen mit einem simulierten GNSS-Ausfall und im Vergleich zu einer Referenzlokalisierung nahezu gleiche Ergebnisse erzielt wie alternative Methoden mit stochastischem Klonen bzw. inkrementellen Faktor-Graph-Algorithmien. Bei den Betrachtungen zum Rechenaufwand hat sich gezeigt, dass die Filter-Methoden mit einer sehr geringen Auslastung der verfügbaren Rechenzeit für reale Anwendungen geeignet sind und mit dem parallelen Verarbeitungsverfahren auch rechenaufwändigere Verfahren wie Scan-Matching integrieren können.

Ein weiterer Aspekt der Arbeit war die möglichst praxisnahe Umsetzung der entwickelten Methoden und deren Einsatz auf realen Robotersystemen. So konnten mit der Implementierung der vorgestellten Algorithmen sowie der Umsetzung der Zeitstempelfilterung und der parallelen Verarbeitung umfassende Verbesserungen bei der Lokalisierung und Kartierung für mobile Robotersysteme geschaffen werden, welche von der Akquise der Sensordaten über die Fusion mehrerer relativer und absoluter Sensoren bis zur Integration in SLAM-Algorithmien reichen. Ein Großteil der vorgestellten Algorithmen ist auf verschiedenen Forschungsplattformen mit unterschiedlichen Sensorausstattungen erfolgreich im Einsatz.

Abstract

Mobile robots require certain skills for autonomous navigation. These can be divided into the basic functionalities localization, mapping and motion planning. To enable more complex tasks, these functionalities can be combined with each other. Simultaneous localization and mapping (SLAM) is considered a particular challenge because the uncertainties in the localization and the generated map resulting from the noisy sensor data are interdependent.

By merging several sensors, a reduction of the uncertainties and consequently a more precise localization can be achieved. With SLAM algorithms, consistent and precise mapping is made possible by taking into account the dependencies between the localization and the map created so far.

In this thesis, methods for the combination and integration of several sensors for robust and precise localization and mapping in heterogeneous outdoor environments are presented. They are designed to allow a robust application in urban as well as in unstructured areas.

The question of how additional sensors can be used for SLAM to further increase robustness and accuracy of the localization and mapping is addressed. A more robust mapping will be achieved by a more reliable recognition of landmarks, and thus already visited places, as well as by a combination of dense and landmark-based mapping methods. Furthermore, increasing the reliability of loop closure by combining SLAM with

multi-sensor fusion to integrate additional absolute sensors is part of the investigations. Additionally, the integration of multiple relative sensors, i.e., sensors whose measurements refer to the difference between two states, is considered. These relative measurements cannot be readily integrated with standard filter methods. Therefore, this thesis covers advanced filter methods to enable the probabilistically correct integration of additional relative measuring sensors.

For a more reliable recognition of landmarks, an extended landmark model with additional visual and location-independent features was developed. Through probabilistic modelling, the extended model can be used not only for recognition but also to calculate particle weights in the underlying particle filter SLAM algorithm, thus further improving robustness.

A modular SLAM algorithm was additionally developed on the basis of the particle filter, which offers the possibility of consistently integrating further sensors, localization methods, or mapping methods via an extended Kalman filter per particle. On the one hand, a hybrid approach was realized, where a dense 2D grid map and a landmark map are built concurrently, in order to exploit their mutual advantages for a more robust mapping and localization in mixed environments in a complementary way. On the other hand, the modular SLAM algorithm allows the integration of absolute sensors, which has the advantage that the uncertainty can be bounded globally, simplifying the closing of loops and ultimately supporting the building of consistent maps.

For the integration of additional relative sensors, methods with stochastic cloning were investigated and a novel method of implicit cloning combined with smoothing was developed. In stochastic cloning, the state is augmented with a clone of the state to which a relative measurement refers, and the system covariance matrix is also augmented. In implicit cloning, the augmented system covariance matrix can be reconstructed on demand and does not need to be propagated through all filter steps. By smoothing, information from intermediate measurements of other sensors is propa-

gated back to the clone, which enables the robust integration of several relative and absolute measurements.

A time stamp estimator was developed for fusion of sensor data at as exact a time as possible in order to compensate for the fluctuating latencies caused by sensor data acquisition and data processing with computers without a special real-time operating system. Furthermore, a parallel processing method was developed, which allows to output the localization estimation at a constant high rate and still integrate the information of sensors, whose processing takes longer compared to the other sensors and therefore would lead to delays if processed sequentially.

In a comparative evaluation, it could be shown that an extended landmark model with location-independent features can contribute to a more reliable data association. Due to the improved data association, the SLAM algorithm required fewer particles to successfully close the loops. Furthermore, it could be shown that when using multiple particles, the accuracy of the mapping can also be increased by improved data association.

With the modular SLAM algorithm, a hybrid configuration utilizing a 2D grid map and a landmark map resulted in an improved mapping and localization compared to the individual methods. Also, by adding an absolute sensor, the robustness of the modular SLAM algorithm regarding the closing of large loops could be effectively increased.

In evaluations with a simulated GNSS outage and in comparison to a reference localization, the method of implicit cloning with smoothing has achieved almost the same results as alternative methods with stochastic cloning or incremental factor graph algorithms for the integration of additional relative sensors. Examinations of the computational demand have shown that the filter methods exhibit a very low utilization of the available computing time. Thus, they are well suited for real applications and with the parallel processing scheme, it is possible to also integrate more computationally demanding methods such as scan matching.

A further aspect of the thesis was the practical application of the developed methods and their use on real robotic systems. Thus, with the im-

plementation of the presented algorithms as well as the realization of the time stamp filtering and parallel processing, comprehensive improvements in localization and mapping for mobile robot systems could be realized, ranging from the acquisition of sensor data, the fusion of several relative and absolute sensors to the integration into SLAM algorithms. A large part of the presented algorithms is successfully in operation on several research platforms with different sensor equipment.

Danksagung

Zuallererst möchte ich meinem Doktorvater Herr Prof. Dr.-Ing. habil. Jürgen Beyerer meinen besonderen Dank für seine kontinuierliche Unterstützung und sein Vertrauen in meine Arbeit aussprechen. Außerdem danke ich ihm für die Möglichkeit, meine Doktorarbeit in einem äußerst inspirierenden Forschungsumfeld am Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung IOSB anzufertigen. Auch möchte ich meinem Korreferenten Herrn Prof. Dr.-Ing. Michael Heizmann für die langjährige Unterstützung und das Interesse an meiner Forschungsarbeit herzlich danken.

Ich bin meinen wunderbaren Kollegen der Abteilung MRD und allen, die zum Gelingen meiner Arbeit beigetragen haben, zutiefst dankbar für die tolle gemeinsame Zeit. Besonders möchte ich Christian Frey und Janko Petereit dafür danken, dass sie immer ein offenes Ohr hatten und mir die Freiheit gegeben haben, meine Forschungsinteressen zu verfolgen. Bei Alexander Albrecht, Christian Frese, Nina Heide, Dominik Kleiser, Philipp Woock und Angelika Zube möchte ich mich für ihre Unterstützung und das Engagement bei der Weiterentwicklung unserer mobilen Robotersysteme bedanken. Außerdem danke ich meinen Studenten für ihre wertvollen Beiträge zu unseren Projekten und meinen Forschungsarbeiten. Es ist für mich eine große Freude, Teil eines so engagierten Teams zu sein. Herzlichen Dank an alle für den Spaß bei der Arbeit und die vielen erhellenden Diskussionen.

Nicht zuletzt danke ich meiner Familie und meinen Freunden für die Unterstützung und Geduld während dieser Zeit. Ganz besonders möchte ich mich bei meiner Frau Uli für Ihren unermesslichen Rückhalt und bei meinen beiden Kindern Pepe und Luke für ihre Lebensfreude bedanken.

Inhaltsverzeichnis

Kurzfassung	i
Abstract	v
Danksagung	ix
Akronyme	xv
Symbole	xvii
1 Einleitung	1
1.1 Problemformulierung	4
1.1.1 Multi-Sensor-Fusion	5
1.1.2 Simultane Lokalisierung und Kartenerstellung	6
1.2 Zielsetzung der Arbeit	8
1.3 Wissenschaftliche Beiträge	9
1.4 Aufbau der Arbeit	10
2 Stand der Forschung	13
2.1 Multi-Sensor-Fusion für die Lokalisierung	13
2.1.1 Kalman-Filter	14
2.1.2 Stochastisches Klonen	16
2.1.3 Partikelfilter	18

2.1.4	Faktor-Graphen	20
2.2	Simultane Lokalisierung und Kartenerstellung	21
2.2.1	Methoden	21
2.2.2	Kartenerstellung	27
2.2.3	Wiedererkennung	31
2.2.4	Kombination mit Multi-Sensor-Fusion	34
3	Theoretische Grundlagen	35
3.1	Multi-Sensor-Fusion	35
3.1.1	Kalman-Filter	36
3.1.2	Erweitertes Kalman-Filter	39
3.1.3	Stochastisches Klonen	40
3.2	Simultane Lokalisierung und Kartenerstellung	45
3.2.1	Partikelfilter	47
3.2.2	Landmarkenkarte	52
3.2.3	2D-Rasterkarte	53
3.2.4	3D-NDT-Karte	56
4	Erweiterte SLAM-Verfahren	61
4.1	Kartierung mit erweitertem Landmarkenmodell	61
4.1.1	Datenzuordnung	62
4.1.2	Erweitertes Landmarkenmodell	63
4.1.3	Zusammenfassung	74
4.2	SLAM – 2D-Rasterkarte	74
4.2.1	Zusammenfassung	77
4.3	Kombination verschiedener Kartenarten	78
4.3.1	Landmarkenbasierte und dichte Karte	78
4.3.2	Zusammenfassung	80
5	Integration von Multi-Sensor-Fusion und SLAM	81
5.1	RBPF-SLAM in Kombination mit Multi-Sensor-Fusion	82
5.1.1	Zusammenfassung	85

5.2	Stochastisches Klonen	85
5.2.1	Implizites stochastisches Klonen	86
5.2.2	Explizites Klonen	90
5.2.3	Partielles explizites Klonen	94
5.2.4	Implizites stochastisches Klonen mit Smoothing	97
5.2.5	Partielles implizites Klonen mit Smoothing	104
5.2.6	Zusammenfassung	105
5.3	3D-Lokalisierung und -Kartierung	106
5.3.1	3D-Lokalisierung	106
5.3.2	3D-Kartierung	109
5.3.3	Schleifenschluss	111
5.3.4	Asynchrone Multi-Sensor-Fusion	112
5.3.5	Zusammenfassung	116
6	Ergebnisse und Auswertung	119
6.1	Erweiterte Landmarken	120
6.1.1	Evaluation	121
6.1.2	Zusammenfassung	125
6.2	SLAM – 2D-Rasterkarte	126
6.2.1	Büroumgebung	127
6.2.2	Lange Gänge	129
6.2.3	Außengelände	131
6.2.4	Zusammenfassung	133
6.3	Kombination von Landmarken und Rasterkarte	134
6.3.1	Evaluation	135
6.3.2	Zusammenfassung	139
6.4	Kombination von Multi-Sensor-Fusion und SLAM mit 2D-Rasterkarte	141
6.4.1	Evaluation	142
6.4.2	Resampling und Laufzeit	148
6.4.3	Zusammenfassung	151

6.5 SC-EKF	151
6.5.1 Implizites Klonen	153
6.5.2 Simulierter GNSS-Ausfall	155
6.5.3 Vergleich mit Faktor-Graph-SLAM	166
6.5.4 Rechenzeit	181
6.5.5 Zeitstempelfilter	187
6.5.6 Alternativer Datensatz	190
6.5.7 Zusammenfassung	197
7 Zusammenfassung und Ausblick	199
7.1 Zusammenfassung	199
7.2 Ausblick	201
Literaturverzeichnis	203
Liste der Publikationen	229
Liste der betreuten studentischen Abschlussarbeiten	233

Akronyme

6DoF	Sechs Freiheitsgrade (englisch: <i>degrees of freedom</i> – DoF)
CEKF	Komprimiertes erweitertes Kalman-Filter (englisch: <i>compressed extended Kalman filter</i>)
DARPA	Defense Advanced Research Projects Agency
DBN	Dynamisches bayessches Netz
EK	Explizites Klonen
EKF	Erweitertes Kalman-Filter (englisch: <i>extended Kalman filter</i>)
FGISAM	Faktor-Graph mit iSAM2-Optimierung
FGLM	Faktor-Graph mit Levenberg-Marquardt-Optimierung
GNSS	Globales Navigationssatellitensystem (englisch: <i>global navigation satellite system</i>)
GPGPU	Allzweck-Berechnung auf Grafikprozessoreinheit(en) (englisch: <i>general-purpose computation on graphics processing units</i>)
GPS	Global Positioning System
HSV	Farbraum mit Farbwert, Sättigung und Hellwert (englisch: <i>hue, saturation, value</i>)
IK	Implizites Klonen mit Smoothing
IMU	Inertiale Messeinheit (englisch: <i>inertial measurement unit</i>)

INS	Inertiales Navigationssystem (englisch: <i>inertial navigation system</i>)
KF	Kalman-Filter
LiDAR	Laserscanner (englisch: <i>light detection and ranging</i>)
LLA	geographischer Breitengrad, geographischer Längengrad und Höhe über dem Referenzellipsoid (WGS84) (englisch: <i>latitude, longitude, altitude</i>)
ML	Maximum Likelihood
NASA	National Aeronautics and Space Administration
NDT	Normal Distributions Transform
NN	Nächster Nachbar (englisch: <i>nearest neighbor</i>)
O	Offset
PCL	Point Cloud Library
PEK	Partielles explizites Klonen
PF	Partikelfilter (englisch: <i>particle filter</i>)
PIK	Partielles implizites Klonen mit Smoothing
RGB	Farbraum mit Rot, Grün und Blau (englisch: <i>red, green, blue</i>)
RBPF	Rao-blackwellisiertes Partikelfilter (englisch: <i>Rao-Blackwellized particle filter</i>)
ROS	Robot Operating System
RTK	Echtzeitkinematik (englisch: <i>real-time kinematics</i>)
RTS	Rauch-Tung-Striebel
SC	Stochastisches Klonen (englisch: <i>stochastic cloning</i>)
SCAAT	englisch: <i>single-constraint-at-a-time</i>
SIR	englisch: <i>sequential importance resampling</i>
SIS	englisch: <i>sequential importance sampling</i>
SLAM	Simultane Lokalisierung und Kartenerstellung (englisch: <i>simultaneous localization and mapping</i>)
UKF	Unscented-Kalman-Filter
UTM	Universal Transverse Mercator

Symbole

Konventionen

\bullet_k	Variable zum Zeitpunkt k
\bullet^k	Sequenz der Variablen bis zum Zeitpunkt k
$\bullet^{[i]}$	Zum i -ten Partikel zugehörige Variable
$\bullet_{k k-1}$	Prädizierte Variable zum Zeitpunkt k
$\bullet_{k k}$	Korrigierte Variable zum Zeitpunkt k
\bullet^s	Mittels Smoothing geglättete Variable
a	Skalar
\hat{a}	Schätzwert
\bar{a}	Arithmetisches Mittel
\mathbf{a}	Vektor
$\Delta \mathbf{a}$	Fehlervektor
\mathbf{a}_a	Teilvektor
\mathbf{a}^\top	Transponierter Vektor
$\hat{\mathbf{a}}$	Schätzvektor
$\check{\mathbf{a}}$	Augmentierter Schätzvektor
\mathbf{A}	Matrix
\mathbf{A}_{ab}	Submatrix
\mathbf{A}^\top	Transponierte Matrix
\mathbf{A}^{-1}	Inverse Matrix
$\check{\mathbf{A}}$	Augmentierte Matrix

Römische Buchstaben

a	Beschleunigungsvektor
<i>b</i>	Histogrammklasse
b	Biasvektor
<i>c</i>	Kamerakonstante
<i>d</i>	Euklidische Distanz
<i>D</i>	Histogrammdistanz
<i>e</i>	Fehler
$f(\bullet)$	Nichtlineares Systemmodell
F	Zustandsübergangsmatrix
G	Messmodell der Landmarke
$h(\bullet)$	Nichtlineares Messmodell
H	Messmatrix
I	Einheitsmatrix
J	Jacobi-Matrix bezüglich der Roboterpose
<i>k</i>	Diskrete Zeit
K	Kalman-Gain-Matrix
L	Kovarianzmatrix der Landmarke
m	Geometrische Merkmale einer Landmarke
n	Vektor von Datenzuordnungen
o	Offset-Vektor
$p(\bullet)$	Wahrscheinlichkeit
p	Punktkoordinaten
P	Systemkovarianzmatrix
<i>q</i>	Relativer Fehler
Q	Kovarianzmatrix des Systemrauschens
<i>r</i>	Radius
r	Messrauschen
R	Kovarianzmatrix des Messrauschens
<i>s</i>	Skalenfaktor
s	Positionsvektor
S	Residualkovarianzmatrix

t	Zeit
\mathbf{u}	Eingangsvektor
\mathbf{v}	Geschwindigkeitsvektor
w	Partikelgewicht
\mathbf{w}	Systemrauschen
x, y, z	Koordinaten der 3D-Position
\mathbf{x}	Zustandsvektor
$\hat{\mathbf{x}}$	Vektor der Zustandsschätzung
$\tilde{\mathbf{y}}$	Residuumsvektor
\mathbf{z}	Messvektor der Beobachtung
$\hat{\mathbf{z}}$	Prädizierter Messvektor der Beobachtung
\mathbf{Z}	Innovationskovarianzmatrix

Griechische Buchstaben

α	Horizontaler Richtungswinkel/Azimut
β	Bhattacharyya-Koeffizient
ϵ	Eigenwert
γ	Histogramm
δ	Dirac-Funktion
ζ	Zellengröße
η	Normalisierungsfaktor für Histogramm
Θ	Lagevektor
κ	Normalisierungsfaktor für Normalverteilungsschätzung
λ	Landmarke (Skalar)
$\boldsymbol{\lambda}$	Landmarke (Merkmalsvektor)
Λ	Karte
ν	Visuelle Signatur einer Landmarke
ξ	Vektor der Histogrammklassen einer Segmentfläche
π	Mittelwert der NDT
$\mathbf{\Pi}$	Kovarianzmatrix der NDT
ρ	Summe der Punktkoordinaten
σ	Standardabweichung

σ^2	Varianz
ς	Strahlaufweitung
Υ	Dyadisches Produkt der Punktkoordinaten
ϕ, θ, ψ	Eulerwinkel (Roll-, Nick- und Gierwinkel)
χ	Modifizierter Normalisierungsfaktor für Normalverteilungsschätzung
χ	Lokalisierung in 2D-Rasterkarte
\mathbf{X}	Kovarianzmatrix der Lokalisierung in 2D-Rasterkarte
ω	Drehratenvektor

Verschiedene Buchstaben

$\mathbf{0}$	Nullmatrix
\mathfrak{d}	Distanz in Pixeln
\mathcal{F}	Kumulatives Produkt der Zustandsübergangsmatrizen
\mathcal{F}	Modifiziertes kumulatives Produkt
\mathcal{L}	Logit-Form
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate Normalverteilung mit Mittelwertvektor $\boldsymbol{\mu}$ und Kovarianzmatrix $\boldsymbol{\Sigma}$
\mathcal{O}	\mathcal{O} -Notation für die Abschätzung der Laufzeit

Kapitel 1

Einleitung

Autonome Robotersysteme und künstliche Intelligenz kommen in immer mehr Umgebungen und in weiteren Aufgabenbereichen zum Einsatz [Sie04; Sic08]. Besonders die autonome Navigation mobiler Systeme hat in den letzten Jahren sehr viel Aufmerksamkeit erregt.

Mit den Wettbewerben der Defense Advanced Research Projects Agency (DARPA), der Grand Challenge in einer Wüstenumgebung und später der Urban Challenge in einer städtischen Umgebung wurden die Möglichkeiten autonomer mobiler Systeme auch einer breiteren Öffentlichkeit bekannt [Thr06b; Urm08]. Teilergebnisse der bisherigen Forschungen im Themenfeld der autonomen Fahrzeuge sind bereits in Fahrerassistenzsysteme eingeflossen [Win09]. Die Entwicklung zu komplett selbstfahrenden Autos im Straßenverkehr ist zwar schon weit fortgeschritten, befindet sich jedoch noch immer in der Weiterentwicklung [Gui11; Bad19].

Andere Anwendungsfelder autonomer mobiler Systeme sind abseits der Straße in unwegsamem Gelände oder in für Menschen gefährlichen oder unzugänglichen Umgebungen. Hierzu zählen unter anderem die Entwicklungen von Robotern zur Erkundung von Planeten wie einem Mars-Rover [Kno03; Sch14]. Weitere Forschungsfelder umfassen Rettungs- und

Aufklärungsroboter oder Roboter zur Überwachung von Liegenschaften [Cub17; Emt07]. Besonders auch in gefährlichen Umgebungen wie bei der zivilen Minenräumung ist der Einsatz von autonomen mobilen Systemen Gegenstand von Forschungsaktivitäten [Por14; Cur15]. Auch für die schnelle Aufklärung und die Durchführung von Manipulationsaufgaben zur Unterstützung der Rettungsarbeiten in Katastrophenszenarien wird an mobilen Robotern geforscht, um möglichst nicht noch weitere Menschenleben zu gefährden [Kun14; Fre15; Tsa17]. Neben speziell für die entsprechenden Einsatzszenarien entwickelten mobilen Plattformen werden jedoch auch prototypisch schwere Arbeitsmaschinen wie Hydraulikbagger mittels zusätzlicher Sensorik und Rechentechnik automatisiert, um beispielsweise kontaminierte Erdschichten abzutragen oder bei der Deponiesanierung zu unterstützen und damit in für den Menschen potentiell gesundheitsgefährdenden Umgebungen eingesetzt zu werden [Emt17; Pet19].

Für die autonome Navigation mobiler Robotersysteme sind gewisse Fähigkeiten notwendig. Diese können in die Grundfunktionalitäten Lokalisierung, Kartierung und Bewegungsplanung unterteilt werden, siehe Abbildung 1.1. Die Kartierung beinhaltet dabei neben der Umgebungsmodellierung auch die Umgebungserfassung, welche auch Perzeption genannt wird. Die Planung kann weiter in Pfad- oder Trajektorienplanung und -regelung unterteilt werden [Pet13; Pet14]. Die grundlegenden Fähigkeiten sind dabei stark voneinander abhängig und werden für höhere Autonomiegrade miteinander kombiniert [Mak02; Fai09]. So wird beispielsweise aus der Kombination von Lokalisierung und Kartierung die simultane Lokalisierung und Kartenerstellung, in der englischen Literatur unter dem Akronym SLAM für *simultaneous localization and mapping* bekannt [Dur06a; Dur06b]. Für die Exploration hingegen ist neben der Kartierung auch noch eine Pfadplanung notwendig, wobei die aus der Umgebungserfassung erstellte Karte als Grundlage für die Pfadplanung und Hindernisvermeidung dient.

Je nach Einsatzszenario oder Aufgabenstellung bestehen spezifische Unterschiede bei der Ausprägung der genannten notwendigen Fähigkei-

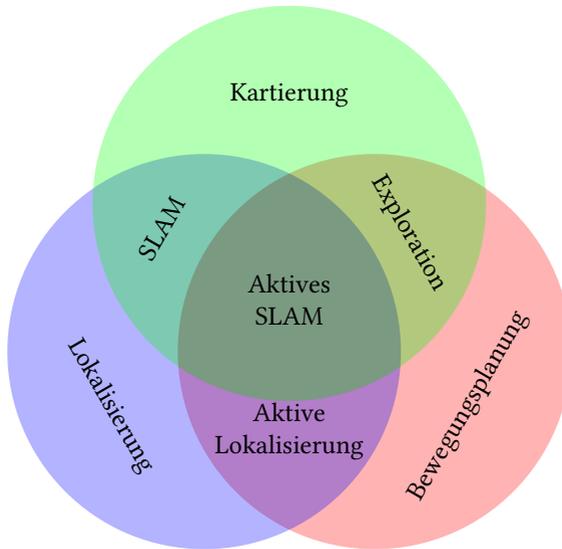


Abbildung 1.1: Fähigkeiten eines mobilen Roboters zur autonomen Navigation. Adaptiert von [Mak02].

ten. So gehört für selbstfahrende Autos, welche am Straßenverkehr teilnehmen, zur Umgebungserfassung beispielsweise auch die Straßenschilderkennung oder Fußgängererkennung [Wah14; Yan18]. Für das Fahren in unstrukturierten und unwegsamen Umgebungen spielen diese Aspekte hingegen keine Rolle, sondern dort ist eher die Erkennung von leichter Vegetation und deren Unterscheidung von harten Hindernissen wie Steinen relevant [Bra07; Ngu10]. Bei mobilen Plattformen, welche am Straßenverkehr teilnehmen, wird vor allem in urbanen Gebieten meist auf präzise Karten zurückgegriffen, welche zuvor mit speziell ausgerüsteten Kartierungsfahrzeugen aufgenommen wurden [Zie14; Pog18]. Dagegen ist für die Lokalisierung in unstrukturierten Umgebungen vorher meist keine Karte vorhanden. Zudem sind in den erwähnten Szenarien abseits der Straße, z. B. zur Aufklärung in Katastrophenszenarien oder bei Erdarbeiten zur Dekontaminierung, Änderungen der Umgebung zu erwarten,

seien diese durch externe Faktoren im Katastrophenfall hervorgerufen oder selbst induziert bei den Aushubarbeiten bei der Dekontaminierung, sodass eine vorher aufgezeichnete Karte bald nicht mehr aktuell wäre. Änderungen können natürlich auch im Straßenverkehr vorkommen, hierbei handelt es sich jedoch meist um kleinere lokale Änderungen, wie geänderte Beschilderung oder Straßenführung, welche separat mittels Änderungsdetektion behandelt werden können [Jo18; Pau18].

1.1 Problemformulierung

In der vorliegenden Arbeit werden von den grundlegenden Fähigkeiten die Themen der Lokalisierung und Kartierung anhand der Sensoren des mobilen Robotersystems in unstrukturierten Außenumgebungen behandelt, wobei die Fusion mehrerer Sensoren betrachtet wird und in der Kombination SLAM-Verfahren Verwendung finden. Die Bewegungsplanung und damit auch die Exploration werden in dieser Arbeit nicht behandelt.

Die Lokalisierung, d. h. die Bestimmung der Position und Orientierung, bzw. Pose, des Robotersystems in seiner Umgebung auf Basis von Sensordaten, kann als eine der fundamentalsten Problemstellungen betrachtet werden, wenn es um die Ausstattung eines mobilen Roboters mit autonomen Fähigkeiten geht [Cox91]. Für die Lokalisierung ist meist die Kombination mehrerer Sensoren notwendig, da unter Umständen nicht alle Freiheitsgrade von einem einzelnen Sensor gemessen werden können [Sic08]. Außerdem sind die Messungen aller Sensoren durch Rauschen und andere Störungen beeinträchtigt, sodass durch die Fusion mehrerer Sensoren die Robustheit und Präzision erhöht werden können.

Eine weitere essentielle Fähigkeit ist die Erstellung konsistenter und präziser Karten. Diese können später für die autonome Navigation wiederverwendet werden. Mit den Karten ist es für den mobilen Roboter möglich, sich bereits befahrene Pfade zu merken. Somit kann verhindert werden, dass er mehrfach in Sackgassen gerät oder lange Zeit damit verbringt, um Hindernisse herum zu manövrieren [Kon09].

Wenn noch keine Karte der Umgebung vorhanden ist, muss diese erst von dem Roboter anhand seiner Sensordaten erstellt werden. Dies beinhaltet die bereits erwähnte simultane Kombination von Lokalisierung und Kartenerstellung. Da auch hier alle Messungen der Sensoren fehlerbehaftet sind, ist dies eine der größten Herausforderungen im Themenfeld der autonomen Navigation, denn die daraus resultierenden Fehler in der Lokalisierung und in der Karte sind voneinander abhängig [Dur06a; Dur06b].

1.1.1 Multi-Sensor-Fusion

Meist ist ein einzelner Sensor nicht ausreichend, um eine eindeutige Interpretation zu erlauben, da er nur begrenzte Informationen über die Umgebung liefert [Dur88]. Auch für die Lokalisierung von mobilen Plattformen sind meist mehrere Sensoren notwendig, damit alle Freiheitsgrade geschätzt werden können. Vor allem bei der Lokalisierung im dreidimensionalen Raum, d. h. der Bestimmung von sechs Freiheitsgraden (englisch: *degrees of freedom* – 6DoF), sind mehrere Sensoren notwendig, um alle Freiheitsgrade schätzen zu können [Wen11].

Außerdem sind alle Sensoren fehlerbehaftet, d. h., deren Daten sind verrauscht und durch andere Störungen beeinflusst. Ein weiteres Ziel der Multi-Sensor-Fusion ist es daher, durch Kombination von mehreren Sensoren eine Verbesserung der Genauigkeit zu erreichen, möglichst unter Berücksichtigung der jeweiligen Unsicherheiten, sodass unterschiedlich genaue Sensoren entsprechend eingebracht werden können. Dabei ist vor allem das Ausgleichen gegensätzlicher Nachteile bzw. sensorspezifischer Fehler von Bedeutung. So sind relativ messende Sensoren lokal oder kurzzeitig sehr genau, können jedoch über die Zeit drifteten. Ein relativ messender Sensor bedeutet dabei, dass eine Differenz zwischen dem aktuellen und einem vergangenen Zustand gemessen wird. Im Gegensatz dazu sind die Fehler von absolut messenden Sensoren zwar global begrenzt, können aber z. B. bei einem globalen Navigationssatellitensystem (englisch: *global navigation satellite system* – GNSS), wie dem Global Positioning System

(GPS, früher NAVSTAR GPS), lokal sehr ungenau sein [Bra17]. Moderne GNSS-Empfänger zeichnen sich dadurch aus, dass sie für die Positionsbestimmung die Satelliten mehrerer Systeme verwenden und dadurch eine verbesserte Zuverlässigkeit und Präzision erzielen, weil sie eine größere Anzahl an gleichzeitig verfügbaren Satelliten nutzen können.

Je nach einsatzspezifischen Anforderungen oder vorhandener Sensorausstattung sind mehrere absolute und relative Sensoren zu integrieren. Während mehrere absolute Sensoren meist einfach zu integrieren sind, ist es je nach verwendeter Methode nicht ohne Weiteres möglich, mehr als einen relativen Sensor zu integrieren [Rou01].

Eine weitere Herausforderung ergibt sich bei der Integration mehrerer Sensoren aus der oftmals fehlenden Synchronisation der Sensoren untereinander. D. h., es kann nicht davon ausgegangen werden, dass die Daten der unterschiedlichen Sensoren zur gleichen Zeit verfügbar sind. Außerdem werden die Daten meist mit unterschiedlichen Raten von den Sensoren ausgegeben [Wel96].

Des Weiteren können einzelne Sensoren ausfallen und es ist anzustreben, dass durch die Verwendung redundanter oder aber auch heterogener Sensoren eine gewisse Ausfallsicherheit erreicht wird [Wen11].

1.1.2 Simultane Lokalisierung und Kartenerstellung

Die Erstellung einer Karte der Umgebung mit den Sensoren eines mobilen Robotersystems ist eine komplexe Problemstellung, denn für die Lokalisierung wird eine genaue Karte benötigt und umgekehrt wird für die Kartierung eine präzise Schätzung der Lokalisierung benötigt [Leo91]. So sind auch hier die Messungen fehlerbehaftet und zusätzlich bestehen die Abhängigkeiten zwischen Karte und Lokalisierung. Das bedeutet, dass die Unsicherheit in der Lokalisierung abhängig von den Fehlern in der bisher erstellten Karte ist und im Folgenden die erstellte Karte ungenau wird. Deshalb ist es essentiell, dass diese Abhängigkeiten in den Algorithmen berücksichtigt werden, d. h. Schätzungen der Lokalisierung und die Kartierung

simultan berechnet werden – daher der Begriff simultane Lokalisierung und Kartenerstellung (englisch: *simultaneous localization and mapping* – SLAM) [Dur06a; Dur06b].

Bei den entstehenden Karten ist jedoch nicht nur die Präzision relevant, sondern es wird zusätzlich noch davon die Konsistenz unterschieden. Unter Konsistenz wird hierbei die topologisch korrekte Modellierung der Umgebung verstanden, d. h., dass z. B. ein bereits besuchter Ort oder Weg wiedererkannt wird und nicht mehrfach in der Karte auftaucht. Während eine geringe Präzision meist kompensierbar ist bzw. über weitere Beobachtungen Verbesserungen erzielt werden können, kann sich eine fehlende Konsistenz eher als fatal erweisen, sofern ein Algorithmus nicht gleichzeitig mehrere potentielle Kartenhypothesen berücksichtigt. Das Wiedererkennen von bereits besuchten Orten ist dabei vor allem beim Schleifenschluss relevant. Von Schleifenschluss wird gesprochen, wenn eine mobile Plattform wieder an einen bereits besuchten und kartierten Ort zurückkehrt, nachdem sie in unbekanntem Gebiet unterwegs war. Hierbei können sich über die Zeit Fehler akkumuliert haben, die die Lokalisierungsgenauigkeit stark verschlechtern und damit auch die Lokalisierung in der bereits aufgenommen Karte erschweren können [Thr05].

Bei Karten, welche auf Landmarken¹ basieren, ist deren robuste Wiedererkennung ebenfalls eine wichtige Voraussetzung, damit eine konsistente Karte entstehen kann. Unter Wiedererkennung wird hier die Datenzuordnung von Beobachtungen zu bereits kartierten Landmarken verstanden. Kommt es hier zu Fehlern, können Landmarken mehrfach in der Karte auftauchen oder Beobachtungen zu falschen Landmarken zugeordnet werden. Dies kann zu inkonsistenten Karten führen und verschlechtert in jedem Fall die erzielbare Präzision [Mon07].

1. Eine Landmarke ist ein statisches Objekt, das sich deutlich von der Umgebung abgrenzt und durch Wiedererkennung zur Orientierung dient.

1.2 Zielsetzung der Arbeit

Die vorliegende Arbeit beschäftigt sich mit der Erarbeitung von Methoden für die Kombination und Integration mehrerer Sensoren zur präzisen Lokalisierung und Kartenerstellung in heterogenen Außenumgebungen. D. h., die entwickelten Verfahren sollen sowohl in urbanen als auch in unstrukturierten Bereichen möglichst robust funktionieren.

Es wird die Fragestellung behandelt, wie zusätzliche Sensoren für SLAM eingesetzt werden können, um die Kartenerstellung robuster und präziser zu machen. Eine Steigerung der Robustheit kann durch zuverlässigeres Wiedererkennen von Landmarken und dadurch auch das Wiedererkennen von bereits besuchten Orten erreicht werden, was das Erzeugen von konsistenten Karten vereinfachen und die Präzision der Karten steigern soll. Neben der zuverlässigeren Wiedererkennung soll die Kombination verschiedener Kartenarten in einem hybriden Algorithmus untersucht werden. Auch hier ist das Ziel die Steigerung der Robustheit und der Präzision der Kartenerzeugung.

Ein weiterer Schritt ist die Kombination von SLAM mit Multi-Sensor-Fusion zur Lokalisierung mit einem zusätzlichen absoluten Positionssensor. Durch einen ergänzenden absoluten Positionssensor kann die globale Lokalisierungsunsicherheit beschränkt werden, was wiederum das Schließen von Schleifen vereinfacht. Es wird so bewusst auf die Beschränkung auf ein minimales Sensorsetup verzichtet, um den Einfluss von zusätzlichen Sensoren aufzuzeigen.

Bei der Multi-Sensor-Fusion ist außerdem die Integration zusätzlicher relativer Sensoren, d. h. Sensoren, deren Messungen sich auf die Differenz zwischen zwei Zuständen beziehen, von Interesse. Diese können jedoch nicht ohne Weiteres mit einfachen Filtermethoden integriert werden, weshalb in der Arbeit auch erweiterte Filtermethoden behandelt werden, um die probabilistisch korrekte Integrierbarkeit von zusätzlichen relativ messenden Sensoren zu ermöglichen. So können diese unter Berücksichtigung der jeweiligen Unsicherheiten zur Verbesserung der Präzision beitragen.

Durch die Integration heterogener Sensoren kann zudem ein wechselseitiges Kompensieren von sensorspezifischen Fehlern erreicht werden. Das bedeutet, es können gegenseitige Nachteile ausgeglichen werden. Des Weiteren soll durch die Kombination mehrerer Sensoren und entsprechende Algorithmen eine gewisse Ausfallsicherheit geschaffen werden.

Die verschiedenen Sensoren sollten sich außerdem ohne zusätzliche aufwändige Synchronisierungshardware integrieren lassen. Vor allem bei prototypischen Plattformen ist dies von Vorteil, wenn neue Sensoren zu integrieren sind oder für Tests bzw. zu Vergleichszwecken temporär alternative Sensoren anzubinden sind.

Ein weiterer Aspekt der Arbeit ist eine möglichst praxisnahe Umsetzung der entwickelten Methoden und deren Evaluation mit realen Daten. Es soll außerdem bei allen Verfahren eine weiche Echtzeitfähigkeit gegeben sein. Die weiche Echtzeitfähigkeit (englisch: *soft real-time*) bedeutet, dass vorgegebene Zeiten für die Berechnungen nicht überschritten werden sollten, aber gelegentlich überschritten werden dürfen [Bur09].

1.3 Wissenschaftliche Beiträge

Die wissenschaftlichen Beiträge der vorliegenden Arbeit beschränken sich nicht auf eine einzelne Methode, sondern widmen sich der im vorangegangenen Abschnitt aufgezeigten vielschichtigen Fragestellung, mehrere Sensoren für die Lokalisierung und Kartenerstellung einzusetzen und zu kombinieren. Aus der Vielschichtigkeit leitet sich die Aufteilung in unterschiedliche Themenblöcke ab:

- Ein **erweitertes SLAM-Verfahren** wurde entwickelt, um die Wiedererkennung von Landmarken robuster zu machen. Hierfür wurde ein erweitertes Landmarkenmodell entworfen, welches neben der Position weitere positionsunabhängige Merkmale einbezieht und somit die Wiedererkennung von der Lokalisierungsunsicherheit weniger abhängig macht [Emt10c].

- Mit der **Kombination verschiedener Kartenarten** wurde ein hybrides SLAM-Verfahren entwickelt, welches die Genauigkeit der Kartierung verbessert [Emt12b].
- Durch die **Kombination von Multi-Sensor-Fusion mit SLAM**, indem zusätzlich ein absoluter Positionssensor mit einbezogen wird, konnte gezeigt werden, dass durch die Beschränkung der globalen Lokalisierungsunsicherheit das Schließen von Schleifen vereinfacht wird [Emt12a].
- Mit einer Methode auf Basis des stochastischen Klonens in Kombination mit Smoothing wurde ein neuartiges Filterverfahren zur **Fusion von mehreren relativen und absoluten Sensoren** für die Lokalisierung und Kartierung in sechs Freiheitsgraden entwickelt [Emt18b; Emt19b; Emt19a].
- Für eine Abrundung der Arbeit hinsichtlich der Praxistauglichkeit wurden zusätzlich noch ein Verfahren zur **Synchronisation über Zeitstempel** und ein Verfahren zur **Parallelverarbeitung für die Integration verzögerter Sensordaten** entwickelt [Kle20; Emt14; Emt18b].

Sämtliche vorgestellten neuen Methoden und Verfahren wurden anhand von realen Sensordaten evaluiert und wo möglich mit etablierten wissenschaftlichen Verfahren in Vergleich gesetzt.

1.4 Aufbau der Arbeit

Das folgende **Kapitel 2** gibt einen Überblick über den Stand der Forschung. Hierbei liegt das Augenmerk auf der Multi-Sensor-Fusion für die Lokalisierung mobiler Robotersysteme und der simultanen Lokalisierung und Kartenerstellung. Anschließend werden in **Kapitel 3** die theoretischen Grundlagen zu Multi-Sensor-Fusion und SLAM behandelt, wobei sich die vorgestellten Themen auf die für die Arbeit relevanten Methoden und

Verfahren beschränken. In den **Kapiteln 4** und **5** werden die in dieser Arbeit entstandenen Methoden und Verfahren vorgestellt. Den Anfang machen hier die erweiterten Methoden zur simultanen Lokalisierung und Kartierung in Abschnitt 4.1 mit dem erweiterten Landmarkenmodell und in Abschnitt 4.3 mit der Kombination verschiedener Kartenarten. Es folgt in Abschnitt 5.1 die Kombination von Multi-Sensor-Fusion mit SLAM zur Verbesserung des Schleifenschlusses mittels zusätzlichem absoluten Sensor. Die Fusion von mehreren relativen und absoluten Sensoren wird in Abschnitt 5.2 vorgestellt, gefolgt von dem Verfahren zur Synchronisation und dem Verfahren zur Parallelverarbeitung in Abschnitt 5.3.4. Anschließend werden in **Kapitel 6** die jeweiligen Ergebnisse zu allen neu entwickelten Methoden anhand von Evaluationen mit realen Sensordaten und im Vergleich zu bestehenden Methoden präsentiert. Abschließend wird die Arbeit in **Kapitel 7** zusammengefasst und ein Ausblick auf mögliche weiterführende Arbeiten gegeben.

Kapitel 2

Stand der Forschung

In diesem Kapitel wird der Stand der Forschung der Multi-Sensor-Fusion im Kontext der Lokalisierung und der simultanen Lokalisierung und Kartenerstellung für autonome mobile Robotersysteme aufgeführt.

2.1 Multi-Sensor-Fusion für die Lokalisierung

Für die Fusion mehrerer Sensoren zur Lokalisierung in der Navigation autonomer mobiler Systeme sind verschiedene Varianten des Kalman-Filters die am weitesten verbreiteten Methoden. Sie gehören zu der Klasse der rekursiven bayesschen Schätzer und es gehen daher auch die Messunsicherheiten der Sensoren in die Schätzung mit ein. Daneben gibt es auch andere Filterverfahren wie Partikelfilter für nicht-parametrische Dichtepäsentationen oder auf graphischen Modellen basierende Verfahren wie Faktor-Graphen, jedoch sind Kalman-Filter aufgrund ihrer Leistungsfähigkeit und Robustheit in den meisten Anwendungen zur Lokalisierung in der Navigation vorherrschend.

2.1.1 Kalman-Filter

Das Kalman-Filter ist ein Optimalfilter für lineare Systeme und ist in unterschiedlichen Varianten mit Erweiterungen eines der meistverwendeten Verfahren für die Fusion mehrerer Sensoren. Das Filter wurde als Least-Squares-Optimalschätzer für lineare Systeme von Rudolf E. Kalman 1960 vorgestellt [Kal60]. Die Einordnung in die bayessche Statistik als rekursives Bayes-Filter erfolgte erst später [Ho64]. Das Filter ist am bekanntesten als rekursiver bayesscher Schätzer für lineare Systeme mit gaußverteilter Systemmodell, es ist aber auch ein optimaler linearer Schätzer, wenn das Rauschen nicht gaußverteilt ist [Sim06]. Die rekursive Filterung ist in zwei Phasen aufgeteilt. Die erste Phase ist die Prädiktion, bei der der Zustand vom vorherigen Filterschritt mittels eines Systemmodells, welches auch Messungen von relativen Sensoren beinhalten kann, zu einer A-priori-Schätzung fortgeführt wird. Die zweite Phase ist der Korrekturschritt, welcher eine Messung mit der A-priori-Schätzung kombiniert und eine verbesserte A-posteriori-Schätzung liefert [Wel95; Goe99].

Bereits sehr kurze Zeit nach der Vorstellung des Kalman-Filters erfolgte 1961 die Erweiterung für nichtlineare Systeme in Form des erweiterten Kalman-Filters (englisch: *extended Kalman Filter* – EKF) für die Apollo-Missionen der National Aeronautics and Space Administration (NASA) [McG85]. Das EKF ermöglicht die Anwendung auf nichtlineare Systeme durch Linearisierung um die aktuelle Schätzung. Da die nichtlinearen Funktionen nicht auf die Matrizen angewendet werden können, werden die Matrizen der ersten partiellen Ableitungen – die Jacobi-Matrizen – mit der aktuellen Schätzung ausgewertet und in die linearen Kalman-Filtergleichungen eingesetzt. In der Anwendung für die Lokalisierung autonomer mobiler Roboter werden hauptsächlich erweiterte Kalman-Filter eingesetzt, da hier nichtlineare System- und Messmodelle vorliegen [Mor08; Gre10; Agh13].

Ein weiteres Anwendungsgebiet bei der Lokalisierung ist die INS/GNSS-Integration. Dabei werden ein inertiales Navigationssystem (englisch: *inertial navigation system* – INS), welches Beschleunigungsmesser und

Drehratensensoren kombiniert, und ein GNSS fusioniert. Durch die Fusion können die lokalen Abweichungen des GNSS durch das INS und umgekehrt die systematische Messabweichung der INS-Sensoren durch das GNSS kompensiert werden. Auch bei der INS/GNSS-Integration kommen meistens erweiterte Kalman-Filter zum Einsatz [Moh01; Wen11; Gro08; Far08].

Eine wichtige Eigenschaft des Kalman-Filters ist die Möglichkeit der Fusion von mehreren asynchronen Sensoren in Form von individuelle Einzelmessungen, welche auch nur partielle Information für die Korrekturen liefern können. Dies wurde in den Arbeiten von Welch et al. anhand einer auf einem EKF basierenden Methode gezeigt [Wel96; Wel97]. Die Methode wurde SCAAT genannt, was für *single-constraint-at-a-time* steht und bedeutet, dass Einzelmessungen als individuelle Korrekturschritte integriert werden können. Dies erlaubt neben der einfacheren Integrierbarkeit mehrerer Sensoren eine hohe Ausgaberate der Schätzung im Vergleich zu anderen Systemen, da nicht mehrere Messungen zusammengefasst werden müssen, sondern einzeln als individuelle Korrekturschritte eingespeist werden können.

Die Sigma-point-Kalman-Filter sind eine Alternative zum erweiterten Kalman-Filter für stark nichtlineare Systeme [Mer04]. Sie zeichnen sich durch deterministisches Sampling der Gaußdichte aus und während das EKF nur die ersten zwei statistischen Momente, d. h. Mittelwert und Varianz, modelliert, können mit den Sigma-point-Kalman-Filtern auch höhere Momente erfasst werden. Das bekannteste Filter dieser Art ist das Unscented-Kalman-Filter (UKF) [Jul95; Jul97]. Während beim EKF die nichtlinearen Funktionen linearisiert werden, werden beim UKF die durch deterministisches Sampling gezogenen Punkte durch die nichtlinearen Funktionen transformiert und anschließend aus den transformierten Samples die Dichte rekonstruiert [Jul04b; Jul04a]. Beim UKF können so die ersten drei statistischen Momente abgebildet werden. Somit können Verbesserungen für stark nichtlineare Systeme bei vergleichbarem Rechenaufwand erzielt werden. Ein weiterer Vorteil der nichtlinearen Transfor-

mation der Punkte ist, dass keine Jacobi-Matrizen hergeleitet werden müssen. Nachteilig sind die zusätzlichen freien Parameter für die Wahl des Samplings, welche zusätzlich bestimmt werden müssen.

Mit den Sigma-point-Kalman-Filtern können Nichtlinearitäten zwar sehr viel besser erfasst werden als mit einem EKF, sie werden jedoch für die Lokalisierung eher selten angewendet, da sie hier kaum Vorteile bringen. So haben sich für die Orientierungsschätzung mittels Quaternionen keine Vorteile des UKF gegenüber einem EKF gezeigt [LaV03]. Auch für die 2D-Lokalisierung eines mobilen Roboters mittels Ultraschallsensoren hatte sich gezeigt, dass ein EKF genauso gute Ergebnisse wie ein UKF liefern kann, d. h., dass auch für diesen Fall die Nichtlinearitäten im Modell nicht so groß sind, als dass ein UKF von Vorteil wäre [DA15]. Für die INS/GNSS-Integration konnte ebenfalls gezeigt werden, dass ein Sigma-point-Kalman-Filter keine Vorteile hinsichtlich der Präzision oder Robustheit gegenüber einem EKF bietet [Wen11].

2.1.2 Stochastisches Klonen

Während mehrere absolute Sensoren einfach in die in den vorangegangenen Abschnitten vorgestellten Kalman-Filter-Varianten zu integrieren sind, ist die Fusion von mehreren relativen Sensoren mit ihnen nicht formal korrekt möglich, da die Filter auf der Markov-Annahme basieren. Die Markov-Annahme besagt, dass der aktuelle Zustand gegeben dem vorherigen Zustand unabhängig von allen weiter zurückliegenden Zuständen ist und dass die aktuelle Messung gegeben dem aktuellen Zustand bedingt unabhängig von allen vorherigen Messungen und Zuständen ist [Sär13]. Eine relative Messung hängt jedoch von einem vorherigen Zustand ab und verletzt damit die Markov-Annahme.

Ein einzelner relativer Sensor kann für die Prädiktion verwendet und mit beliebig vielen absoluten Sensoren mittels der genannten Kalman-Filter-Varianten fusioniert werden. Kommt jedoch ein zweiter relativer Sensor hinzu, kann dieser nicht ohne Weiteres als Korrekturschritt inte-

griert werden. Weil eine relative Messung Differenzen zwischen einem vorangegangenen und dem aktuellen Zustand misst, werden hierdurch Korrelationen eingeführt, welche die Markov-Annahme verletzen. Methoden zur Rekonstruktion können zwar angewendet werden, sie sind jedoch mit Nachteilen verbunden. Eine Methode ist die Berechnung einer pseudo-absoluten Messung aus dem letzten Zustand und der relativen Messung [Hof99]. Diese ist jedoch suboptimal, da hierdurch das Filter überoptimistisch wird und folglich andere Sensoren zu wenig eingehen. Alternativ kann durch Ableitung aus der relativen Messung eine absolute Messung rekonstruiert werden, wie z. B. durch Bildung des Differenzenquotienten aus einer Strecke und der Zeit die Durchschnittsgeschwindigkeit ableitbar ist. Hierzu müssen die Messungen jedoch in einer hohen Rate zur Verfügung stehen und auch im Zustandsraum abgebildet werden [Rou02]. Zusätzlich kommt es bei der Ableitung von verrauschten Signalen zu einem Anstieg des hochfrequenten Rauschens [Pic16]. Eine weitere Möglichkeit besteht darin, die relativen Sensoren alternierend einzusetzen, wobei bevorzugt der genaueste Sensor verwendet wird und der alternative Sensor nur dann, wenn der Hauptsensor aussetzt [Agr06]. Hierdurch werden zwar keine Approximationen notwendig, es werden jedoch nicht alle vorhandenen Informationen einbezogen, was ebenfalls sub-optimal ist.

Zur formal korrekten Integration von zusätzlichen relativen Messungen stellten Roumeliotis et al. *stochastic cloning* mit einem Kalman-Filter vor [Rou01; Rou02]. Hierbei wird der Zustand mit einem Klon des zurückliegenden Zustands erweitert, womit sich dessen Korrelationen mit dem aktuellen Zustand korrekt abbilden lassen. Das Verfahren wurde daher Stochastic-cloning-Kalman-Filter (SC-KF) genannt. In den genannten Arbeiten wurden zwei relative Sensoren fusioniert und es konnte gezeigt werden, dass in diesem Fall nicht explizit geklont werden muss, sondern eine Rekonstruktion der erweiterten Systemkovarianzmatrix erfolgen kann. Auch in späteren Arbeiten wurden zwei relative Sensoren fusioniert, aber es wurde darauf hingewiesen, dass diese vereinfachende Rekonstruk-

tion bei der Hinzunahme weiterer Sensoren nicht mehr gültig ist und ein explizites Klonen erforderlich wird [Mou06; Mou07].

In [Chi11] wurden ebenfalls nur relative Sensoren auf einem Laufroboter miteinander fusioniert. Neben einer IMU wurden visuelle Odometrie und die Odometrie der Beine integriert. In dieser Arbeit wurde ein indirektes Informationsfilter statt eines Kalman-Filters eingesetzt. Beim Informationsfilter handelt es sich um die inverse Formulierung des Kalman-Filters, d. h., es werden statt dem Zustand und der Kovarianzmatrix die Information und die Informationsmatrix zugrunde gelegt [Gro02]. In der Arbeit wird nicht der komplette geschätzte Zustand geklont, sondern nur relevante Zustandsvariablen, um Rechenzeit zu sparen. Als relevante Zustandsvariablen wurden die der Pose identifiziert, da die verwendeten relativen Sensoren die Differenz zwischen zwei Posen messen. In [Kle11] wurde für die Personennavigation mit einem EKF mit stochastischem Klonen zusätzlich ein GPS als absoluter Sensor integriert, wobei ebenfalls nicht der komplette geschätzte Zustand geklont wurde, sondern nur die Position.

In [Lyn13] wird ebenfalls die Fusion mit absoluten Sensoren integriert. Das Filter ist so aufgebaut, dass wie in den ursprünglichen Veröffentlichungen von Roumeliotis et al. kein explizites Klonen erfolgt, obwohl dies bei der Hinzunahme absoluter Sensoren erfolgen müsste, siehe [Mou07]. Für die zusätzlichen relativen Sensoren werden jedoch in den Zustand weitere Variablen aufgenommen, welche die Drift der relativen Sensoren abbilden.

2.1.3 Partikelfilter

Das Partikelfilter ist ebenfalls ein rekursives Bayes-Filter. Es ist jedoch kein parametrisches Filter wie das Kalman-Filter, sondern modelliert die Wahrscheinlichkeit durch Sampling des Zustandsraums und kann dadurch prinzipiell beliebige Dichten repräsentieren. Als rekursives Bayes-Filter existieren bei einem Partikelfilter ebenfalls zwei Phasen wie beim Kalman-Filter. In der ersten Phase wird für jedes Partikel basierend auf der Sys-

temdynamik ein Sample des neuen Zustands gezogen und in der zweiten Phase mit den Beobachtungen gewichtet [Che03].

Im Kontext der Navigation mobiler Robotersysteme werden Partikelfilter meist für SLAM eingesetzt, siehe Abschnitt 2.2. Bei der reinen Lokalisierung können Partikelfilter jedoch auch von Vorteil sein, wenn schon eine Karte vorhanden ist, vor allem wenn Mehrdeutigkeiten zu erwarten sind, da auch multimodale Dichten abgebildet werden können. In [Del99] wurde ein Partikelfilter für die 2D-Lokalisierung von mobilen Robotern in bekannten Karten mittels Sonar oder 2D-LiDAR (englisch: *light detection and ranging* – LiDAR) vorgestellt. Da mit dem Partikelfilter eine multimodale Dichte modelliert wird, konnte der Roboter aus einer völlig unbekanntem Anfangspose über die Zeit auch in mehrdeutigen Umgebungen seine Pose eindeutig bestimmen. Für die Lokalisierung in einer bereits vorhandenen Karte existieren noch weitere auf Bayes-Filtern basierende Methoden wie topologische Ansätze oder auf Rasterisierung basierende Ansätze, diese haben jedoch signifikante Nachteile bezüglich der Genauigkeit bzw. Effizienz [Fox03].

Zur Lokalisierung mobiler Roboter mit Partikelfiltern ohne bekannte Karte gibt es sehr wenige Arbeiten und Veröffentlichungen. Zwar konnte für die 2D-Lokalisierung eines mobilen Roboters in Simulationen eine bessere Schätzung im Vergleich zum erweiterten Kalman-Filter erzielt werden, allerdings mit einem vielfachen Rechenaufwand [Rig07]. Bei der Schätzung einer Lokalisierung in 3D ergibt sich im Vergleich zur Schätzung in 2D ein sehr viel höherdimensionaler Zustandsraum, da neben den sechs Freiheitsgraden zusätzlich die systematischen Abweichungen der Sensoren einer inertialen Messeinheit (englisch: *inertial measurement unit* – IMU) hinzukommen und dadurch sehr viele Partikel notwendig werden, um den Zustandsraum noch adäquat abzudecken. In einer Untersuchung konnte jedoch durch eine Marginalisierung, bei der der lineare Teil mit einem Kalman-Filter und nur der nichtlineare Teil mit einem Partikelfilter berechnet wurde, die notwendige Partikelzahl signifikant verringert werden. Den-

noch hat sich ein zum Vergleich herangezogener EKF für die Posenschätzung als präziser und robuster erwiesen [Ble08].

2.1.4 Faktor-Graphen

Bei Faktor-Graphen handelt es sich um ein graphisches Modell, welches zur Schätzung der A-posteriori-Wahrscheinlichkeit über die Zeit dient [Ind12]. Ein Faktor-Graph ist ein bipartiter Graph und besteht aus Variablenknoten, welche den zu schätzenden Systemzuständen entsprechen, und Faktorknoten, welche die Messungen oder A-priori-Informationen repräsentieren. Da es sich um einen bipartiten Graphen handelt, verlaufen die Kanten nur zwischen Variablenknoten und Faktorknoten. Bei der Lokalisierung sind jeweils die einzelnen Posen Variablenknoten und die Navigationslösung des gesamten Pfades wird dann über eine nichtlineare Optimierung des Graphen geschätzt [Del12].

Da der Rechenaufwand zur Berechnung des gesamten Graphen mit dessen Größe anwächst, sind diese zunächst nicht für Anwendungen mit Echtzeitanforderungen geeignet. Der Anstieg des Rechenaufwands hängt hierbei stark von der Art der Problemstellung ab, da diese die Struktur des Graphen beeinflusst. Bei der Lokalisierung sind die Knoten des Graphen meist nur mit jeweils wenigen anderen Knoten verknüpft, sodass mittels Ansätzen für dünnbesetzte lineare Berechnungen (englisch: *sparse linear algebra*) eine effizientere Berechnung realisiert werden kann [Gri10; Del12]. In [Ind13] wurden zusätzlich zwei entscheidende Erweiterungen angewendet, um eine echtzeitfähige Lokalisierung in 3D mittels IMU und visueller Odometrie (auch bekannt als visuelle inertielle Navigation) zu realisieren. Die erste Erweiterung ist ein IMU-Faktor, welcher mehrere aufeinanderfolgende IMU-Messungen bündelt, damit die Knotenanzahl kleiner gehalten werden kann. Die zweite Erweiterung basiert auf einem Verfahren – iSAM2 –, mit dem der Graph nicht in jedem Schritt komplett neu optimiert werden muss, sondern inkrementell auf vorherige Ergebnisse aufbauen kann [Kae12]. Mit diesen Erweiterungen konnten zwar geringe Berech-

nungszeiten erreicht werden, es traten aber einige Ausreißer auf, welche über den Echtzeitanforderungen lagen. In einer späteren Veröffentlichung mit einem verbesserten IMU-Faktor und mit zusätzlich besser optimiertem iSAM2 konnte dann die prinzipielle Echtzeitfähigkeit für visuelle inertielle Navigation gezeigt werden [For15].

2.2 Simultane Lokalisierung und Kartenerstellung

Bei der simultanen Lokalisierung und Kartenerstellung sind verschiedene Aspekte relevant. So gibt es unterschiedliche SLAM-Methoden für die Lösung des Problems. Des Weiteren können, je nach eingesetztem perzeptorischem Sensor, verschiedene Kartenarten unterschieden werden. Außerdem ist die Wiedererkennung von bereits kartierten Landmarken oder Orten von großer Bedeutung. Die genannten Aspekte lassen sich je nach Ansatz nicht immer scharf trennen, da teilweise die Kartenart eng mit der SLAM-Methode oder der Art des Wiedererkennungsverfahrens verknüpft ist.

2.2.1 Methoden

Da alle Sensordaten fehlerbehaftet sind und die Abhängigkeiten der Lokalisierung und der Karte zu modellieren sind, kommen bei SLAM meist probabilistische Methoden zum Einsatz. Ein konzeptueller Durchbruch wurde mit der Erkenntnis erzielt, dass das SLAM-Problem konvergiert, wenn die Lokalisierung und die Karte als ein gemeinsames Schätzproblem formuliert werden, d. h., dass die volle A-posteriori-Wahrscheinlichkeit über die Lokalisierung und die Karte berechnet wird [Dur06a].

Bayes-Filter

Ein großer Anteil der SLAM-Verfahren basiert auf rekursiven Bayes-Filtern, was erweiterte Kalman-Filter-Varianten und Partikelfilter umfasst [Dur06a; Aul08].

Mit der Annahme eines gaußverteilten Bewegungs- und Messmodells sind auch bei SLAM erweiterte Kalman-Filter im Einsatz. Die A-posteriori-Wahrscheinlichkeit des SLAM-Problems ist dann beim EKF-SLAM ebenfalls eine multivariate Gaußverteilung [Dis01].

Da die volle A-posteriori-Wahrscheinlichkeit berechnet wird, werden bei EKF-SLAM alle Landmarken Teil des Zustands. D. h., der Speicherbedarf wächst unter Vernachlässigung der Pose des Roboters quadratisch mit der Anzahl der Landmarken N und die Laufzeit liegt bei mindestens $\mathcal{O}(N^{2,3737})$ aufgrund der Matrixmultiplikationen¹.

Um dem bei vielen Landmarken anwachsenden Rechenaufwand des EKF-SLAM entgegenzuwirken, wurden verschiedene Verfahren entwickelt. Eines davon reduziert die Anzahl der Landmarken, indem relevante Landmarken selektiert und unwichtige entfernt werden [Dis00]. Durch das Entfernen von Landmarken wird jedoch die Lokalisierungsgenauigkeit verringert, sodass hier ein Kompromiss gefunden werden muss. Mit einem anderen Verfahren, dem komprimierten EKF (englisch: *compressed EKF* – CEKF), werden nur lokale SLAM-Aktualisierungen mit der vollen Rate durchgeführt und später auf die volle SLAM-Lösung propagiert [Gui01; Gui02b]. Dadurch ist der Rechenaufwand meistens niedrig, während nur Aktualisierungen für die Landmarken in der lokalen Umgebung berechnet werden. Erst wenn das lokale Gebiet verlassen wird, muss eine volle Aktualisierung durchgeführt werden. Diese volle Aktualisierung ist ohne Informationsverlust möglich, sodass die Lokalisierungsgenauigkeit nicht verringert wird. Eine weitere Methode basiert auf der inversen Formulierung des EKF, dem erweiterten Informationsfilter. Sie ist eine approxi-

1. Die Laufzeit der Matrizenmultiplikation ist $\mathcal{O}(N^3)$ mit dem naiven Standardalgorithmus. Mit optimierten Algorithmen liegt die Laufzeit niedriger und befindet sich momentan bestenfalls bei $\mathcal{O}(N^{2,3737})$ [Wil12].

mative Methode, welche auf der Erkenntnis beruht, dass die meisten Elemente der Informationsmatrix außerhalb der Diagonalen nahe 0 sind und mit entsprechender Verarbeitung effiziente Aktualisierungsalgorithmen für dünnbesetzte Matrizen eingesetzt werden können [Thr04; Eus05].

Neben dem hohen Rechenaufwand ist bei EKF-SLAM nachteilig, dass nur unimodale Dichten modelliert werden können und nur eine Hypothese für die Datenzuordnungen der Beobachtungen zu den kartierten Landmarken verfolgt werden kann, was den Algorithmus anfällig für mehrdeutige Situationen macht. Diese können gravierende Folgen haben, da die Integration falscher Beobachtungen nicht mehr rückgängig gemacht werden kann. Treten zu viele Fehlzuordnungen auf, kann der Algorithmus divergieren. Um den Einfluss falscher Zuordnungen zu verringern, wurde im Kontext der Zielverfolgung ein Verfahren entwickelt, welches mehrere Hypothesen verfolgen kann [Rei79]. Da für jede Hypothese ein eigenes Filter berechnet werden muss, vervielfacht sich jedoch der Rechenaufwand, weshalb dieser Ansatz noch nicht mit EKF-SLAM kombiniert wurde [Dur06b].

Partikelfilter können hingegen implizit mehrere Hypothesen bezüglich der Datenzuordnung verfolgen und auch multi-modale Dichten modellieren. Partikelfilter repräsentieren eine Wahrscheinlichkeitsdichte mit einem Set von gewichteten Samples, welche entsprechend der Wahrscheinlichkeit des Zustands verteilt und gewichtet sind. Sie sind allerdings nur für niedrig-dimensionale Probleme geeignet, da die Anzahl der benötigten Partikel mit der Dimension des Zustandsraums ansteigt. Da die Dimension des SLAM-Problems mit der Anzahl der Landmarken ansteigt, ist die direkte Anwendung von Partikelfiltern nicht möglich. Es ist jedoch möglich, die benötigte Anzahl von Partikeln durch die Anwendung einer Rao-Blackwellisierung zu reduzieren. Bei dem daraus entstehenden rao-blackwellisierten Partikelfilter (englisch: *rao-blackwellized particle filter* – RBPF) wird nicht mehr nur die Pose des Roboters geschätzt, sondern dessen Pfad, wodurch die Landmarken bedingt unabhängig werden und das SLAM-Problem in einzelne, unabhängige Landmarkenschätzer faktorisiert werden kann [Mur99; Dou00a]. Mit dem FastSLAM genannten Algorith-

mus wurde eine effiziente Umsetzung eines RBPF vorgestellt, bei dem der Pfad des Roboters durch das Partikelfilter geschätzt wird und jedes Partikel eine eigene Karte mit jeweils unabhängigen Landmarkenschätzern mit sich führt [Mon02]. Ein mögliches Problem bei dem RBPF ist, dass die Partikel durch die Systemdynamik in der ersten Phase möglicherweise zu weit verteilt werden und viele Partikel in der zweiten Phase sehr geringe Gewichte erhalten werden. Mit FastSLAM 2.0 wurde dies behandelt, indem die Beobachtungen bereits in die erste Phase mit eingehen, wodurch die Partikel näher an der A-posteriori-Wahrscheinlichkeit liegen und damit auch höhere Gewichte erhalten [Mon03b]. Es konnte gezeigt werden, dass durch diese Verbesserung im Vergleich zum originalen FastSLAM wesentlich weniger Partikel notwendig sind, und es konnte sogar ein Beweis geführt werden, dass der Algorithmus mit nur einem Partikel konvergiert.

Wenn die zwei Phasen des Partikelfilters für lange Zeit fortgesetzt werden, kann es zu einer sogenannten Degeneration kommen, was bedeutet, dass viele Partikel ein sehr geringes Gewicht bekommen. Da diese Partikel einen Zustand mit sehr geringer Wahrscheinlichkeit repräsentieren, ist es sinnvoll, diese auszusortieren. Für ein statistisch korrektes Aussortieren wurde die Methode des *sequential importance resampling* (SIR) von Rubin entwickelt [Rub88]. Bei dieser Methode wird vor jeder ersten Phase ein neues Set aus ungewichteten Partikeln aus dem aktuellen Set gezogen mit Zurücklegen mit einer Wahrscheinlichkeit proportional zum jeweiligen Gewicht. Diese Methode findet auch in den erwähnten FastSLAM und FastSLAM 2.0 SLAM-Algorithmen Verwendung.

Durch das Ziehen mit Zurücklegen werden Partikel mit hohen Gewichten mehrfach gezogen und Partikel mit geringen Gewichten werden dafür aussortiert. Wenn das Resampling in jedem Schritt ausgeführt wird, kann u. U. über die Zeit die Diversität zu sehr verringert werden. D. h., die adäquate Abdeckung des Zustandsraums ist dann nicht mehr gegeben. Um diesen Effekt abzuschwächen, wurde ein adaptives Resampling eingeführt, was nur dann durchgeführt wird, wenn die Gewichte der Partikel zu stark variieren [Dou00b].

Für die 2D-Kartierung haben sich die Partikelfilter vorteilhaft gegenüber den Kalman-Filtern gezeigt. Für SLAM mit 3D-Sensoren sind sie jedoch weniger geeignet. Zum einen werden sehr viele Partikel für die Abdeckung des größeren Zustandsraums benötigt und zum anderen wird eine Karte pro Partikel benötigt, was zu unpraktikablen Speicheranforderungen führen würde.

Graph-basierte Verfahren

Das SLAM-Problem lässt sich auch als Graph formulieren. Die Elemente des Graphen sind die Posen des Roboters über die Zeit und Messungen werden als Randbedingungen zwischen den Posen modelliert. Die beste Lösung entspricht der Konstellation aller Posen, welche mit allen Messungen am konsistentesten ist. Da den Posen auch Messungen der Umgebung zugeordnet sind, wird so auch implizit eine Karte mitgeschätzt. Für das Finden der Lösung wird meist auf Methoden der kleinsten Quadrate (englisch: *least squares*) zur Fehlerminimierung zurückgegriffen und da der volle Pfad geschätzt wird, werden diese Ansätze auch oft als Smoothing bezeichnet [Gri10].

In [Lu97] wurde das SLAM-Problem als ein Netzwerk aus räumlichen Relationen durch Rad-Odometrie und das Scan-Matching mit 2D-Laserscans formuliert. Der Pfad wird anschließend mittels eines iterativen Optimierungsalgorithmus geschätzt. Die initiale Schätzung der Posen für den iterativen Algorithmus wird aus der Odometrie berechnet. Wenn diese zu ungenau ist, kann es vorkommen, dass der Algorithmus in ein lokales Minimum konvergiert, was zu einer inkonsistenten Karte führen kann. Um eine Erkennung von Schleifen zu ermöglichen, wurde in darauf aufbauenden Arbeiten neben anderen Verbesserungen ein zusätzliches Verfahren eingeführt, um schon besuchte Orte durch Korrelation von Kartenteilen wiederzuerkennen [Gut99; Gut00].

Da der Rechenaufwand mit der Größe des Graphen stark ansteigt, gibt es Methoden für die Reduktion der Komplexität des Gesamtproblems. Mit dem Atlas genannten Ansatz wurde hierfür eine Graphstruktur mit lokalen

Karten vorgestellt. Dabei wurde innerhalb der lokalen Karten mit einem erweiterten Kalman-Filter kartiert und nur die globalen Zusammenhänge zwischen den lokalen Karten mit einem Graphen modelliert, wodurch dieser mit weniger Elementen effizienter optimiert werden konnte [Bos03; Bos04].

In [Kae12] wurde das schon in Abschnitt 2.1.4 erwähnte Verfahren iSAM2 für SLAM vorgestellt. Es erlaubt eine sehr effiziente inkrementelle Optimierung des Faktor-Graphen und ist damit prinzipiell für Echtzeitanwendungen geeignet. So ist bei der Befahrung und Kartierung neuer Gebiete nur die Neuberechnung eines kleinen Teils des Graphen notwendig. Bei dem Schließen von Schleifen müssen jedoch große Teile des Graphen neu optimiert werden, sodass immer wieder längere Berechnungszeiten auftreten können und damit die Echtzeitfähigkeit beeinträchtigt wird. Um eine echtzeitfähige Schätzung zu erlauben, wurde daher in [Wil14] eine Aufteilung des Faktor-Graphen in eine Echtzeitkomponente und eine Komponente für das Schließen von Schleifen vorgestellt. Während erstere Komponente ein echtzeitfähiges Filter ist, läuft die andere Komponente parallel im Hintergrund und optimiert den ganzen Pfad inklusive Schleifenschlüssen mit iSAM2.

In [Zha14b] wurde ein Verfahren für LiDAR-Odometrie und Kartierung vorgestellt, welches Laserscans als einzige Sensordaten verwendet, wobei für die Bewegungsschätzung ein Scan-Matching anhand von extrahierten Merkmalspunkten von Linien und Ebenen zum Einsatz kommt. In [Sha18b] wurde es um das Schließen von Schleifen mittels ICP-Algorithmus (englisch: *iterative closest point* – ICP) erweitert, wobei für die finale Schätzung des Pfades für die Kartierung ebenfalls das o. g. iSAM2 zum Einsatz kommt. In [Ye17] wurde ein Verfahren vorgestellt, welches für das Scan-Matching eine Punkt-zu-Ebene-Variante des ICP-Algorithmus anwendet und für die Bewegungsschätzung mit den Daten einer IMU fusioniert, wobei auch im Vergleich mit dem Verfahren aus [Zha14b] gezeigt werden konnte, dass mit den zusätzlichen Daten der IMU eine robustere und präzisere Kartierung erzielt werden kann.

2.2.2 Kartenerstellung

Eine Karte dient dem mobilen Roboter als Modell der Umgebung für die Navigation. Die Art der erstellten Karte ist abhängig von dem auf dem mobilen Roboter verwendeten perzeptorischen Sensor und der gegebenenfalls vorhandenen Vorverarbeitung der Sensordaten. Die Karte kann aus unverarbeiteten Sensordaten, wie z. B. Punktwolken, bestehen oder aber die Punkte beispielsweise in einem Raster aggregieren. Mit diesen Arten von Karten erhält man eine dichte Repräsentation der Umgebung. Es können jedoch auch Landmarken aus den Sensordaten extrahiert werden und die entstehenden Karten bestehen folglich aus einem Tupel von Landmarken.

2D-Sensoren

Die Nutzung von Rasterkarten für 2D-SLAM wurde 1987 in [Elf87] als Belegtheitskarte (englisch: *occupancy grid map*) eingeführt und das Konzept wurde in vielen späteren SLAM-Verfahren genutzt [Häh03; Eli03; Gri05; Koh11; Sha18a]. Die Fläche wird dabei in ein gleichmäßiges Raster unterteilt und jede Zelle enthält eine Belegtheitswahrscheinlichkeit. Die Modellierung mit einer Belegtheitswahrscheinlichkeit hat den Vorteil, dass sowohl die Information über Objekte in der Umgebung als auch über den freien Raum abbildbar ist. Diese Art der Karte lässt sich mit Sensoren erstellen, welche Entfernungen zu Objekten messen, wie einem Array aus Sonarsensoren oder einem 2D-LiDAR.

In [Lu97] wurden unverarbeitete Punktwolken für die Kartierung verwendet. Wie in Abschnitt 2.2.1 dargestellt, handelt es sich hier um eine graph-basierte Methode, bei der jedoch keine explizite Karte aufgebaut wird. In den darauf aufbauenden Arbeiten wurden für das Schließen von Schleifen temporäre Rasterkarten erzeugt, um mit diesen durch Korrelation die beste Übereinstimmung des aktuellen Kartenteils mit einem bereits besuchten Teil zu finden [Gut99; Gut00].

Weit verbreitet sind auch Karten aus Punktlandmarken, welche in verschiedenen Veröffentlichungen der auf Bayes-Filter basierenden Algorithmen, wie EKF-SLAM und FastSLAM bzw. FastSLAM 2.0, Verwendung fanden [Dis01; Dur06a; Mon02; Mon03b]. Die Punktlandmarken sind aus Sensordaten extrahiert und lediglich über ihren Ort definiert, was sie anfällig für Fehlzuordnungen macht.

Neben den Karten aus Punktlandmarken lassen sich auch Rasterkarten mit Partikelfiltern kombinieren. In [Häh03] wurde die Kartierung einer Belegtheitskarte mit dem FastSLAM-Algorithmus vorgestellt. Da jedes Partikel seine eigene Kartenschätzung besitzt, müssen diese im Speicher gehalten und beim Resampling mit kopiert werden. Um hier den Speicher- und Kopieraufwand zu verringern, wurde in [Eli03] eine effiziente Kartenrepräsentation eingeführt, bei der nicht für jedes Partikel eine komplette Karte vorgehalten werden muss. Stattdessen wurde eine Datenstruktur eingeführt, bei der nur eine einzige Karte existiert, aber dafür in jeder Zelle eine Baumstruktur gespeichert ist, die die Information enthält, welche Partikel die Belegtheit in der jeweiligen Zelle verändert haben. Eine weitere Möglichkeit, den Aufwand zu verringern, wurde durch die Verwendung von FastSLAM 2.0 erzielt, wobei die notwendige Partikelanzahl im Vergleich zu [Häh03] wesentlich verringert werden konnte [Gri05; Gri07].

3D-Sensoren

Sensoren wie 3D-LiDAR oder Tiefenkameras liefern 3D-Punktwolken und für deren Kartierung sind die Anforderungen weitaus höher, da mehr Freiheitsgrade zu schätzen sind und die Datenmenge im Vergleich zu 2D-Sensoren sehr viel größer ist. Folglich sind auch die Speicheranforderungen für die 3D-Karten höher.

Für 3D-SLAM werden häufig graph-basierte Methoden eingesetzt, bei denen keine explizite Karte aufgebaut wird, sondern Randbedingungen von paarweisem Scan-Matching zwischen Punktwolken in den Graph eingefügt werden [Nüc09]. Für das Scan-Matching wird oft der ICP-Algorithmus herangezogen [Sur03; Kat06; Nüc07]. Als punkt-basierte Methode sind

keine Annahmen über die Struktur der Umgebung und die anzutreffenden Objekte notwendig, weshalb es sich auch besonders für unstrukturierte Außenumgebungen eignet [Rus01]. Es existieren weitere Varianten des ICP-Algorithmus, wie z. B. Punkt-zu-Fläche-Registrierung, um das Scan-Matching in strukturierten Umgebungen genauer zu machen, wodurch jedoch die Präzision in unstrukturierten Umgebungen verschlechtert wird [Pom13]. Eine Variante, welche durch eine lokale probabilistische Flächenmodellierung für heterogene Umgebungen geeignet ist und speziell für Punktwolken mehrlagig scannender Sensoren entwickelt wurde, ist der generalisierte ICP-Algorithmus (Generalized-ICP – GICP) [Seg09].

Das Scan-Matching mit Punktwolken ist nur eine implizite Kartierung und die konkrete Karte muss in einem Nachverarbeitungsschritt aufgebaut werden. Dies kann beispielsweise als Punktwolkenkarte erfolgen, was eine Aggregation aller Punktwolken bedeutet. Wenn über lange Zeit kartiert wird, ist dies jedoch mit einem enormen Speicheraufwand verbunden, da die Anzahl der Punkte mit der Zeit ansteigt. Außerdem wird die Karte an mehrfach besuchten Orten eine sehr viel höhere Punktdichte aufweisen als an seltener besuchten Orten. Um den Speicheraufwand zu verringern und für eine gleichmäßigere Punktdichte zu sorgen, kann die Punktwolkenkarte anschließend räumlich ausgedünnt werden. Es existieren jedoch auch andere speichereffizientere Methoden, welche im Folgenden vorgestellt werden.

Eine sehr kompakte Repräsentation ist eine gerasterte Höhenkarte, welche in einem 2D-Raster in jeder Zelle einen Höhenwert speichert. Mit dieser Art von Karte lassen sich Oberflächen modellieren, jedoch keine Durchgänge oder überhängenden Objekte. In [Tri06] wurde deswegen eine Erweiterung der Höhenkarte vorgestellt, welche für jede Zelle mehrere Höhenlevel speichert und so auch mehrere Ebenen und vertikale Strukturen modellieren kann.

Ein anderer Ansatz basiert auf einer Karte, welche eine 3D-Rasterung der Umgebung enthält und für jede belegte Zelle eine lokale Normalverteilung der darin enthaltenen Punkte schätzt, was als Normal Distributions

Transform (NDT) bezeichnet wird [Mag07]. Die Zellen sind kubische Volumenelemente und werden daher auch als Voxel bezeichnet (von englisch *volume element* in Anlehnung an Pixel für *picture element*). Diese Art der Karte erlaubt eine sehr kompakte 3D-Repräsentation, wobei durch die Normalverteilungen die Formen der Oberflächen in der Umgebung modelliert sind, weshalb die Elemente der Karten auch manchmal als Surfel (kurz für englisch: *surface element*) bezeichnet werden [Klä12]. In [Sto13] wurde gezeigt, dass in einigen Umgebungen eine NDT-Karte mit vielfach geringerer Auflösung äquivalent zu einer reinen Belegtheitskarte ist und dabei auch die Aktualisierung wesentlich weniger Rechenaufwand benötigt.

Die Rasterung der Umgebung muss nicht zwangsläufig regelmäßig sein, sondern kann auch hierarchisch erfolgen. Die in [Wur10] vorgestellte und Octomap genannte Karte verwendet eine Octree-basierte Unterteilung der Umgebung, welche in jedem Knoten ein kubisches Volumen repräsentiert, das ebenfalls als Voxel bezeichnet wird. Jedes Voxel beinhaltet eine Belegtheitswahrscheinlichkeit wie die in Abschnitt 2.2.2 erläuterten 2D-Belegtheitskarten und kann somit auch Informationen über freien Raum beinhalten.

In [Moo11] besteht die Karte aus Ebenen, welche aus den Punktwolken extrahiert werden. Die Karte ist dabei in ein 3D-Raster aufgeteilt und jede Zelle beinhaltet maximal eine Ebene. Die aus der aktuellen Punktwolke extrahierten Ebenen werden für die Lokalisierung in der Karte verwendet, indem eine Registrierung der Ebenen mit einer Variante des ICP-Algorithmus durchgeführt wird. Da in diesem Ansatz nur Ebenen für die Kartierung und Registrierung verwendet werden, ist er hauptsächlich in urbanen oder anderen strukturierten Umgebungen anwendbar.

Kameras

In der Literatur wird bei der Verwendung von bildgebenden Sensoren von Visual-SLAM gesprochen. In den merkmalsbasierten Visual-SLAM-Verfahren werden zunächst in einem Vorverarbeitungsschritt Merkmale aus den Bildern extrahiert, welche dann als Landmarken dienen. Die meis-

ten Arbeiten verwenden lokale Bildmerkmale und es existieren Veröffentlichungen mit verschiedenen lokalen Merkmalen, wie z. B. SIFT-Merkmale in [Sim05; Eli06; Gil06], SURF-Merkmale in [Ste08] oder ORB-Merkmale in [Mur15]. Da in den meisten Umgebungen viele dieser Merkmale in den Bildern gefunden werden, wird auch eine große Menge an Landmarken in die Karte eingetragen. Die Dichte der resultierenden Karten liegt zwischen den dünnbesetzten Landmarkenkarten und den dichten Karten aus LiDAR-Daten.

Neben den merkmalsbasierten Visual-SLAM-Verfahren existieren auch direkte Methoden wie beispielsweise LSD-SLAM, das direkt auf den Intensitätsbildern arbeitet und eine semi-dichte Punktwolkenkarte aus Tiefenbildern konstruiert [Eng14].

2.2.3 Wiedererkennung

Die Wiedererkennung spielt in SLAM in verschiedenen Kontexten eine große Rolle. So ist bei Landmarkenkarten die Wiedererkennung bereits kartierter Landmarken eine wichtige Voraussetzung für die Erstellung einer konsistenten Karte. Schon wenige Fehlzuordnung können zum Scheitern des SLAM-Algorithmus führen. Außerdem müssen bereits besuchte Orte wiedererkannt werden, um Schleifen zu schließen. Auch hier kann schon eine nicht oder falsch erkannte Schleife zu einer inkonsistenten Karte führen.

Landmarken

Die Wiedererkennung im Kontext der Landmarken wird auch Datenzuordnung genannt, da hier die aktuelle Beobachtung einer bereits kartierten Landmarke zugeordnet werden muss. Die Datenzuordnung ist ein schon lange bekanntes Problem in der Literatur über Zielverfolgung (englisch: *target tracking*) und das einfachste Verfahren ist die Maximum-Likelihood-Datenzuordnung, auch bekannt als Zuordnung des nächsten Nachbarn (englisch: *nearest neighbor* – NN) [Mon07]. Diese einfache Zuordnung mit-

tels Maximum Likelihood (ML) kann sehr effektiv sein, wenn die Landmarken nicht zu dicht beieinander liegen. Sind jedoch Mehrdeutigkeiten vorhanden, können diese leicht zu Fehlzuordnung führen.

Liegen Landmarken dicht beieinander, kann dies vor allem bei der Verwendung von Punktländmarken leicht zu Fehlzuordnungen führen, da diese nur über ihren Ort definiert sind. Nahe beieinanderliegende Landmarken werden jedoch sehr wahrscheinlich auch gleichzeitig beobachtet. Anstatt die beobachteten Landmarken einzeln sequenziell zuzuordnen, ist es sinnvoll, diese gemeinsam zuzuordnen, wofür verschiedene Verfahren entwickelt wurden [Nei01; Bai02]. In [Ram07] wurde eine Methode vorgestellt, welche ein erweitertes Landmarkenmodell verwendet, und gezeigt, dass damit eine robustere Wiedererkennung erzielt werden kann. Das einfache Punktländmarkenmodell aus Daten eines 2D-LiDARs wird dabei um eine aus Kameradaten extrahierte, ansichtsbasierte Komponente erweitert.

Mit Partikelfiltern werden mehrere Kartenhypothesen und damit auch implizit individuelle Datenzuordnungshypothesen verfolgt. Dies kann sogar noch erweitert werden, indem nicht immer die am besten passende Zuordnung verwendet wird, sondern die Zuordnungen zufällig mit einer Wahrscheinlichkeit proportional zu ihrem Likelihood gezogen werden, was als Monte-Carlo-Datenzuordnung in [Mon03a] vorgestellt wurde. Eine weitere Methode wurde in [Nie03] vorgestellt, bei der im Falle von Mehrdeutigkeiten für jede alternative Hypothese ein zusätzliches Partikel zum Filter hinzugefügt wird. Damit die Anzahl von Partikeln nicht stetig anwächst, wird deren Anzahl beim nächsten Resampling wieder auf die ursprüngliche Anzahl reduziert.

Da bei dichten Karten keine Landmarken im eigentlichen Sinne vorhanden sind, kommen hier Scan-Matching-Verfahren zum Einsatz, welche die relative Position und Ausrichtung der aktuellen Beobachtung zur Karte oder zu einer vorangegangenen Beobachtung bestimmen sollen. Hierbei handelt es sich beispielsweise um Gradientenabstiegsverfahren oder den ICP-Algorithmus in verschiedenen Varianten. Es kann auch hier zu Feh-

lern kommen, da es sich zum einen um lokale Verfahren handelt und zum anderen auch die Umgebung mehrdeutig sein kann [Gri07; Pom13; Sta08].

Schleifen

Wenn der mobile Roboter für einige Zeit unbekanntes Gebiet erkundet und wieder an einen bekannten Platz zurückkehrt, muss dies anhand der aufgezeichneten Karte wiedererkannt werden. Dies wird in der SLAM-Literatur in Englisch als *loop closure* bezeichnet, also das Schließen von Schleifen, und ist eine der größten Herausforderungen, da die Bewegungsunsicherheit nach der Befahrung unbekanntes Gebietes besonders groß ist und sich die im vorherigen Abschnitt beschriebene Wiedererkennung von Landmarken damit als besonders schwierig erweist [Mon07].

Bei Landmarkenkarten ist die Gefahr für Fehlzuordnungen beim Schließen von Schleifen besonders hoch. Deshalb wurden verschiedene und zum Teil sehr aufwändige Verfahren entwickelt, um bereits besuchte Plätze wiederzuerkennen. In [Ho05] wurde ein Ansatz mit mehreren Sensoren verfolgt. So werden als Landmarken SIFT-Merkmale aus Kamerabildern mit räumlichen Deskriptoren von Segmenten von 2D-Laserscans kombiniert, wobei diese Landmarken ausschließlich für das robuste Detektieren von Schleifen verwendet werden. Das im vorherigen Abschnitt bereits vorgestellte Verfahren von [Ram07] verwendet ebenfalls Landmarken aus LiDAR- und Kameradaten und es konnte gezeigt werden, dass damit Schleifen zuverlässiger geschlossen werden können.

Bei dichten 2D-Karten aus Punktwolken wurden für das Schließen von Schleifen in [Gut99], wie in Abschnitt 2.2.2 bereits beschrieben, zusätzliche Rasterkarten erzeugt, um mit diesen durch Korrelation die beste Übereinstimmung des aus aktuellen Beobachtungen erzeugten Kartenteils mit bereits besuchten Teilen der Karte zu ermitteln. Bei den 2D-Rasterkarten kommt für das Schließen von Schleifen ebenfalls das Scan-Matching zum Einsatz [Gri05]. In [Nüc07] werden 3D-Punktwolkenkarten erzeugt und der Schleifenschluss erfolgt über ein Abstandsmaß in Kombination mit

dem ICP-Algorithmus, welcher auch für das Scan-Matching der aufeinanderfolgenden Punktwolken Verwendung findet.

In [Koh11] wird EKF-SLAM mit einer 2D-Rasterkarte durchgeführt und es wird gezeigt, dass die Genauigkeit ausreicht, um kleinere Schleifen zu schließen, ohne dass potentielle Mehrdeutigkeiten gesondert behandelt werden müssen.

2.2.4 Kombination mit Multi-Sensor-Fusion

Für die meisten SLAM-Methoden ist lediglich ein Sensor für die Messung der Bewegung und ein Sensor zur Perception der Umgebung notwendig, weshalb die Fusion weiterer Sensoren in nur wenigen Veröffentlichungen eine Rolle spielt. So wurde in [Fre10] angemerkt, dass in den bisherigen Forschungen zu SLAM der Fokus auf die Abwesenheit von absoluten Sensoren wie GPS und wenig auf reale Anwendungen gelegt wurde.

In [Gui02a] wurde ein hochgenaues RTK-GPS² neben Rad-Odometrie und 2D-LiDAR als zusätzlicher Sensor in einem EKF-SLAM-Algorithmus verwendet und es konnte gezeigt werden, dass es als absoluter Sensor zu einer Reduzierung der Unsicherheiten in der Lokalisierung und der Karte beiträgt.

Auch in [Thr06a] kam ein zusätzliches GPS zum Einsatz, wobei hier ein Graph-SLAM-Verfahren verwendet wurde. Auch hier konnte das GPS zu einer Verbesserung der Präzision der Lokalisierung und Kartierung beitragen und dies selbst wenn das GPS nur sporadisch verfügbar war.

Eine Integration eines hochgenauen RTK-GPS in einen RBPF-SLAM-Algorithmus wurde in [Sha18a] vorgestellt. Dabei werden die GPS-Messungen über die Partikelgewichte fusioniert und es konnte auch hier eine präzisere und konsistentere Kartierung erzielt werden.

2. RTK steht im Englischen für *real-time kinematics* und bezeichnet ein Verfahren, mit dem mittels zusätzlicher Korrektursignale von einer Referenzstation im Idealfall Genauigkeiten im Zentimeterbereich erzielt werden können.

Kapitel 3

Theoretische Grundlagen

Im Folgenden werden die theoretischen Grundlagen der bestehenden Methoden zur Multi-Sensor-Fusion und SLAM erläutert, auf denen diese Arbeit aufbaut. Es werden an den entsprechenden Stellen Verweise auf die in dieser Arbeit entwickelten Methoden angegeben, um sie in Kontext zu den bestehenden Methoden zu setzen.

3.1 Multi-Sensor-Fusion

Die in dieser Arbeit verwendeten Methoden zur Multi-Sensor-Fusion basieren auf Bayes-Filtern. Diese erlauben durch ihre probabilistische Modellierung eine Fusion unter Berücksichtigung der jeweiligen Unsicherheiten.

Ein rekursives Bayes-Filter basiert auf der Markov-Annahme, d. h., dass zum Zeitpunkt k alle Information in dem aktuellen Zustand \mathbf{x}_k und in der aktuellen Messung \mathbf{z}_k vorhanden ist. Der aktuelle Zustand \mathbf{x}_k ist bedingt unabhängig von allen weiter zurückliegenden Zuständen gegeben dem vorherigen Zustand \mathbf{x}_{k-1} und die aktuelle Messung \mathbf{z}_k ist bedingt unabhängig von allen zurückliegenden Messungen und Zuständen gegeben dem aktuellen Zustand \mathbf{x}_k [Sär13].

3.1.1 Kalman-Filter

Das Kalman-Filter gehört zu den rekursiven Bayes-Filtern und ist ein optimaler Zustandsschätzer für lineare Systeme im Sinne eines minimalen quadratischen Fehlers [Sim06].

Es wird ein lineares dynamisches Modell angenommen, bei dem der Zustand \mathbf{x}_k zum Zeitpunkt k aus dem Zustand zum Zeitpunkt $k - 1$ hervorgeht:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k, \quad (3.1)$$

mit dem Zustandsübergangsmodell \mathbf{F}_k , der optionalen Eingabe \mathbf{u}_k und deren Dynamik \mathbf{B}_k . Das System ist durch ein additives Systemrauschen \mathbf{w}_k gestört, welches als normalverteilt angenommen wird:

$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k). \quad (3.2)$$

Der wahre Zustand \mathbf{x}_k wird zum Zeitpunkt k durch die Messung \mathbf{z}_k beobachtet:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{r}_k, \quad (3.3)$$

mit dem Messmodell \mathbf{H}_k und dem additiven Messrauschen \mathbf{r}_k , welches ebenfalls als normalverteilt angenommen wird:

$$\mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k). \quad (3.4)$$

Da der wahre Zustand \mathbf{x} selbst nicht beobachtbar ist, wird anhand von Beobachtungen die Zustandsschätzung $\hat{\mathbf{x}}$ in rekursiven Filterschritten berechnet. Dabei kann ein Filterschritt des Kalman-Filters in zwei Phasen unterteilt werden. Die erste Phase wird Prädiktion genannt und bildet zum Zeitpunkt k die A-priori-Zustandsschätzung $\hat{\mathbf{x}}_{k|k-1}$, indem die A-posteriori-Zustandsschätzung $\hat{\mathbf{x}}_{k-1|k-1}$ des vorangegangenen Zeitschritts $k - 1$ propagiert wird. Die Propagation erfolgt dabei mit der Zustands-

übergangsmatrix \mathbf{F}_k und optional der zusätzlichen Eingabe \mathbf{u}_k und deren Dynamik \mathbf{B}_k :

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k . \quad (3.5)$$

Die A-priori-Systemkovarianzmatrix $\mathbf{P}_{k-1|k-1}$ wird ebenfalls propagiert und es wird ein additives Systemrauschen mit der Kovarianzmatrix \mathbf{Q}_k hinzugefügt, welches die Modellunsicherheiten berücksichtigt:

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^\top + \mathbf{Q}_k . \quad (3.6)$$

Die zweite Phase beinhaltet die Aktualisierung durch eine Messung \mathbf{z}_k und wird Korrekturschritt genannt. Die Messung wird entsprechend der Unsicherheiten mit der A-priori-Zustandsschätzung $\hat{\mathbf{x}}_{k|k-1}$ kombiniert und das Resultat ist eine verbesserte A-posteriori-Zustandsschätzung $\hat{\mathbf{x}}_{k|k}$. Dabei wird zunächst die prädiizierte Zustandsschätzung $\hat{\mathbf{x}}_{k|k-1}$ mittels der Messmatrix \mathbf{H}_k in den Messraum abgebildet. Aus der Differenz zur eigentlichen Messung \mathbf{z}_k wird das sogenannte Residuum $\tilde{\mathbf{y}}_k$ gebildet, welches mit der Kalman-Gain-Matrix \mathbf{K}_k gewichtet die A-priori-Schätzung korrigiert. In die Kalman-Gain-Matrix \mathbf{K}_k gehen dabei neben der A-priori-Systemkovarianzmatrix $\mathbf{P}_{k|k-1}$ und der Messmatrix \mathbf{H}_k die Kovarianzmatrix der Messunsicherheit \mathbf{R}_k ein.

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \quad (3.7)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k)^{-1} \quad (3.8)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \quad (3.9)$$

Für die Berechnung der A-posteriori-Systemkovarianzmatrix $\mathbf{P}_{k|k}$ gibt es verschiedene Formulierungen. Bei der Folgenden handelt es sich um die

Joseph-Form, welche durch ihre Symmetrien besonders stabil gegenüber numerischen Ungenauigkeiten ist [Gre14]:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^\top + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^\top, \quad (3.10)$$

mit \mathbf{I} als Einheitsmatrix.

Während die Prädiktion in jedem Zeitschritt genau einmal ausgeführt wird, können mehrere gleichzeitige Messungen in mehreren Korrekturschritten integriert werden. Es kann aber auch vorkommen, dass keine Messungen vorhanden sind und lediglich prädiziert wird, wobei dann natürlich die Unsicherheit immer weiter anwächst.

Abbildung 3.1 zeigt ein dynamisches bayessches Netz (DBN) eines Kalman-Filters, aus dem die eingangs erwähnte Markov-Annahme deutlich wird. Das dynamische bayessche Netz findet hier als graphisches Modell Verwendung, um die Zustände und deren Abhängigkeiten über die Zeit zu visualisieren [Rus09]. Die Zustände zum Zeitpunkt k sind mit \mathbf{x}_k in Hellblau dargestellt und die Messungen \mathbf{z}_k in Orange. Der wahre Zustand selbst ist nicht direkt beobachtbar, sondern kann nur anhand der Beobachtungen geschätzt werden. Da der wahre Zustand somit verborgen ist, wird es im Englischen als *hidden Markov model* (HMM) bezeichnet [Rus09].

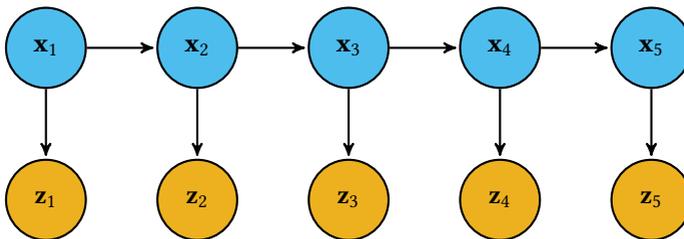


Abbildung 3.1: DBN eines Kalman-Filters.

3.1.2 Erweitertes Kalman-Filter

Das erweiterte Kalman-Filter (EKF) ist eine Variante für nichtlineare Systeme und basiert auf der Linearisierung der Filtergleichungen um den aktuell geschätzten Zustand [Sär13].

Im Prädiktionsschritt zum Zeitpunkt k wird die A-priori-Zustandsschätzung $\hat{\mathbf{x}}_{k|k-1}$ wie folgt berechnet:

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k), \quad (3.11)$$

wobei f eine nichtlineare Funktion ist, welche die Schätzung des prädizierten Zustands aus dem vorhergehenden Zustand und der Eingabe \mathbf{u}_k berechnet.

Die Berechnung der Systemkovarianzmatrix $\mathbf{P}_{k|k-1}$ erfolgt wie beim Kalman-Filter mit

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^\top + \mathbf{Q}_k. \quad (3.12)$$

Verglichen mit Gleichung (3.6) beim linearen Kalman-Filter hat sich die Berechnung nicht geändert, jedoch ist \mathbf{F}_k hier die Jacobi-Matrix von f bezüglich $\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k$:

$$\mathbf{F}_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k}. \quad (3.13)$$

Der Korrekturschritt integriert auch hier die Messung \mathbf{z}_k und berechnet die A-posteriori-Zustandsschätzung $\hat{\mathbf{x}}_{k|k}$.

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}) \quad (3.14)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k)^{-1} \quad (3.15)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \quad (3.16)$$

Bei der Berechnung des Residuums $\tilde{\mathbf{y}}_k$ ist nun eine nichtlineare Funktion h zur Ermittlung der prädizierten Messung auf Basis der A-priori-

Zustandsschätzung $\hat{\mathbf{x}}_{k|k-1}$ vorhanden und \mathbf{H}_k ist deren Jacobi-Matrix bezüglich $\hat{\mathbf{x}}_{k|k-1}$:

$$\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}}. \quad (3.17)$$

Die Berechnung der A-posteriori-Systemkovarianzmatrix $\mathbf{P}_{k|k}$ kann wieder identisch zum linearen Kalman-Filter erfolgen:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^\top + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^\top. \quad (3.18)$$

3.1.3 Stochastisches Klonen

Wenn mehrere relative Messungen vorhanden sind, ist die in Abschnitt 3.1.1 vorgestellte Formulierung des Standard-Kalman-Filters nicht mehr ausreichend. Während ein einzelner relativ messender Sensor in die Prädiktion eingehen kann, ist die Integration weiterer relativer Messungen nur über den Korrekturschritt möglich. Bei relativen Messungen hängt jedoch der aktuelle Zustand von einem vorhergehenden Zustand ab und verletzt damit die Markov-Annahme. Um eine zusätzliche relative Messungen formal korrekt in ein Kalman-Filter zu integrieren, stellten Roumeliotis et al. die Methode des sogenannten stochastischen Klonens (englisch: *stochastic cloning* – SC) mit einem Kalman-Filter (SC-KF) vor, welche den Zustand mit einem Klon des zu einer relativen Messung zugehörigen Zustands erweitert [Rou02].

Abbildung 3.2 zeigt ein DBN mit relativen Messungen $\mathbf{z}_{k,k+m}$ zwischen den zwei Zuständen zu den Zeitpunkten k und $k+m$ für $m=1$, welche in Hellgrün dargestellt sind. Darin wird deutlich, dass durch Abhängigkeiten von vorangegangenen Zuständen die Markov-Annahme verletzt wird.

Für eine relative Messung $\mathbf{z}_{k,k+m}$ zwischen den zwei Zuständen zu den Zeitpunkten k und $k+m$ wird die Zustandsschätzung um einen Klon von

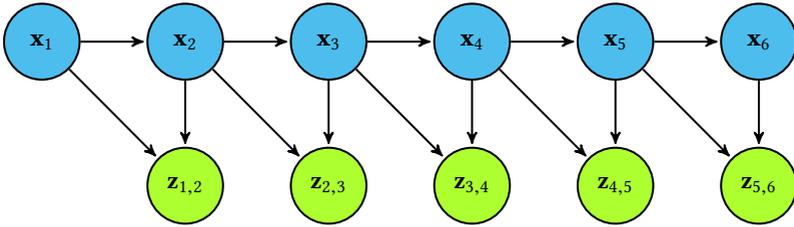


Abbildung 3.2: DBN mit relativen Messungen.

$\hat{\mathbf{x}}_{k|k}$ erweitert und man erhält die augmentierte Zustandsschätzung

$$\check{\mathbf{x}}_{k|k} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{x}}_{k|k} \end{bmatrix}. \quad (3.19)$$

Zum Zeitpunkt $k + m$ ergibt sich dann nach m Prädiktionen die augmentierte Zustandsschätzung

$$\check{\mathbf{x}}_{k+m|k} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{x}}_{k+m|k} \end{bmatrix}. \quad (3.20)$$

Weil $\hat{\mathbf{x}}_{k|k}$ ein Klon der zurückliegenden Zustandsschätzung ist, die durch die Prädiktion unverändert bleibt, wurde sie in [Rou02] als statisch bezeichnet. Im Gegensatz dazu wird $\hat{\mathbf{x}}_{k+m|k}$ im Filter durch die Prädiktion propagiert und wurde deshalb als sich entwickelnde Zustandsschätzung bezeichnet.

Die Systemkovarianzmatrix \mathbf{P} wird ebenfalls erweitert und hat zum Zeitpunkt k die Form

$$\check{\mathbf{P}}_{k|k} = \begin{bmatrix} \mathbf{P}_{k|k} & \mathbf{P}_{k|k} \\ \mathbf{P}_{k|k} & \mathbf{P}_{k|k} \end{bmatrix}. \quad (3.21)$$

Die linke obere Sub-Matrix der augmentierten Systemkovarianzmatrix $\check{\mathbf{P}}$ entspricht dabei dem statischen Teil der Zustandsschätzung und bleibt ebenfalls unverändert durch die Prädiktion:

$$\check{\mathbf{P}}_{k+1|k} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{k+1} \end{bmatrix} \check{\mathbf{P}}_{k|k} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{k+1}^\top \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{k+1} \end{bmatrix}. \quad (3.22)$$

Nach m Prädiktionsschritten wird die augmentierte Systemkovarianzmatrix $\check{\mathbf{P}}$ dann zu

$$\check{\mathbf{P}}_{k+m|k} = \begin{bmatrix} \mathbf{P}_{k|k} & \mathbf{P}_{k|k} \mathcal{F}_{k+m,k}^\top \\ \mathcal{F}_{k+m,k} \mathbf{P}_{k|k} & \mathbf{P}_{k+m|k} \end{bmatrix} \quad (3.23)$$

mit dem kumulativen Produkt der Zustandsübergangsmatrizen

$$\mathcal{F}_{k+m,k} = \mathbf{F}_{k+m} \mathbf{F}_{k+m-1} \dots \mathbf{F}_{k+2} \mathbf{F}_{k+1} = \prod_{i=1}^m \mathbf{F}_{k+i}. \quad (3.24)$$

Wird nach m Schritten eine relative Messung integriert, ist deren Residuum

$$\tilde{\mathbf{y}}_{k+m} = \mathbf{z}_{k,k+m} - \hat{\mathbf{z}}_{k,k+m}, \quad (3.25)$$

wobei $\hat{\mathbf{z}}_{k,k+m}$ die Differenz der prädierten Messungen $h(\hat{\mathbf{x}}_{k|k})$ und $h(\hat{\mathbf{x}}_{k+m|k})$ ist und $\mathbf{z}_{k,k+m}$ die relative Messung zwischen den Zuständen zu den Zeitpunkten k bzw. $k+m$. D. h., die Prädiktion ist im Vergleich zu der Messung mit einer m -fachen Rate verfügbar.

Die augmentierte Kalman-Gain-Matrix $\check{\mathbf{K}}_{k+m}$ kann analog zum Standard-Kalman-Filter berechnet werden:

$$\check{\mathbf{K}}_{k+m} = \check{\mathbf{P}}_{k+m|k} \check{\mathbf{H}}_{k+m}^\top \check{\mathbf{S}}_{k+m}^{-1}, \quad (3.26)$$

mit

$$\check{\mathbf{S}}_{k+m} = \check{\mathbf{H}}_{k+m} \check{\mathbf{P}}_{k+m|k} \check{\mathbf{H}}_{k+m}^\top + \mathbf{R}_{k,k+m}, \quad (3.27)$$

wobei $\mathbf{R}_{k,k+m}$ die Messfehlerkovarianzmatrix der relativen Messung ist. Die augmentierte Messmatrix

$$\check{\mathbf{H}}_{k+m} = [\mathbf{H}_k \quad \mathbf{H}_{k+m}] \quad (3.28)$$

beinhaltet beide Jacobi-Matrizen, welche bezüglich der jeweiligen Zustandsschätzungen $\hat{\mathbf{x}}_{k|k}$ bzw. $\hat{\mathbf{x}}_{k+m|k}$ gebildet werden.

Die resultierende augmentierte Kalman-Gain-Matrix $\check{\mathbf{K}}_{k+m}$ (3.26) ist ebenfalls eine Zusammensetzung aus den Kalman-Gain-Matrizen für die statische und die sich entwickelnde Zustandsschätzung:

$$\check{\mathbf{K}}_{k+m} = \begin{bmatrix} \mathbf{K}_k \\ \mathbf{K}_{k+m} \end{bmatrix}. \quad (3.29)$$

In [Rou02] wurde weiterhin argumentiert, dass die relative Messung keine neue Information bezüglich des geklonten Zustands \mathbf{x}_k liefert, sodass \mathbf{K}_k von Gleichung (3.29) $\mathbf{0}$ ist und

$$\check{\mathbf{K}}_{k+m} = \begin{bmatrix} \mathbf{0} \\ \mathbf{K}_{k+m} \end{bmatrix} \quad (3.30)$$

gilt.

Folglich wird die aktuelle Schätzung des Zustands und die Systemkovarianzmatrix nur mit der unteren Sub-Matrix \mathbf{K}_{k+m} von Gleichung (3.29) berechnet:

$$\hat{\mathbf{x}}_{k+m|k+m} = \hat{\mathbf{x}}_{k+m|k} + \mathbf{K}_{k+m} \check{\mathbf{y}}_{k+m}, \quad (3.31)$$

$$\mathbf{P}_{k+m|k+m} = \mathbf{P}_{k+m|k} - \mathbf{K}_{k+m} \check{\mathbf{S}}_{k+m} \mathbf{K}_{k+m}^\top. \quad (3.32)$$

Es wird deutlich, dass für eine einzelne zusätzliche Korrektur mittels stochastischem Klonen die augmentierte Systemkovarianzmatrix $\check{\mathbf{P}}$ nicht explizit durch alle Filterschritte propagiert werden muss, sondern zum Zeitpunkt $k + m$ rekonstruiert werden kann. Hierfür müssen lediglich die

vorangegangene Systemkovarianzmatrix $\mathbf{P}_{k|k}$ und alle seither angewendeten Zustandsübergangsmatrizen $\mathbf{F}_{k+1}\dots\mathbf{F}_{k+m}$ in den Gleichungen (3.23) und (3.24) verrechnet werden. Zusammen mit der geklonten Zustandsschätzung $\hat{\mathbf{x}}_{k|k}$, welche zur Berechnung der prädizierten Messung $\hat{\mathbf{z}}_{k,k+m}$ notwendig ist, können alle nachfolgenden Gleichungen (3.25) bis (3.32) berechnet werden, um die A-posteriori-Zustandsschätzung $\hat{\mathbf{x}}_{k+m|k+m}$ und deren Systemkovarianzmatrix $\mathbf{P}_{k+m|k+m}$ zu erhalten.

Auch in [Mou07] wurden zwei relative Sensoren fusioniert, aber die Integration weiterer Sensoren wurde als mögliche Erweiterung diskutiert. Die Vereinfachung der Berechnung der augmentierten Systemkovarianzmatrix $\check{\mathbf{P}}_{k+m|k}$ mittels der Gleichungen (3.23) und (3.24) ist demnach nicht mehr gültig. Stattdessen muss die Zustandsschätzung tatsächlich geklont und die augmentierte Systemkovarianzmatrix $\check{\mathbf{P}}$ explizit durch alle Filterschritte propagiert werden. Hierzu konnte jedoch keine weiterführende Literatur ermittelt werden.

Es existieren lediglich – soweit bekannt – Veröffentlichungen zu Methoden, bei denen bei der Integration weiterer Sensoren nur ein Teil der Zustandsschätzung geklont wird, wobei die teil-augmentierte Zustandsschätzung und Systemkovarianzmatrix in alle Filterberechnungen eingehen [Chi11; Kle11; Ste12]. Die partiell augmentierte Zustandsschätzung zum Zeitpunkt $k + m$ ergibt sich dann zu

$$\check{\mathbf{x}}_{k+m|k} = \begin{bmatrix} \hat{\mathbf{x}}_{c,k|k} \\ \hat{\mathbf{x}}_{k+m|k} \end{bmatrix}, \quad (3.33)$$

wobei $\hat{\mathbf{x}}_{c,k|k}$ der Teil des Zustandsraums ist, welcher von der relativen Messung betroffen ist und geklont wird, was mit der Kennzeichnung mit c verdeutlicht wird. Die partiell augmentierte Systemkovarianzmatrix hat zum Zeitpunkt k die Form

$$\check{\mathbf{P}}_{k|k} = \begin{bmatrix} \mathbf{P}_{cc,k|k} & \mathbf{P}_{ac,k|k}^\top \\ \mathbf{P}_{ac,k|k} & \mathbf{P}_{k|k} \end{bmatrix}, \quad (3.34)$$

wobei es sich bei $\mathbf{P}_{ac,k|k}$ um eine $\dim(\hat{\mathbf{x}}) \times \dim(\hat{\mathbf{x}}_c)$ -Matrix handelt und a für den Teil des aktuellen Zustands steht. Bei $\dim(\hat{\mathbf{x}})$ handelt es sich um die Dimension des Zustandsraums und bei $\dim(\hat{\mathbf{x}}_c)$ um die Dimension des geklonten Teils des Zustandsraums.

In Abschnitt 5.2 wird eine Realisierung der von [Mou07] vorgeschlagenen Erweiterung auf mehrere Sensoren vorgestellt und eine neuartige alternative Methode präsentiert, bei der dennoch eine Rekonstruktion der augmentierten Systemkovarianzmatrix ermöglicht wird.

3.2 Simultane Lokalisierung und Kartenerstellung

Bei der simultanen Lokalisierung und Kartenerstellung sind die Unsicherheit in der Lokalisierung und die Fehler in der Karte voneinander abhängig. Das SLAM-Problem und seine Abhängigkeiten können als Zufallsvariablen modelliert und wie die Zustandsschätzung des Kalman-Filters als dynamisches bayessches Netz (DBN) visualisiert werden, siehe Abbildung 3.3 [Thr05]. Die Posen des Roboters \mathbf{x}_k sind in Hellblau dargestellt, wobei k die Zeitschritte sind. Die Messungen eines Lokalisierungssensors sind in Rot als Eingabe \mathbf{u}_k dargestellt. Die Messungen eines perzeptorischen Sensors \mathbf{z}_k sind in Orange dargestellt und die mit diesem beobachteten Landmarken λ_l der Umgebung in Grün. Die Datenzuordnung \mathbf{n}_k repräsentiert die Zuordnung der Beobachtungen zu den bereits kartierten Landmarken und ist eine sehr wichtige Voraussetzung, da erst die wiederholte Beobachtung von Landmarken und deren Korrektur in der Karte die Genauigkeit der Karte und damit auch der Lokalisierung verbessert. Falsch zugeordnete Landmarken können hingegen zu Inkonsistenzen oder sogar zu einem Scheitern der Kartierung führen.

Die Lösung des SLAM-Problems ist dann die Schätzung des vollen SLAM-Posteriors

$$p(\mathbf{x}^k, \Lambda | \mathbf{z}^k, \mathbf{u}^k, \mathbf{n}^k), \quad (3.35)$$

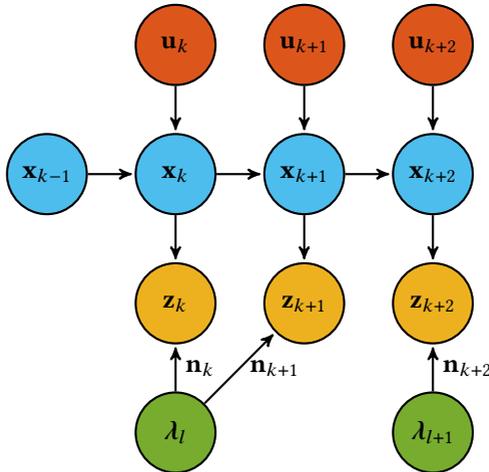


Abbildung 3.3: Das SLAM-Problem als DBN (nach [Thr05]).

wobei das hochgestellte k bedeutet, dass hiermit die komplette Sequenz bis zum Zeitpunkt k eingeschlossen ist, und $\mathbf{\Lambda}$ als Karte alle Landmarken λ enthält.

Damit die Schätzung des SLAM-Problems zu einer besseren Lösung konvergiert, sind bei der wiederholten Beobachtung bereits besuchter Kartenteile mehrere Kriterien ausschlaggebend. Das erste Kriterium ist die Einbeziehung der Abhängigkeiten zwischen den Posen des mobilen Roboters und den Landmarken sowie der Abhängigkeiten der Landmarken untereinander. Das zweite Kriterium ist eine robuste Datenzuordnung, da Fehlzugeordnungen zu ungenauen oder gar inkonsistenten Karten führen können. Als drittes Kriterium kann man die Erkennung oder Behandlung von dynamischen Objekten nennen, da diese nicht dauerhaft mit in die Karte aufgenommen und nicht für die Lokalisierung verwendet werden sollten.

Das SLAM-Problem kann auch als Markov-Kette interpretiert und mit einem Bayes-Filter als rekursive Schätzung mit inkrementeller Kartierung gelöst werden. Der rekursive Korrekturschritt ist jedoch nicht geschlos-

sen lösbar, sodass approximative Verfahren zum Einsatz kommen. Wenn beispielsweise die Variablen, d. h. das Bewegungsmodell, das Messmodell und der SLAM-Posterior, als multivariate Gaußdichten modelliert werden, kann ein EKF zur Schätzung zum Einsatz kommen. Als wesentlich flexibler, was die Modellierung der Dichten und die Verfolgung mehrerer Datenzuordnungshypothesen angeht, haben sich Partikelfilter erwiesen.

3.2.1 Partikelfilter

Im Gegensatz zu Ansätzen mit Kalman-Filtern, welche eine parametrische Approximation vornehmen, wird beim Partikelfilter die Wahrscheinlichkeitsdichte mit Samples des Zustandsraums repräsentiert. Durch die Samples können multi-modale Dichten modelliert und implizit mehrere Hypothesen für die Datenzuordnung verfolgt werden.

Ein Partikelfilter ist jedoch nur für Probleme mit wenigen Dimensionen geeignet, da die benötigte Anzahl von Partikeln im ungünstigsten Fall exponentiell mit der Dimension des Zustandsraum wachsen kann. Da die Dimension des Zustandsraums bei SLAM mit der Anzahl der Landmarken anwächst, wäre die einfache Anwendung eines Partikelfilters ungeeignet. Betrachtet man das bayessche Netz in Abbildung 3.3, so wird deutlich, dass unter der Bedingung eines bekannten Pfades die Landmarken voneinander unabhängig werden und in einzelne Schätzer faktorisiert werden können. Es ist daher möglich, das Partikelfilter durch Rao-Blackwellisierung geeignet umzuwandeln, indem anstatt nur der fortlaufenden Pose der gesamte Pfad geschätzt wird und jedes Partikel eine Pfadhypothese verfolgt [Mur99].

Das resultierende Filter ist ein rao-blackwellisiertes Partikelfilter (RBPF) und eine der bekanntesten Implementierungen ist der in [Mon02] vorgestellte FastSLAM-Algorithmus mit folgender Faktorisierung:

$$p(\mathbf{x}^k, \Lambda | \mathbf{z}^k, \mathbf{u}^k, \mathbf{n}^k) = p(\mathbf{x}^k | \mathbf{z}^k, \mathbf{u}^k, \mathbf{n}^k) \prod_{l=1}^L p(\lambda_l | \mathbf{x}^k, \mathbf{z}^k, \mathbf{u}^k, \mathbf{n}^k), \quad (3.36)$$

wobei nur noch die A-posteriori-Verteilung des Pfades durch die Partikel repräsentiert wird:

$$p(\mathbf{x}^k | \mathbf{z}^k, \mathbf{u}^k, \mathbf{n}^k). \quad (3.37)$$

Jedes Partikel führt dann L unabhängige Landmarkenschätzer mit sich, d. h., auch jedes Partikel besitzt eine eigene Kartenhypothese. Unter der Annahme, dass die Beobachtungen gaußverteilt sind, können die Landmarkenschätzer als individuelle EKFs realisiert werden. Werden mehrere Landmarken gleichzeitig beobachtet, können diese nacheinander aktualisiert werden, da sie voneinander bedingt unabhängig sind.

Es ist nicht möglich, die Partikel direkt aus der eigentlichen A-posteriori-Wahrscheinlichkeitsverteilung nach Gleichung (3.37) des Pfades zu ziehen, welche in der englischen Literatur als *target distribution* bezeichnet wird und sich mit Zielverteilung übersetzen lässt. Die rekursive Filterung kann jedoch, wie bereits angesprochen, in zwei Phasen aufgeteilt werden. Bei der in der englischen Literatur als *sequential importance sampling* (SIS) bezeichneten Methode wird in der ersten Phase jedes Partikel von einer als *proposal distribution* bezeichneten Verteilung gezogen [Che03]:

$$p(\mathbf{x}^{k,[i]} | \mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{n}^{k-1,[i]}) = p(\mathbf{x}_k^{[i]} | \mathbf{u}_k, \mathbf{x}_{k-1}^{[i]}) \cdot p(\mathbf{x}^{k-1,[i]} | \mathbf{z}^{k-1}, \mathbf{u}^{k-1}, \mathbf{n}^{k-1,[i]}), \quad (3.38)$$

welche aus der Propagation der Wahrscheinlichkeitsverteilung aus dem vorhergehenden Zeitschritt $p(\mathbf{x}^{k-1,[i]} | \mathbf{z}^{k-1}, \mathbf{u}^{k-1}, \mathbf{n}^{k-1,[i]})$ mit dem probabilistischen Bewegungsmodell $p(\mathbf{x}_k^{[i]} | \mathbf{u}_k, \mathbf{x}_{k-1}^{[i]})$ gebildet wird. Dabei wird eine zum i -ten Partikel zugehörige Variable mit hochgestelltem $[i]$ gekennzeichnet.

In der zweiten Phase wird jedes Partikel mit einem sogenannten *importance weight* $w_k^{[i]}$ gewichtet. Dieses Gewicht entspricht dem Quotienten aus Zielverteilung (*target distribution*) und vorgeschlagener Verteilung (*proposal distribution*). Folglich wird der SLAM-Posterior durch die gewichteten Partikel repräsentiert und die Gewichte $w_k^{[i]}$ haben die folgende

Form [Mon07]:

$$w_k^{[i]} = \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{p(\mathbf{x}^{k,[i]} | \mathbf{z}^k, \mathbf{u}^k, \mathbf{n}^{k,[i]})}{p(\mathbf{x}^{k,[i]} | \mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{n}^{k-1,[i]})} \propto p(\mathbf{z}_k | \mathbf{x}^{k,[i]}, \mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{n}^{k,[i]}), \quad (3.39)$$

d. h., das Gewicht ist proportional zum Likelihood der Beobachtung. Da es sich bei dem Schätzer für die Landmarken um einen EKF handelt, kann der Likelihood der Beobachtung und damit das Gewicht aus der Innovation, d. h. der Differenz der Beobachtung \mathbf{z}_k und der zugeordneten prädierten Beobachtung $\hat{\mathbf{z}}_{n_k^{[i]},k}$, berechnet werden und hat für Punktlandmarken die folgende Form:

$$w_k^{[i]} \propto \frac{1}{\sqrt{2\pi|\mathbf{Z}_k|}} \exp\left(-\frac{1}{2}\left(\mathbf{z}_k - \hat{\mathbf{z}}_{n_k^{[i]},k}\right)^\top \mathbf{Z}_k^{-1}\left(\mathbf{z}_k - \hat{\mathbf{z}}_{n_k^{[i]},k}\right)\right), \quad (3.40)$$

wobei \mathbf{Z}_k die Innovationskovarianzmatrix ist:

$$\mathbf{Z}_k = \mathbf{G}_k \mathbf{L}_{n_k^{[i]},k-1} \mathbf{G}_k^\top + \mathbf{R}_k. \quad (3.41)$$

Dabei ist \mathbf{G}_k das linearisierte Messmodell, $\mathbf{L}_{n_k^{[i]},k-1}$ die Kovarianz der zugeordneten Landmarke und \mathbf{R}_k die Messunsicherheit [Mon07].

Wenn der rekursive Algorithmus lange Zeit ausgeführt wird, können die Gewichte der Partikel degenerieren, d. h., die meisten Gewichte $w_k^{[i]}$ werden sehr klein. Um dieser Degeneration entgegen zu wirken, wurde ein Algorithmus namens *sequential importance resampling* (SIR) entwickelt [Rub88]. Dabei werden nach der Gewichtung die Partikel aus den aktuellen Partikeln proportional zu $w_k^{[i]}$ gezogen mit Zurücklegen. Danach werden alle Gewichte auf $w_k^{[i]} = 1/I$ gesetzt, wobei I die Anzahl der Partikel ist. Dies bedeutet, dass anschließend alle Partikel dasselbe Gewicht haben. Durch dieses sogenannte Resampling werden Partikel mit geringem Gewicht, d. h. mit einer niedrigen Wahrscheinlichkeit, aussortiert,

während Partikel mit hohem Gewicht und damit auch hoher Wahrscheinlichkeit mehrfach gezogen werden.

Das Resampling kann zwar effektiv die Degeneration verhindern, kann jedoch auf der anderen Seite zu einer Verminderung der Diversität führen, d. h., dass eine ausreichende Abdeckung des Zustandsraums nicht mehr gegeben ist. In [Dou00b] wurde deshalb eine adaptive Resampling-Strategie eingeführt, welche das Resampling nur ausführt, wenn die Partikel die Zielverteilung nicht mehr gut approximieren, was der Fall ist, wenn die Varianz der Gewichte aller Partikel zu groß wird. Ein Maß für die Qualität der Approximation kann demnach durch die effektive Sampleanzahl geschätzt werden:

$$\hat{I}_{\text{eff}} = \frac{1}{\sum_{i=1}^I \left(w_k^{[i]}\right)^2}. \quad (3.42)$$

Nur wenn \hat{I}_{eff} unter einen gewissen Schwellwert fällt, wird das Resampling ausgeführt.

Eine Situation, in der meist sehr viele Partikel beim Resampling eliminiert werden, ist das Schließen von Schleifen. Durch die Verwendung eines relativen Bewegungssensors wie Rad-Odometrie und die Kartierung neuer Gebiete entlang des Pfades in unbekannter Umgebung akkumulieren sich die Fehler. Wenn der mobile Roboter eine Schleife schließt, kehrt er in bekanntes Gebiet zurück und die Unsicherheit in der Pose verringert sich. Dadurch erhalten einige Partikel ein sehr hohes Gewicht, während viele Partikel ein niedriges Gewicht erhalten und im Resampling mit hoher Wahrscheinlichkeit aussortiert werden. D. h., ein Aspekt im Zusammenhang mit dem Schließen von Schleifen ist die nötige Anzahl von Partikeln, um deren Diversität aufrecht zu erhalten. Je größer die Schleife ist, desto mehr Partikel werden benötigt, um den Zustandsraum adäquat abzudecken [Mon07].

Verbesserung der vorgeschlagenen Verteilung

Partikelfilter liefern die besten Ergebnisse, wenn die vorgeschlagene Verteilung und die Zielverteilung ähnlich sind. Aus diesem Grund kann es nachteilig sein, die Partikel nur von dem Bewegungsmodell zu ziehen, falls die Bewegungssensoren des mobilen Roboters sehr viel ungenauer sind als der perzeptorische Umgebungssensor, da in diesem Fall viele Partikel ein geringes Gewicht erhalten und beim Resampling verworfen werden. Um eine vorgeschlagene Verteilung zu erhalten, welche näher an der Zielverteilung liegt, sollten die Beobachtungen aus dem perzeptorischen Umgebungssensor mit eingehen [Mon07].

Dies wurde mit FastSLAM 2.0 umgesetzt, bei dem die Partikel $\mathbf{x}_k^{[i]}$ dann eine Pose proportional zu einer neuen vorgeschlagenen Verteilung $p(\mathbf{x}_k^{[i]} | \mathbf{x}^{k-1, [i]}, \mathbf{z}^k, \mathbf{u}^k, \mathbf{n}^{k, [i]})$ ziehen, welche nun auch die aktuelle Beobachtung \mathbf{z}_k beinhaltet. Die neue vorgeschlagene Verteilung wird mittels eines EKF für jedes Partikel realisiert, wobei die Prädiktion mit dem Bewegungsmodell durchgeführt und die Messung \mathbf{z}_k mit dem Korrekturschritt integriert wird.

Nach [Dur06a] lassen sich die Gewichte dann mit

$$w_k^{[i]} = w_{k-1}^{[i]} \frac{p(\mathbf{x}_k^{[i]} | \mathbf{x}_{k-1}^{[i]}, \mathbf{u}_k) p(\mathbf{z}_k | \mathbf{x}_k^{[i]}, \mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{n}^{k, [i]})}{p(\mathbf{x}_k^{[i]} | \mathbf{x}^{k-1, [i]}, \mathbf{z}^k, \mathbf{u}^k, \mathbf{n}^{k, [i]})} \quad (3.43)$$

berechnen. Dabei lässt sich der Nenner aus der verbesserten vorgeschlagenen Verteilung berechnen. $p(\mathbf{x}_k^{[i]} | \mathbf{x}_{k-1}^{[i]}, \mathbf{u}_k)$ ergibt sich aus dem Bewegungsmodell und der Likelihood der Beobachtung $p(\mathbf{z}_k | \mathbf{x}_k^{[i]}, \mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{n}^{k, [i]})$ berechnet sich analog zu Gleichung (3.40).

Betrachtet man die Laufzeit des RBPF-SLAM, so steigt diese bei der Berechnung der vorgeschlagenen Verteilung, der Berechnung der Gewichte und der Aktualisierung der Karte lediglich linear mit der Anzahl der Partikel: $\mathcal{O}(I)$. Außerdem sind diese Teile des Algorithmus einfach parallel zu berechnen, da die Partikel voneinander unabhängig sind. Die

Laufzeit des Tests, ob Resampling notwendig ist, steigt ebenfalls linear mit der Anzahl der Partikel, der Test lässt sich jedoch nicht parallel berechnen. Das Resampling selbst hat im schlechtesten Fall eine Laufzeit mit $\mathcal{O}(L \cdot I)$, d. h., sie steigt mit der Anzahl der Partikel mal die Kartengröße, die Ausführung ist jedoch bei adaptivem Resampling sehr selten notwendig. Die Berechnung der Lokalisierung in der Karte lässt sich wieder in einfacher Weise parallelisieren, hängt jedoch von der Kartenart ab.

3.2.2 Landmarkenkarte

Eine Landmarkenkarte besteht aus einzelnen markanten Objekten, welche durch ihre beobachtbaren Merkmale beschrieben sind. Für die präzise und korrekte Kartierung mit SLAM ist eine robuste Datenzuordnung zwischen den Beobachtungen und den bereits kartierten Landmarken unerlässlich, weshalb sie auch als zweites Konvergenzkriterium geführt wird, vgl. Abschnitt 3.2.

Wie bereits erwähnt, repräsentiert ein Partikelfilter mehrere Hypothesen der Datenzuordnung, da diese pro Partikel durchgeführt wird. Partikel mit falschen Datenzuordnungen werden geringere Gewichte erhalten und werden mit höherer Wahrscheinlichkeit in einem späteren Resampling-Schritt verworfen, was als verzögerte Entscheidungsfindung betrachtet werden kann. Obwohl dies ein großer Vorteil ist, können viele falsche Zuordnungen einen negativen Einfluss haben, da das Verwerfen vieler Partikel die Diversität verringert und damit indirekt das erste Konvergenzkriterium beeinträchtigt [Mon07].

Werden die Merkmale der Landmarke als gaußverteilt modelliert, kann die Wahrscheinlichkeit der Beobachtung einer bereits in der Karte vorhandenen Landmarke als Funktion der Innovation pro Partikel berechnet und damit am Beispiel einer Punktlandmarke zur Datenzuordnung verwendet

werden:

$$\hat{n}_k^{[i]} = \arg \max_{n_k^{[i]}} p(\mathbf{z}_k | \mathbf{x}^{k,[i]}, \mathbf{z}^{k-1}, \mathbf{u}^k, n_k^{[i]}, \mathbf{n}^{k-1,[i]}) \quad (3.44)$$

$$= \arg \max_{n_k^{[i]}} \frac{1}{\sqrt{2\pi} |\mathbf{Z}_k|} \exp\left(-\frac{1}{2} (\mathbf{z}_k - \hat{\mathbf{z}}_{n_k^{[i]},k})^\top \mathbf{Z}_k^{-1} (\mathbf{z}_k - \hat{\mathbf{z}}_{n_k^{[i]},k})\right), \quad (3.45)$$

mit der Innovationskovarianzmatrix \mathbf{Z}_k entsprechend Gleichung (3.41). Dies ist eine Instanz des Maximum-Likelihood-Schätzers (ML-Schätzer) für multivariate Gaußdichten [Kro04]. Die ML-Datenzuordnung hat eine Laufzeit von schlechtestenfalls $\mathcal{O}(L \cdot I)$, wenn alle Landmarken überprüft werden. Wenn der Likelihood für alle möglichen Landmarkenkandidaten unter einem gewissen Schwellwert liegt, wird eine neue Landmarke in der Karte instanziiert. Ist dieser Schwellwert zu hoch gewählt, werden Landmarken fälschlicherweise mehrfach in die Karte eingetragen. Ist er zu niedrig gewählt, werden neu beobachtete Landmarken fälschlicherweise bereits bestehenden zugeordnet.

In den meisten Veröffentlichungen über SLAM mit Landmarkenkarten werden die Landmarken als Punktlandmarken modelliert, d. h., dass sie lediglich über ihren Ort definiert sind. In Abschnitt 4.1 wird zur Behandlung dieser Problematik ein erweitertes Landmarkenmodell für eine robustere Datenzuordnung vorgestellt.

3.2.3 2D-Rasterkarte

Eine 2D-Rasterkarte mit Belegtheitsinformation (englisch: *occupancy grid map*) kann sowohl Informationen über den belegten Raum als auch über freie bzw. befahrbare Bereiche repräsentieren. Jede Zelle der Karte speichert dafür ihre Belegtheitswahrscheinlichkeit. Diese geht von 0, was sicher frei bedeutet, über unbekannt bei 0,5 bis zu 1, was sicher belegt bedeutet [Thr05]. Die 2D-Rasterkarte ist eine dichte Umgebungsreprä-

sensation ohne Annahmen über die Struktur der Umgebung, d. h., sie ist sowohl für strukturierte als auch unstrukturierte Umgebungen geeignet.

Bei der Umgebungsrepräsentation in Form einer 2D-Rasterkarte wird das Problem der Kartenschätzung in separate Schätzer je Zelle aufgeteilt, d. h., die A-posteriori-Schätzung über die Karte wird in das Produkt der einzelnen Schätzer faktorisiert:

$$p(\Lambda | \mathbf{z}^k, \mathbf{x}^k) = \prod_{l=1}^L p(\lambda_l | \mathbf{z}^k, \mathbf{x}^k), \quad (3.46)$$

mit Λ als Karte und λ_l als l -te Zelle. Durch diese Aufteilung kann die Belegheitswahrscheinlichkeit sehr effizient geschätzt werden. Abhängigkeiten zwischen benachbarten Zellen können jedoch nicht modelliert werden.

Für die Berechnung der Belegheitswahrscheinlichkeiten der Zellen wird die sogenannte Logit-Form verwendet:

$$\mathcal{L}_{k,l} = \log \frac{p(\lambda_l | \mathbf{z}^k, \mathbf{x}^k)}{1 - p(\lambda_l | \mathbf{z}^k, \mathbf{x}^k)}. \quad (3.47)$$

Der Vorteil der Logit-Form ist, dass sie keine numerischen Instabilitäten wie die Wahrscheinlichkeitsdarstellung bei Multiplikationen nahe 0,0 und 1,0 aufweist. Die entsprechende Wahrscheinlichkeit kann aus der Logit-Form in einfacher Weise wieder rekonstruiert werden:

$$p(\lambda_l | \mathbf{z}^k, \mathbf{x}^k) = 1 - \frac{1}{1 + \exp(\mathcal{L}_{k,l})}. \quad (3.48)$$

Die Aktualisierung mit einer neuen Beobachtung wird rekursiv mittels Bayes-Filter durchgeführt:

$$\mathcal{L}_{k,l} = \mathcal{L}_{k-1,l} + \log \frac{p(\lambda_l | \mathbf{z}_k, \mathbf{x}_k)}{1 - p(\lambda_l | \mathbf{z}_k, \mathbf{x}_k)} - \log \frac{p(\lambda_l)}{1 - p(\lambda_l)}, \quad (3.49)$$

wobei $p(\lambda_l)$ die A-priori-Belegtheitswahrscheinlichkeit ist. Ein weiterer Vorteil der Logit-Form ist hier, dass die Multiplikation bei der Aktualisierung durch eine Addition ersetzt wird, was wesentlich effizienter zu berechnen ist [Thr05].

Die Lokalisierung in einer 2D-Rasterkarte kann über Scan-Matching mit einem 2D-LiDAR-Scan erfolgen. Dieses Scan-Matching kann beispielsweise mittels Bergsteigeralgorithmus (in der englischen Literatur als *hill-climbing* bekannt) durchgeführt werden [Sic08; Rus09]. Dabei wird die Zielfunktion aus den kleinsten Entfernungen der Strahlenendpunkte des 2D-LiDAR-Scans zur jeweils nächsten belegten Zelle berechnet [Thr05].

In [Gri07] wurde ein SLAM-Verfahren vorgestellt, das eine 2D-Rasterkartierung mit Belegtheitswahrscheinlichkeiten mit dem FastSLAM-2.0-Algorithmus anhand von 2D-LiDAR-Daten und Rad-Odometrie durchführt. Die Verwendung von FastSLAM 2.0 bedeutet, dass die Information des Umgebungssensors bereits in die vorgeschlagene Verteilung mit einbezogen wird. Sie ist im Falle eines 2D-LiDARs i. d. R. sehr viel genauer als die Bewegungsschätzung lediglich aus der Rad-Odometrie und die vorgeschlagene Verteilung wird somit auch hier ähnlicher zur Zielverteilung. Nach der Propagation der Partikel aus dem vorherigen Zeitschritt anhand der Odometriedaten wird zunächst für jedes Partikel ein Scan-Matching wie oben beschrieben durchgeführt. Anschließend wird für die Lokalisierung jeweils eine Normalverteilung $\mathcal{N}(\mathbf{x}_k^{[i]}, \mathbf{P}_k^{[i]})$ als vorgeschlagene Verteilung geschätzt, indem die Zielfunktion $p(\mathbf{z}_k | \Lambda_{k-1}^{[i]}, \mathbf{x}_j)$ zusammen mit dem Bewegungsmodell $p(\mathbf{x}_j | \mathbf{x}_{k-1}^{[i]}, \mathbf{u}_{k-1})$ in einem Intervall um die bei der Registrierung gefundenen Pose an J Stellen abgetastet wird:

$$\hat{\mathbf{x}}_k^{[i]} = \frac{1}{\kappa^{[i]}} \sum_{j=1}^J p(\mathbf{z}_k | \Lambda_{k-1}^{[i]}, \mathbf{x}_j) p(\mathbf{x}_j | \mathbf{x}_{k-1}^{[i]}, \mathbf{u}_{k-1}) \mathbf{x}_j, \quad (3.50)$$

$$\hat{\mathbf{P}}_k^{[i]} = \frac{1}{\kappa^{[i]}} \sum_{j=1}^J p(\mathbf{z}_k | \Lambda_{k-1}^{[i]}, \mathbf{x}_j) p(\mathbf{x}_j | \mathbf{x}_{k-1}^{[i]}, \mathbf{u}_{k-1}) (\mathbf{x}_j - \hat{\mathbf{x}}_k^{[i]})(\mathbf{x}_j - \hat{\mathbf{x}}_k^{[i]})^\top, \quad (3.51)$$

mit dem Normalisierungsfaktor

$$\kappa^{[i]} = \sum_{j=1}^J p(\mathbf{z}_k | \Lambda_{k-1}^{[i]}, \mathbf{x}_j) p(\mathbf{x}_j | \mathbf{x}_{k-1}^{[i]}, \mathbf{u}_{k-1}). \quad (3.52)$$

Die Gewichte der Partikel lassen sich dann wie folgt berechnen:

$$w_k^{[i]} \propto \sum_{j=1}^J p(\mathbf{z}_k | \Lambda_{k-1}^{[i]}, \mathbf{x}_j) p(\mathbf{x}_j | \mathbf{x}_{k-1}^{[i]}, \mathbf{u}_{k-1}) = \kappa^{[i]}. \quad (3.53)$$

In Abschnitt 4.2 wird ein alternativer Ansatz für die Berechnung der vorgeschlagenen Verteilung auf Basis eines EKF vorgestellt, welcher einen modularen Aufbau zur flexiblen Kombination mit anderen Kartenrepräsentationen und zusätzlichen Lokalisierungssensoren bietet. Mit diesem Ansatz wird in Abschnitt 4.3 ein hybrider SLAM-Algorithmus mit einer Kombination aus dichter Karte und Landmarkenkarte vorgestellt. In Abschnitt 5.1 erfolgt die Vorstellung der Integration eines zusätzlichen absoluten Lokalisierungssensors.

3.2.4 3D-NDT-Karte

Für die 3D-Kartierung ist eine speichereffiziente Datenstruktur wünschenswert, und es ist von Vorteil, wenn der Speicherbedarf nicht mit der Anzahl

der aufgenommenen Punkte bzw. Kartierungsdauer, sondern lediglich von der Kartengröße und deren Auflösung abhängt.

Eine 3D-NDT-Karte erlaubt eine kompakte Repräsentation der Umgebung und es müssen nicht alle Daten der 3D-Punktwolken gespeichert werden. Dennoch wird bei NDT-Karten im Vergleich zu reinen Belegtheitskarten eine Schätzung über die ursprüngliche Verteilung der Punkte und damit eine feiner aufgelöste Repräsentation der Umgebung bei gleicher Auflösung erzielt. D. h., dass eine NDT-Karte bei gleicher Auflösung expressiver ist als eine reine Belegtheitskarte und dabei robuster gegenüber verrauschten Sensordaten ist [Saa13].

Bei einer 3D-NDT-Karte wird die Umgebung in ein 3D-Raster aus Voxeln aufgeteilt und für jedes Voxel wird anhand der darin liegenden Punkten eine Normalverteilung geschätzt, d. h., die Punkte werden in Normalverteilungen transformiert. Die Normalverteilung des l -ten Voxels $\mathcal{N}(\boldsymbol{\pi}_l, \mathbf{\Pi}_l)$ wird für die M Punkte \mathbf{p} mit der korrigierten Stichprobenkovarianz folgendermaßen geschätzt [Mag09]:

$$\hat{\boldsymbol{\pi}}_l = \frac{1}{M} \sum_{m=1}^M \mathbf{p}_m, \quad (3.54)$$

$$\hat{\mathbf{\Pi}}_l = \frac{1}{M-1} \sum_{m=1}^M (\mathbf{p}_m - \hat{\boldsymbol{\pi}}_l)(\mathbf{p}_m - \hat{\boldsymbol{\pi}}_l)^\top. \quad (3.55)$$

Abbildung 3.4 zeigt eine Visualisierung unterschiedlicher 3D-Normalverteilungen als 3σ -Ellipsoide. In Abbildung 3.4a sind die Punkte gleichverteilt, d. h., alle Eigenwerte der entsprechenden Kovarianzmatrix sind zirka gleich groß: $\epsilon_x \approx \epsilon_y \approx \epsilon_z$. Abbildung 3.4b entspricht einer länglichen Struktur, d. h., ein Eigenwert ist sehr viel größer als die beiden anderen: $\epsilon_x \approx \epsilon_y \ll \epsilon_z$. Abbildung 3.4c entspricht einer flachen horizontalen Fläche, wie z. B. einer Straßenoberfläche, d. h., ein Eigenwert ist sehr viel kleiner als die beiden anderen: $\epsilon_x \approx \epsilon_y \gg \epsilon_z$.

Mit den Gleichungen (3.54) und (3.55) ist keine effiziente inkrementelle Aktualisierung möglich. Außerdem müssten alle Punkte gespeichert wer-

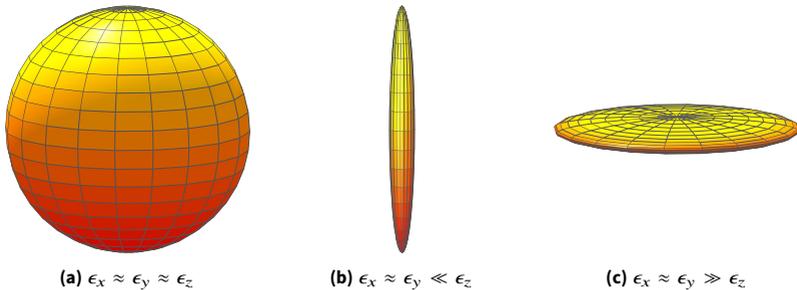


Abbildung 3.4: Beispiele für NDT-Ellipsoide.

den, was der Forderung nach einer speichereffizienten Datenstruktur entgegen steht. Daher wurde in [Tak06] eine Methode für eine inkrementelle Aktualisierung vorgestellt, welche allerdings im Gegensatz zu Gleichung (3.55) auf einer nicht-korrigierten Kovarianzschätzung beruht.

In dynamischen Umgebungen, d. h. in Umgebungen mit sich bewegenden Objekten, ist es von Vorteil, auch Information der Messungen über den freien Raum mit einzubeziehen, wie es auch bei der 2D-Belegtheitskarte der Fall ist. Dadurch können dynamische Objekte aus der Karte entfernt werden und zusätzlich auch durch Ungenauigkeiten fälschlicherweise kartierte Bereiche wieder als frei markiert werden. In [Saa13] wurde ein NDT-OM genanntes Verfahren vorgestellt, welches ebenfalls eine Belegtheitswahrscheinlichkeit mit der Logit-Form wie bei der im vorherigen Abschnitt vorgestellten 2D-Belegtheitskarte einbezieht. Hierbei wird ein heuristisches Verfahren eingesetzt, welches die Wahrscheinlichkeit jedes Voxels auf dem Weg des LiDAR-Strahls verringert, bis dieser an einem Objekt reflektiert wird, d. h. auf ein Hindernis trifft. In der Zelle, in der der Strahl auf ein Hindernis trifft, wird die Wahrscheinlichkeit für die Belegtheit erhöht. Ein wichtiger Unterschied zu einer reinen Belegtheitskarte ergibt sich daraus, dass bei der NDT-Karte in dem Voxel eine zusätzliche räumliche Struktur modelliert ist, wohingegen das Volumen des Voxels bei einer Belegtheitskarte als homogen angenommen wird. Da die räumliche Struktur nur einen

Teil des Voxels belegt, tragen durch das Voxel hindurchgehende Strahlen weniger negative Information, als dies bei einer reinen Belegtheitskarte der Fall wäre, und die Verringerung der Belegtheitswahrscheinlichkeit erfolgt entsprechend konservativer.

In Abschnitt 5.3.2 wird eine performante Implementierung mit inkrementeller Aktualisierung auf Basis einer korrigierten Kovarianzschätzung und einer zusätzlichen Existenzwahrscheinlichkeit für die Berücksichtigung negativer Information mittels einer effizienten Umsetzung eines Strahlverfolgungsalgorithmus vorgestellt.

Kapitel 4

Erweiterte SLAM-Verfahren

In diesem Kapitel werden Erweiterungen vorgestellt, welche eine robustere Kartierung erlauben. So kann durch die Modellierung von Landmarken mit zusätzlichen Merkmalen die Zuverlässigkeit der Datenzuordnung zwischen Beobachtungen und bereits kartierten Landmarken verbessert werden. Des Weiteren können durch die Kombination von dichten Karten mit landmarkenbasierten Karten die Robustheit und Präzision in heterogenen Umgebungen erhöht werden.

4.1 Kartierung mit erweitertem Landmarkenmodell

Die robuste Datenzuordnung beobachteter Landmarken zu bereits kartierten Landmarken ist eine wichtige Voraussetzung für die Erzeugung konsistenter und präziser Karten und das zweite Konvergenzkriterium für SLAM-Algorithmen, siehe Abschnitt 3.2. So können falsche Zuordnungen dazu führen, dass das SLAM-Verfahren divergiert und die erzeugte Karte ungenau wird. Werden sehr viele Beobachtungen falsch zugeordnet, kann dies das Erzeugen einer konsistenten Karte sogar unmöglich machen. Eine

Modellierung der Landmarken durch zusätzliche Merkmale, welche eine Unterscheidung distinktiver machen, soll eine zuverlässigere Zuordnung erlauben. Dies führt im Fall von ansonsten vereinzelt aufgetretenen Fehlzuidnungen zu einer präziseren Karte. Sind sehr viele Beobachtungen ohne die zusätzlichen Merkmale nicht robust zuzuordnen, so kann die Erweiterungen eine korrekte Kartierung überhaupt erst ermöglichen.

4.1.1 Datenzuordnung

In vielen EKF-basierten SLAM-Algorithmen und den ursprünglichen FastSLAM- und FastSLAM-2.0-Algorithmen sind Landmarken als 2D-Punktobjekte modelliert, d. h., sie sind lediglich über ihre zwei Koordinaten definiert. Dies macht die Datenzuordnung anfällig für Mehrdeutigkeiten in Regionen, in denen die Landmarken dicht beieinander liegen und vor allem beim Schließen von Schleifen, da hier die Bewegungsunsicherheit i. A. am größten ist. Letzteres ist der Fall, weil die Unsicherheit des mobilen Roboters bei der Fahrt durch unbekanntes Gebiet anwächst und erst nach dem Schließen der Schleife durch die Beobachtung bekannter Landmarken reduziert werden kann.

Werden Landmarken lediglich über ihre Position modelliert, gibt es zwei Ursachen, welche die Zuverlässigkeit beeinflussen. Diese sind die in Abbildung 4.1 dargestellte Bewegungsunsicherheit und die Messunsicherheit. Während die Messunsicherheit zu einzelnen Fehlzuidnungen führen kann, kommt es durch die Bewegungsunsicherheit möglicherweise zu mehreren Fehlzuidnungen gleichzeitig. Zudem ist die Bewegungsunsicherheit aufgrund der vergleichsweise ungenauen Rad-Odometrie meist größer als die Messunsicherheit, wie in Abschnitt 3.2.1 erläutert.

Da die Schwierigkeit der Unterscheidung vor allem damit zusammenhängt, dass die Position das einzige Merkmal ist, liegt es nahe, dem Modell der Landmarke weitere vom Ort unabhängige, also z. B. ansichtsbasierte, Merkmale hinzuzufügen, um ihre Unterscheidbarkeit zu verbessern. Dies sollte vor allem in Situationen, in denen die Landmarken dicht beisammen

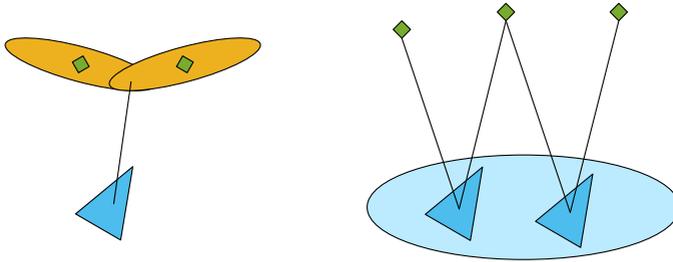


Abbildung 4.1: Mehrdeutigkeit durch Messunsicherheit links und Mehrdeutigkeit durch Lokalisierungsunsicherheit rechts, nach [Mon07]. ▲: Mobiler Roboter, ◆: Landmarke, ○: Messunsicherheit, ○: Lokalisierungsunsicherheit.

sind oder die Unsicherheit in der Pose des Roboters besonders groß ist, die Datenzuordnung verbessern [Emt10a].

Als zugrundeliegender SLAM-Algorithmus wurde ein rao-blackwellisiertes Partikelfilter (RBPF) gewählt, da dieses, wie in Abschnitt 2.2.1 dargestellt, gegenüber den auf dem Kalman-Filter basierenden Methoden einige Vorteile bietet. So können bei EKF-basierten Verfahren lediglich gaußverteilte Merkmale direkt integriert werden, während die bedingten Unabhängigkeiten beim RBPF eine flexiblere Gestaltung des Landmarkenmodells erlauben.

Da als SLAM-Verfahren ein Partikelfilter zum Einsatz kommt, sollen die zusätzlichen Merkmale nicht nur für die Zuordnung verwendet werden, sondern auch in die Berechnung der Gewichte der Partikel einfließen. Dies hat darüber hinaus den Vorteil, dass Partikel, welche dennoch falsch zugeordnete Beobachtungen integriert haben, niedrigere Gewichte besitzen und damit bei einem späteren Resampling wahrscheinlicher eliminiert werden.

4.1.2 Erweitertes Landmarkenmodell

Für eine robuste Datenzuordnung wurde ein erweitertes Landmarkenmodell entworfen, welches vertikale zylindrische Objekte, wie Baumstämme

und Laternenpfähle, abbildet. Zusätzlich zu der Position enthält es eine visuelle Signatur als ansichtsbasiertes Merkmal und den Radius der Landmarke. Durch den Radius wird die horizontale Ausdehnung einer Landmarke beschrieben. Idealerweise sind die Signatur und der Radius ortsunabhängig, sodass sie auch bei großer Lokalisierungsunsicherheit des mobilen Roboters eine zuverlässigere Zuordnung erlauben.

Das erweiterte Landmarkenmodell λ beschreibt ein vertikales zylindrisches Objekt mit dem Merkmalsvektor $\lambda = [x_\lambda, y_\lambda, r_\lambda, \nu]$, welcher sich aus geometrischen Merkmalen $\mathbf{m} = [x_\lambda, y_\lambda, r_\lambda]$ für (x -Koordinate, y -Koordinate und Radius) und einer visuellen Signatur ν zusammensetzt. Dabei entsprechen die x - und y -Komponenten in Kartenkoordinaten dem einfachen Punktlandmarkenmodell. Die beiden anderen Merkmale werden als unabhängig vom Ort angenommen und erweitern das einfache Punkt-Landmarkenmodell.

Zur Extraktion der Landmarken wird ein hierarchisches Extraktionsverfahren basierend auf 2D-LiDAR- und Kamerabilddaten angewendet. Die Kombination dieser beiden Sensoren hat den Vorteil, dass die hohe Entfernungsgenauigkeit des LiDARs mit der höheren Winkelauflösung der Kamera verbunden wird. Die beiden Sensoren werden aufeinander kalibriert, sodass die LiDAR-Punkte in das Kamerabild projiziert werden können, in Abbildung 4.2 als farbige Punkte dargestellt. Die weitaus höhere Auflösung der Kamera wird anhand des Vergleichs der Pixelgröße zu dem weiten horizontalen Abstand der Punkte des LiDAR-Scans deutlich [Ulr10].

Detektion

Zur Detektion der Landmarken wird eine hierarchische Strategie mit den Daten aus beiden Sensoren angewendet. Bei der Detektion werden zuerst die LiDAR-Daten betrachtet, da sich die Landmarken in dem Scan deutlich als lokale Entfernungsminima abzeichnen. In Abbildung 4.2 sind die Punkte der Landmarkenkandidaten durch größere farbige Quadrate gekennzeichnet. Da im Scan auch einige andere Objekte, wie z. B. Steine oder Büsche, als Kandidaten bei der Auswertung der LiDAR-Daten erkannt werden, wird im



Abbildung 4.2: Ein in das Kamerabild referenzierter Scan des 2D-LiDARs (farbige Punkte). Größer hervorgehobene Punkte zeigen mögliche Landmarkenkandidaten nach dem ersten Detektionsschritt. Das grüne Rechteck bezeichnet eine nach dem zweiten Schritt verifizierte Landmarke und den Ausschnitt, welcher zur Berechnung der visuellen Signatur verwendet wird [Ulr10].

Kamerabild zusätzlich nach vertikalen Kanten gesucht, um solche Fehldektionen auszuschließen bzw. schmale vertikale Strukturen zu verifizieren. Dafür werden vertikale Kanten links und rechts von dem Kandidaten in dem Kamerabild gesucht. Nur Kandidaten, welche zwei deutliche vertikale Kanten aufweisen, entsprechen dem Landmarkenmodell und werden als Detektion gewertet. In Abbildung 4.2 zeigt das grüne Rechteck eine final verifizierte Landmarkendetektion, wobei die horizontale Ausdehnung des Rechtecks durch die linke und die rechte aus dem Kamerabild extrahierte Kanten ermittelt wurde. Dieses dient als Segmentfläche für die Bestimmung der visuellen Signatur [Ulr10].

Merkmalsbestimmung

Nach der Detektion werden die Messdaten der Landmarke ermittelt. Der horizontale Richtungswinkel (Azimut) α_λ und die Entfernung zur Oberfläche der Landmarke d_L werden zunächst aus den Messungen des LiDARs bestimmt, da dessen Entfernungsmessungen eine hohe Genauigkeit aufweisen. Der Radius der Landmarke r_λ wird wegen der höheren Winkelauflösung aus den Bilddaten anhand der vom LiDAR ermittelten Entfernung d_L vom LiDAR bestimmt, welche auch der Entfernung von der Kamera entspricht:

$$r_\lambda = d_L \frac{\mathfrak{d}}{2c}, \quad (4.1)$$

wobei c für die Kamerakonstante der Kamera in horizontaler Richtung des Sensors steht und \mathfrak{d} für die bei der Detektion ermittelte Distanz zwischen der rechten und der linken Kante der Landmarke in Pixeln im Bild. Da d_L der Entfernung zur Oberfläche der Landmarke entspricht, ergibt sich durch Addition des Radius r_λ auf die Entfernung d_L die beobachtete Entfernung des Zentrums der Landmarke d_λ [Ulr10]. Zusammen mit dem Azimut α_λ und dem Radius r_λ ergeben sich die drei geometrischen Komponenten der Beobachtung der Landmarke:

$$\mathbf{z}_{\mathbf{m},k} = \begin{bmatrix} d_L + r_\lambda \\ \alpha_\lambda \\ r_\lambda \end{bmatrix} = \begin{bmatrix} d_\lambda \\ \alpha_\lambda \\ r_\lambda \end{bmatrix}. \quad (4.2)$$

Die visuelle Signatur der Landmarke ν besteht aus einer Schätzung der Farbverteilung, welche über die Segmentfläche innerhalb des detektierten Rechtecks aus den Bilddaten bestimmt wird. Die Farbinformationen werden über die jeweilige Segmentfläche berücksichtigt. Als Farbmodell wird das in [Com00] vorgeschlagene HSV-Farbmodell verwendet. HSV steht im Englischen für *hue* – Farbwert, *saturation* – Sättigung und *value* – Hellwert oder Helligkeit [Smi78]. Der HSV-Farbraum wurde gewählt, da dieser im Vergleich zum RGB-Farbraum die Farbinformation von der Helligkeit entkoppelt und robuster gegenüber Veränderungen in der Umgebungshelligkeit ist. Damit ist die visuelle Signatur unabhängiger von lokalen Abschattungen.

Für die Schätzung der Farbverteilung werden zunächst Histogramme über die Kanäle mit den Klassen γ_H , γ_S und γ_V gebildet. Dabei werden für γ_H und γ_S nur Pixel verwendet, deren Farbwert bzw. Sättigung über einem jeweiligen Schwellwert liegen, um hauptsächlich die Farbinformation zu berücksichtigen. Die übrigen Pixel werden dann für die Bildung von γ_V verwendet, da auch die nahezu farblosen Bereiche Informationen für die Wiedererkennung liefern [Pér02]. Jedes Histogramm beinhaltet B Klassen und bei der Berechnung der Farbverteilung werden die Klassen zu einem Gesamthistogramm $\gamma_\nu = \gamma_H \gamma_S + \gamma_V$ zusammengefasst. Der Vektor

ξ beinhaltet die Histogrammklassen der Segmentfläche mit U Pixeln und $\xi(u) \in 1, \dots, B$ bezeichnet die Histogrammkategorie an der Pixelstelle u , korrespondierend mit dem Pixel im HSV-Farbraum. Die geschätzte Dichte der beobachteten Farbverteilung wird dann nach [Pér02] über

$$\mathbf{z}_v(b) = \eta \sum_{u=1}^U \delta(\xi(u) - b) \quad (4.3)$$

für alle $b \in 1, \dots, B$ bestimmt. Dabei ist δ die Dirac-Funktion

$$\delta(v) = \begin{cases} 1, & v = 0 \\ 0, & \text{sonst} \end{cases} \quad (4.4)$$

und η ein Normalisierungsfaktor, sodass $\sum_{b=1}^B \mathbf{z}_v(b) = 1$ ist. Damit entspricht \mathbf{z}_v einer diskretisierten Dichte und kann als visuelle Signatur der Landmarke dienen [Ulr10; Emt11].

Wiedererkennung

Für die Wiedererkennung einer bereits in der Karte existierenden Landmarke und für die Berechnung der Gewichte der Partikel für den zugrundeliegenden Partikelfilter-SLAM-Algorithmus wird ein statistisches Distanzmaß benötigt. Hierfür bietet sich der häufig verwendete Likelihood an. Mit diesem kann sowohl mittels der Maximum-Likelihood-Methode die wahrscheinlichste Zuordnung einer Beobachtung zu einer bereits kartierten Landmarke ermittelt werden als auch nach der Zuordnung das jeweilige Gewicht berechnet werden. Die statistische Unabhängigkeit der visuellen Signatur \mathbf{v} von der Pose des Roboters ist dabei ein wichtiges Kriterium, denn nach dem Multiplikationssatz können individuell berechnete Likelihoods verknüpft werden, wenn es sich um voneinander unabhängige Merkmale handelt. Der Likelihood der Beobachtung der geometrischen Merkmale $\mathbf{z}_{m,k}$ ist demnach unabhängig von dem Likelihood der Beobachtung der visuellen Signatur $\mathbf{z}_{v,k}$. Daraus folgt, dass der Likelihood der

gesamten Beobachtung $\mathbf{z}_{\lambda,k}$ aufgeteilt werden kann:

$$\begin{aligned}
 & p\left(\mathbf{z}_{\lambda,k} \mid \mathbf{x}^{k,[i]}, \mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{n}^{k,[i]}\right) \\
 &= p\left(\mathbf{z}_{\mathbf{m},k} \mid \mathbf{x}^{k,[i]}, \mathbf{z}_{\mathbf{m}}^{k-1}, \mathbf{u}^k, \mathbf{n}^{k,[i]}\right) \cdot p\left(\mathbf{z}_{\mathbf{v},k} \mid \mathbf{z}_{\mathbf{v}}^{k-1}, \mathbf{n}^{k,[i]}\right) \\
 &= p_{\mathbf{m},k}^{[i]} \cdot p_{\mathbf{v},k}^{[i]}.
 \end{aligned} \tag{4.5}$$

D. h., der Gesamt-Likelihood wird durch die Multiplikation des Likelihood bezüglich der geometrischen Beobachtungen $\mathbf{z}_{\mathbf{m}}$ und dem Likelihood bezüglich der Beobachtung der visuellen Signatur $\mathbf{z}_{\mathbf{v}}$ berechnet. Während die geometrischen Beobachtungen als normalverteilt angenommen werden, wird die Beobachtung der visuellen Signatur, wie im vorherigen Abschnitt beschrieben, durch ein Histogramm modelliert. Die visuelle Signatur ist zudem von der Pose und der Bewegung des Roboters unabhängig.

Für die Berechnung des Likelihoods der Beobachtung der geometrischen Merkmale wird zunächst für das i -te Partikel die prädierte Beobachtung $\hat{\mathbf{z}}_{\mathbf{m},n_k^{[i]}}$ anhand der zugeordneten Landmarke und der geschätzten Roboterpose $\hat{\mathbf{x}}_{k|k-1}^{[i]} = [\hat{x}^{[i]} \hat{y}^{[i]} \hat{\psi}^{[i]}]^\top$ berechnet:

$$\hat{\mathbf{z}}_{\mathbf{m},n_k^{[i]}} = \begin{bmatrix} \hat{d} \\ \text{atan2}(\hat{d}_y, \hat{d}_x) - \hat{\psi}^{[i]} \\ \hat{r}_{\lambda,n_k^{[i]}} \end{bmatrix}, \tag{4.6}$$

mit

$$\hat{d}_x = \hat{x}_{\lambda,n_k^{[i]}} - \hat{x}^{[i]}, \tag{4.7}$$

$$\hat{d}_y = \hat{y}_{\lambda,n_k^{[i]}} - \hat{y}^{[i]}, \tag{4.8}$$

$$\hat{d} = \sqrt{\hat{d}_x^2 + \hat{d}_y^2}, \tag{4.9}$$

wobei atan2 eine Erweiterung des Arkustangens ist, welche alle vier Quadranten abdeckt.

Das zugehörige linearisierte Messmodell \mathbf{G}_k ist die Jacobi-Matrix bezüglich der Beobachtung:

$$\mathbf{G}_k = \begin{bmatrix} \hat{d}_x/\hat{d} & \hat{d}_y/\hat{d} & 0 \\ -\hat{d}_y/\hat{d}^2 & \hat{d}_x/\hat{d}^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.10)$$

Der Likelihood der Beobachtungen der geometrischen Merkmale wird anschließend mittels der Differenz zwischen der Beobachtung $\mathbf{z}_{\mathbf{m},k}$ nach Gleichung (4.2) und der prädizierten Beobachtung $\hat{\mathbf{z}}_{\mathbf{m},n_k^{[i]}}$ nach Gleichung (4.6) berechnet:

$$p_{\mathbf{m},k}^{[i]} = \frac{1}{(2\pi)^{3/2} \sqrt{|\mathbf{Z}_k|}} \exp\left(-\frac{1}{2} \left(\mathbf{z}_{\mathbf{m},k} - \hat{\mathbf{z}}_{\mathbf{m},n_k^{[i]}}\right)^\top \mathbf{Z}_k^{-1} \left(\mathbf{z}_{\mathbf{m},k} - \hat{\mathbf{z}}_{\mathbf{m},n_k^{[i]}}\right)\right), \quad (4.11)$$

mit der Innovationskovarianzmatrix

$$\mathbf{Z}_k = \mathbf{G}_k \mathbf{L}_{n_k^{[i]},k-1} \mathbf{G}_k^\top + \mathbf{R}_k, \quad (4.12)$$

welche sich aus dem linearisierten Messmodell \mathbf{G}_k nach Gleichung (4.10), der Kovarianzmatrix der in der Karte zugeordneten Landmarke $\mathbf{L}_{n_k^{[i]},k-1}$ und der Messfehlerkovarianz \mathbf{R}_k zusammensetzt [Mon07].

Der Likelihood des Histogrammvergleichs zwischen der Beobachtung $\mathbf{z}_{\mathbf{v},k}$ und der geschätzten visuellen Signatur der zugeordneten Landmarke $\hat{\mathbf{v}}_{n_k^{[i]}}$ kann wie folgt berechnet werden [Com00]:

$$p_{\mathbf{v},k}^{[i]} = \exp\left(-\zeta D_{\mathcal{Y}}(\mathbf{z}_{\mathbf{v},k}, \hat{\mathbf{v}}_{n_k^{[i]}})\right). \quad (4.13)$$

Dabei ist

$$D_\gamma(\mathbf{z}_{v,k}, \hat{\mathbf{v}}_{n_k^{[i]}}) = \sqrt{1 - \sum_{b=1}^B \sqrt{\mathbf{z}_{v,k}(b) \cdot \hat{\mathbf{v}}_{n_k^{[i]}(b)}}} \quad (4.14)$$

die Distanz zwischen den Histogrammen, wobei b die jeweilige Histogrammklasse bezeichnet und B die Gesamtzahl der Klassen. Diese Form der Histogrammdistanz D_γ wurde von [Com00] zum Histogrammvergleich herangezogen und beinhaltet den Bhattacharyya-Koeffizienten

$$\beta(\mathbf{z}_{v,k}, \hat{\mathbf{v}}_{n_k^{[i]}}) = \sum_{b=1}^B \sqrt{\mathbf{z}_{v,k}(b) \hat{\mathbf{v}}_{n_k^{[i]}(b)}}. \quad (4.15)$$

Es wurde weiterhin gezeigt, dass die Distanz D_γ einer Metrik entspricht, wohingegen die Bhattacharyya-Distanz $D_B = -\ln(\beta)$ selbst und die oft verwendete Kullback-Leibler-Divergenz mindestens eines der Axiome verletzen [Pér02]. So wird z. B. in [Cov91] angemerkt, dass die Kullback-Leibler-Divergenz zwar oft als Distanz zwischen zwei Wahrscheinlichkeitsdichten angegeben wird, sie jedoch die Dreiecksungleichung nicht erfüllt. Die Bhattacharyya-Distanz erfüllt ebenso nicht die Dreiecksungleichung [Bha09].

Die letztendliche Datenzuordnung wird pro Partikel mit dem Maximum-Likelihood-Schätzer durchgeführt:

$$\begin{aligned} \hat{n}_k^{[i]} &= \arg \max_{n_k^{[i]}} p(\mathbf{z}_{\lambda,k} | \mathbf{x}^{k-1,[i]}, \mathbf{z}^{k-1}, \mathbf{u}^k, n_k^{[i]}, \mathbf{n}^{k-1,[i]}) \\ &= \arg \max_{n_k^{[i]}} p_{\mathbf{m},k}^{[i]} \cdot p_{\mathbf{v},k}^{[i]}, \end{aligned} \quad (4.16)$$

mit $p_{\mathbf{m},k}^{[i]}$ für die geometrischen Merkmale nach Gleichung (4.11) und $p_{\mathbf{v},k}^{[i]}$ für die visuelle Signatur nach Gleichung (4.13). Somit kann der Gesamt-Likelihood für die Datenzuordnung einer Beobachtung zu einer bereits kartierten Landmarke direkt berechnet werden.

Verbesserung der vorgeschlagenen Verteilung

Bei dem Partikelfilter wird angelehnt an den FastSLAM-2.0-Algorithmus eine Verbesserung der vorgeschlagenen Verteilung durch die Beobachtungen bekannter und damit bereits kartierter Landmarken durchgeführt.

Zunächst wird aus der zugeordneten Landmarke und der geschätzten Roboterpose des i -ten Partikels die prädierte Beobachtung $\hat{\mathbf{z}}_{\mathbf{m},n_k}^{[i]}$ nach Gleichung (4.6) berechnet. Für die weitere Berechnung wird zusätzlich wieder die Innovationskovarianzmatrix \mathbf{Z}_k nach Gleichung (4.12) und außerdem die Jacobi-Matrix bezüglich der Pose des Roboters benötigt:

$$\mathbf{J}_k = \begin{bmatrix} -\hat{d}_x/\hat{d} & -\hat{d}_y/\hat{d} & 0 \\ \hat{d}_y/\hat{d}^2 & -\hat{d}_x/\hat{d}^2 & -1 \\ 0 & 0 & 0 \end{bmatrix}. \quad (4.17)$$

Daran, dass hier in der letzten Zeile alle Elemente 0 sind, ist zu erkennen, dass der Radius der Landmarke unabhängig von der Pose des Roboters angenommen wird.

Das Residuum $\tilde{\mathbf{y}}_k$ berechnet sich aus der Differenz der Messung und der prädierten Messung der in der Karte zugeordneten Landmarke:

$$\tilde{\mathbf{y}}_k = \mathbf{z}_{\mathbf{m},k} - \hat{\mathbf{z}}_{\mathbf{m},n_k}^{[i]}. \quad (4.18)$$

Anschließend kann die neue Systemkovarianzmatrix $\mathbf{P}_{k|k}^{[i]}$ des i -ten Partikels berechnet werden:

$$\mathbf{P}_{k|k}^{[i]} = \left[\mathbf{J}_k^\top \mathbf{Z}_k^{-1} \mathbf{J}_k + \left[\mathbf{P}_{k|k-1}^{[i]} \right]^{-1} \right]^{-1}. \quad (4.19)$$

Die verbesserte Posenschätzung wird schließlich wie folgt berechnet:

$$\hat{\mathbf{x}}_{k|k}^{[i]} = \hat{\mathbf{x}}_{k|k-1}^{[i]} + \mathbf{P}_{k|k}^{[i]} \mathbf{J}_k^\top \mathbf{Z}_k^{-1} \tilde{\mathbf{y}}_k. \quad (4.20)$$

Hierbei ist zu beachten, dass es mehrere Korrekturschritte pro Zeitschritt k geben kann, wenn mehrere Landmarken gleichzeitig beobachtet werden. Ist dies der Fall, werden die Berechnungen von Gleichung (4.6) bis (4.20) für jede Landmarke nacheinander ausgeführt und verbessern so inkrementell die vorgeschlagene Verteilung. Da die visuelle Signatur unabhängig vom Ort ist, kann diese auch nichts zur Verbesserung der vorgeschlagenen Verteilung beitragen.

Berechnung der Partikelgewichte

Anschließend können die jeweiligen Gewichte der Partikel aktualisiert werden. Das Gewicht des i -ten Partikels wird nach Gleichung (3.43) mit den erweiterten Likelihoods $p_{v,k}^{[i]}$ und $p_{\mathbf{m}_k}^{[i]}$ wie folgt berechnet:

$$w_k^{[i]} = w_{k-1}^{[i]} \frac{p(\mathbf{x}_k^{[i]} | \mathbf{x}_{k-1}^{[i]}, \mathbf{u}_k)}{p(\mathbf{x}_k^{[i]} | \mathbf{x}^{k-1,[i]}, \mathbf{z}^k, \mathbf{u}^k, \mathbf{n}^{k,[i]})} p_{v,k}^{[i]} p_{\mathbf{m}_k}^{[i]}, \quad (4.21)$$

wobei es sich bei $p(\mathbf{x}_k^{[i]} | \mathbf{x}_{k-1}^{[i]}, \mathbf{u}_k)$ wieder um das Bewegungsmodell handelt und sich $p(\mathbf{x}_k^{[i]} | \mathbf{x}^{k-1,[i]}, \mathbf{z}^k, \mathbf{u}^k, \mathbf{n}^{k,[i]})$ aus der verbesserten vorgeschlagenen Verteilung mit den Gleichungen (4.19) und (4.20) berechnen lässt.

Aktualisierung der Landmarken

Die bisher in der Karte aufgezeichneten Landmarken werden nach der Zuordnung mit den jeweils zugeordneten Beobachtungen aktualisiert, indem die neuen Informationen integriert werden. Wie bereits dargestellt, werden die geometrischen Merkmale als gaußverteilt angenommen und deren Schätzung $\hat{\mathbf{m}}$ kann somit in einem EKF-Korrekturschritt aktualisiert

werden:

$$\tilde{\mathbf{y}}_k = \mathbf{z}_{\mathbf{m},k} - h\left(\hat{\mathbf{m}}_{n_k^{[i]},k-1}\right) = \mathbf{z}_{\mathbf{m},k} - \hat{\mathbf{z}}_{\mathbf{m},n_k^{[i]}}, \quad (4.22)$$

$$\mathbf{S}_k = \mathbf{G}_k \mathbf{L}_{n_k^{[i]},k-1} \mathbf{G}_k^\top + \mathbf{R}_k, \quad (4.23)$$

$$\mathbf{K}_k = \mathbf{L}_{n_k^{[i]},k-1} \mathbf{G}_k^\top \mathbf{S}_k^{-1}, \quad (4.24)$$

$$\hat{\mathbf{m}}_{n_k^{[i]},k} = \hat{\mathbf{m}}_{n_k^{[i]},k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k, \quad (4.25)$$

$$\mathbf{L}_{n_k^{[i]},k} = (\mathbf{I} - \mathbf{K}_k \mathbf{G}_k) \mathbf{L}_{n_k^{[i]},k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{G}_k)^\top + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^\top, \quad (4.26)$$

wobei sich die prädierte Messung $\hat{\mathbf{z}}_{\mathbf{m},n_k^{[i]}}$ und das zugehörige linearisierte Messmodell \mathbf{G}_k wieder aus Gleichungen (4.6) bzw. (4.10) berechnet.

Die Schätzung der visuellen Signatur $\hat{\mathbf{v}}$ ist durch eine diskrete Dichte bzw. ein Histogramm repräsentiert und kann mittels eines diskreten Bayes-Filters aktualisiert werden [Thr05]. Für alle Klassen b_j des Histogramms wird eine Prädiktion durchgeführt:

$$\hat{\mathbf{v}}_{n_k^{[i]},k|k-1}(b_j) = \sum_i p(b_j | \mathbf{u}_k, b_i) \hat{\mathbf{v}}_{n_k^{[i]},k-1|k-1}(b_i). \quad (4.27)$$

$p(b_j | \mathbf{u}_k, b_i)$ ist dabei die Übergangswahrscheinlichkeit der Klasse b_i zu Klasse b_j . Weil die visuelle Signatur von der Position des Roboters und damit auch von der Bewegung \mathbf{u}_k unabhängig ist, gilt $p(b_j | \mathbf{u}_k, b_i) = 0$ für alle $b_j \neq b_i$. Daraus folgt, dass sich die Prädiktion zu $\hat{\mathbf{v}}_{n_k^{[i]},k|k-1} = \hat{\mathbf{v}}_{n_k^{[i]},k-1|k-1}$ vereinfacht.

Beim Korrekturschritt werden die jeweiligen Histogrammklassen b der Landmarke mit denen der Beobachtung $\mathbf{z}_{\mathbf{v},k}(b)$ multipliziert:

$$\hat{\mathbf{v}}_{n_k^{[i]},k|k}(b) = \eta \mathbf{z}_{\mathbf{v},k}(b) \hat{\mathbf{v}}_{n_k^{[i]},k|k-1}(b), \quad (4.28)$$

wobei η ein Normalisierungsfaktor ist, mit dem das Histogramm normalisiert wird, sodass wieder $\sum_{b=1}^B \hat{\mathbf{v}}_{k|k}(b) = 1$ gilt [Ulr10; Emt10c].

4.1.3 Zusammenfassung

Bei dem vorgestellten Verfahren wird ein erweitertes Landmarkenmodell dazu verwendet, die Wiedererkennung durch zusätzliche ortsunabhängige und visuelle Merkmale robuster zu machen. Bei dem zugrundeliegenden SLAM-Algorithmus handelt es sich um ein RBPF inklusive der Verbesserung der vorgeschlagenen Verteilung. Während in [Dur06b] eine ansichtsbasierte Signatur lediglich zur Erkennung eines Schleifenschlusses dient, werden hier die zusätzlichen Merkmale in probabilistisch konsistenter Weise in die Schätzung des SLAM-Posteriors integriert.

In [Ram07] wurde ein Verfahren vorgestellt, welches ebenfalls ein erweitertes Landmarkenmodell mit visueller Signatur zur robusteren Wiedererkennung verwendet. Im Gegensatz zu dem hier vorgestellten Verfahren, welches auf einem RBPF basiert, wurde dort ein EKF als Schätzer verwendet, welcher gegenüber dem Partikelfilter einige Nachteile aufweist. So können nur multivariate Gaußdichten für die Merkmalsmodellierung verwendet werden, wie in Abschnitt 2.2.1 erläutert. Bei dem hier vorgestellten SLAM-Algorithmus kann durch die Verwendung des Partikelfilters eine wesentlich flexiblere Modellierung der Landmarkenmerkmale erfolgen. Durch die angenommene Unabhängigkeit der visuellen Signatur kann diese mit dem Multiplikationssatz separiert und ein separater Likelihood über ein geeignetes Abstandsmaß berechnet werden.

4.2 SLAM – 2D-Rasterkarte

In unstrukturierten Umgebungen, in denen keine charakteristischen und klar unterscheidbaren Objekte vorhanden sind, ist die Kartierung mit dichten Karten u. U. besser geeignet. Hierfür wird im Folgenden ein eigenes Verfahren vorgestellt, welches einen Ansatz für eine flexible Integration von Erweiterungen verfolgt. Das Verfahren basiert auf einem RBPF und lehnt sich stark an [Gri05] und [Gri07] an, vgl. Abschnitt 3.2.3. Es hat jedoch entscheidende Unterschiede, um die modulare Erweiterbarkeit zu

realisieren. Diese Erweiterungen werden anschließend in den folgenden Abschnitten 4.3 und 5.1 vorgestellt.

Da es sich um 2D-SLAM handelt, ist der Zustand der mobilen Plattform deren 2D-Pose $\mathbf{x} = [x, y, \psi]$, d. h. x -Koordinate, y -Koordinate und Gierwinkel ψ . Es wird zunächst, wie bei dem in Abschnitt 3.2.3 erläuterten Verfahren, für jedes Partikel eine Prädiktion mit den Messungen aus der Rad-Odometrie mit einem entsprechenden Bewegungsmodell und anschließender Registrierung in der zum Partikel gehörigen Karte durchgeführt. Neben dem generischen Bewegungsmodell aus [Thr05] ist je nach Roboterplattform auch die Verwendung alternativer Bewegungsmodelle möglich. Die Modelle umfassen hierbei zusätzlich eines für mobile Roboter mit Differential-Antrieb, d. h. zwei individuell angetriebene Räder senkrecht zum Drehpunkt und passive Schleppräder zur Abstützung, und ein ebenfalls generisches Modell, bestehend aus Vorwärtsgeschwindigkeit und Drehrate.

Das anschließende Scan-Matching wird mit einer Methode basierend auf dem Bergsteigeralgorithmus realisiert. Die dabei eingesetzte Zielfunktion besteht aus einem modifizierten probabilistischen Modell für die kleinsten Entfernungen der Strahlenendpunkte des 2D-LiDAR-Scans nach [Thr05]. Das modifizierte probabilistische Modell deckt zusätzlich zur Varianz des Messfehlers σ_L^2 die Tatsache ab, dass die Laserstrahlen mit der zurückgelegten Distanz zum LiDAR-Sensor d_L divergieren und damit eine zusätzliche Messunsicherheit vorhanden ist, welche proportional zur Messdistanz steigt. Der Messfehler σ_L^2 bezieht sich auf die Abstandsrichtung, wohingegen die Strahlaufweitung senkrecht dazu ist. Dennoch hat die Strahlaufweitung auch einen Einfluss auf die Entfernungsmessung, wenn der Einfallswinkel zur Oberfläche flach ist. Zusätzlich ergibt sich durch die Rasterung der Belegtheitskarte eine diskrete Abtastung mit einer entsprechenden Fehlerverteilung, welche approximativ als gleichverteilt über die Zellengröße ζ angenommen werden kann. Damit wird die Gesamt-Varianz

σ_n^2 des n -ten Laserstrahls wie folgt berechnet:

$$\sigma_n^2 = \sigma_L^2 + ((\varsigma d_{L,n})^2 + (\zeta)^2)/12, \quad (4.29)$$

mit ς als Faktor für die Strahlaufweitung, welche neben der Varianz des Messfehlers σ_L^2 aus dem Datenblatt des LiDARs entnommen werden kann.

Es werden zunächst die euklidischen Distanzen d_n von den N Strahlenendpunkten zur jeweils nächsten belegten Zelle ermittelt. Der Likelihood des gesamten Scans bestehend aus N Strahlen ergibt sich zu

$$p_N = \prod_{n=1}^N \frac{1}{\sqrt{2\pi\sigma_n}} e^{-\frac{1}{2}\left(\frac{d_n}{\sigma_n}\right)^2}. \quad (4.30)$$

Nach dem Scan-Matching wird wieder ähnlich zu dem Ansatz in Abschnitt 3.2.3 eine Normalverteilung der Lokalisierung geschätzt. Während dort in Gleichungen (3.50) und (3.51) die Bewegungsschätzung durch Gewichtung bei der Mittelwertbestimmung bzw. der Konstruktion der Kovarianzmatrix mit eingeht, wird hier nur eine Schätzung der Lokalisierung in der Karte für jedes Partikel durchgeführt:

$$\hat{\chi}_k^{[i]} = \frac{1}{\chi^{[i]}} \sum_{j=1}^J p(\mathbf{z}_k | \Lambda_{k-1}^{[i]}, \mathbf{x}_j) \mathbf{x}_j, \quad (4.31)$$

$$\hat{\mathbf{X}}_k^{[i]} = \frac{1}{\chi^{[i]}} \sum_{j=1}^J p(\mathbf{z}_k | \Lambda_{k-1}^{[i]}, \mathbf{x}_j) (\mathbf{x}_j - \hat{\chi}_k^{[i]})(\mathbf{x}_j - \hat{\chi}_k^{[i]})^\top, \quad (4.32)$$

mit dem modifizierten Normalisierungsfaktor

$$\chi^{[i]} = \sum_{j=1}^J p(\mathbf{z}_k | \Lambda_{k-1}^{[i]}, \mathbf{x}_j), \quad (4.33)$$

wobei $p(\mathbf{z}_k | \Lambda_{k-1}^{[i]}, \mathbf{x}_j)$ gerade dem Likelihood des gesamten Scans p_N mit der jeweiligen Pose \mathbf{x}_j entspricht.

Die Fusion der Prädiktion aus der Bewegungsschätzung mit der Lokalisierung erfolgt anschließend über ein erweitertes Kalman-Filter für jedes Partikel, wobei hier die Messmatrix entfällt, da die Schätzung von $\hat{\mathbf{x}}_k^{[i]}$ und $\hat{\mathbf{X}}_k^{[i]}$ im Zustandsraum erfolgt.

$$\tilde{\mathbf{y}}_k^{[i]} = \hat{\mathbf{x}}_k^{[i]} - \hat{\mathbf{x}}_{k|k-1}^{[i]} \quad (4.34)$$

$$\mathbf{K}_k^{[i]} = \mathbf{P}_{k|k-1}^{[i]} \left(\mathbf{P}_{k|k-1}^{[i]} + \hat{\mathbf{X}}_k^{[i]} \right)^{-1} \quad (4.35)$$

$$\hat{\mathbf{x}}_{k|k}^{[i]} = \hat{\mathbf{x}}_{k|k-1}^{[i]} + \mathbf{K}_k^{[i]} \tilde{\mathbf{y}}_k^{[i]} \quad (4.36)$$

$$\mathbf{P}_{k|k}^{[i]} = \left(\mathbf{I} - \mathbf{K}_k^{[i]} \right) \mathbf{P}_{k|k-1}^{[i]} \left(\mathbf{I} - \mathbf{K}_k^{[i]} \right)^\top + \mathbf{K}_k^{[i]} \hat{\mathbf{X}}_k^{[i]} \mathbf{K}_k^{[i]\top} \quad (4.37)$$

Somit geht auch die Lokalisierung in die vorgeschlagene Verteilung mit ein.

Die Gewichte der Partikel lassen sich anschließend wie folgt berechnen:

$$w_k^{[i]} = w_{k-1}^{[i]} \frac{p\left(\mathbf{x}_k^{[i]} \mid \mathbf{x}_{k-1}^{[i]}, \mathbf{u}_k\right)}{p\left(\mathbf{x}_k^{[i]} \mid \mathbf{x}^{k-1, [i]}, \mathbf{z}^k, \mathbf{u}^k, \mathbf{n}^{k, [i]}\right)} \mathcal{X}^{[i]}, \quad (4.38)$$

wobei es sich bei $p\left(\mathbf{x}_k^{[i]} \mid \mathbf{x}_{k-1}^{[i]}, \mathbf{u}_k\right)$ wieder um das Bewegungsmodell handelt und sich $p\left(\mathbf{x}_k^{[i]} \mid \mathbf{x}^{k-1, [i]}, \mathbf{z}^k, \mathbf{u}^k, \mathbf{n}^{k, [i]}\right)$ aus der verbesserten vorgeschlagenen Verteilung mit den Gleichungen (4.36) und (4.37) berechnen lässt.

4.2.1 Zusammenfassung

Das vorgestellte SLAM-Verfahren dient in dieser Form zur Erstellung von dichten 2D-Rasterkarten auf Basis von Rad-Odometrie- und 2D-LiDAR-Daten. Der zugrundeliegende SLAM-Algorithmus ist an [Gri05] und [Gri07] angelehnt und es handelt sich um ein RBPF mit einer Verbesserung der vorgeschlagenen Verteilung. Es unterscheidet sich jedoch von diesen in der Fusion mit einem EKF je Partikel für die Integration der Lokalisierung, wodurch eine Modularität hinsichtlich der Kombination

mehrerer Kartierungs- und damit zusammenhängender Lokalisierungsmethoden erreicht wird. Somit bietet das modulare SLAM-Verfahren die Möglichkeit, über das EKF pro Partikel probabilistisch konsistent weitere Sensoren oder Lokalisierungsmethoden zu integrieren, was in den folgenden Abschnitten vorgestellt wird.

4.3 Kombination verschiedener Kartenarten

Werden als Landmarken bestimmte Objektarten verwendet, kann es in unstrukturierten Bereichen, in denen dieser Objekttyp nicht vorhanden ist, zu einer Verschlechterung der Lokalisierungsgenauigkeit und damit auch der gesamten Kartierung kommen. In gemischten Umgebungen, d. h. Umgebungen, die sowohl vereinzelte Landmarkenobjekte beinhalten als auch unstrukturierte Bereiche, kann es sinnvoll sein, verschiedene Kartenarten zu kombinieren. Diese Kombination verschiedener Kartenarten kann auch als eine Fusion auf Kartenebene interpretiert werden.

4.3.1 Landmarkenbasierte und dichte Karte

In unstrukturierten Bereichen sind Landmarken meist schwierig zu extrahieren, weshalb hier Rasterkarten besser geeignet sein können. Im Gegensatz dazu kann es in dünn besetzten Gebieten bei Rasterkarten eher dazu kommen, dass das Scan-Matching ungenau ist oder fehlschlägt, wohingegen einzelne Landmarken hier leichter zu detektieren sind und die Lokalisierung in einer Landmarkenkarte erlauben. Dadurch, dass die Methoden in jeweils unterschiedlichen Bereichen eine genaue Lokalisierung erlauben, können sie sich durch ihre Kombination ergänzen.

Für die Kombination verschiedener Kartierungsarten wurde ein modularer SLAM-Algorithmus entwickelt [Emt12b]. Dieser basiert auf dem in Abschnitt 4.2 eingeführten RBPF und jedes Partikel enthält einen EKF für die Berechnung und Verbesserung der vorgeschlagenen Verteilung. Hierdurch wird die Einbeziehung mehrerer Lokalisierungsschätzungen von

unterschiedlichen Karten ermöglicht. Mit der Fusion in die vorgeschlagene Verteilung und damit deren Verbesserung basiert der Ansatz weiterhin auf dem Konzept des FastSLAM-2.0-Algorithmus. Dabei handelt es sich um einen modularen Ansatz, d. h., jedes Kartierungsmodul kann als Komponente angesehen werden und es ist möglich, eine beliebige Anzahl von Kartierungsmodulen in das SLAM-Verfahren zu integrieren. Voraussetzung hierfür ist, dass ein Modul neben einer Lokalisierungsschätzung die Möglichkeit bietet, einen Likelihood der aktuellen Beobachtungen bezüglich der zugehörigen Karte für die Bestimmung der Gewichte der Partikel zu berechnen [Emt12a].

Mit dem modularen SLAM-Algorithmus lassen sich die in den vorangegangenen Abschnitten 4.1 und 4.2 vorgestellten Methoden für gemischte Umgebungen zu einem hybriden SLAM-Algorithmus kombinieren. Die Einbeziehung der Lokalisierungsschätzungen erfolgt dabei zum einen über die inkrementelle Verbesserung der vorgeschlagenen Verteilung mit den beobachteten und in der Karte zugeordneten Landmarken, wie in Abschnitt 4.1.2 mit den Gleichungen von (4.6) bis (4.20) beschrieben. Zum anderen erfolgt sie durch die Fusion der Lokalisierung in der Rasterkarte in dem EKF, wie in Abschnitt 4.2 mit den Gleichungen von (4.34) bis (4.37) erläutert.

Die Gewichte der Partikel lassen sich für den hybriden SLAM-Algorithmus wie folgt berechnen:

$$w_k^{[i]} = w_{k-1}^{[i]} \frac{p(\mathbf{x}_k^{[i]} | \mathbf{x}_{k-1}^{[i]}, \mathbf{u}_k)}{p(\mathbf{x}_k^{[i]} | \mathbf{x}^{k-1, [i]}, \mathbf{z}^k, \mathbf{u}^k, \mathbf{n}^{k, [i]})} p_{\mathbf{m}, k}^{[i]} p_{\mathbf{v}, k}^{[i]} \chi^{[i]}, \quad (4.39)$$

d. h. durch eine Kombination von Gleichung (4.21) von den erweiterten Landmarken und Gleichung (4.38) von der Rasterkarte.

4.3.2 Zusammenfassung

Durch die konsequente Auslegung in probabilistisch konsistenter Weise der in Abschnitten 4.1 und 4.2 vorgestellten Verfahren können diese mit dem modularen SLAM-Algorithmus in einem integrierten hybriden Ansatz kombiniert werden, um deren gegenseitige Vorteile für eine robustere Kartierung und Lokalisierung in gemischten Umgebungen in ergänzender Weise zu nutzen.

Kapitel 5

Integration von Multi-Sensor-Fusion und SLAM

Bei den im vorherigen Kapitel vorgestellten SLAM-Verfahren kommt als Bewegungssensor lediglich Rad-Odometrie zum Einsatz. Bei der Odometrie handelt es sich um einen relativ messenden Sensor, d. h., es wird die Differenz bzw. die Relation zwischen zwei Posen gemessen. Als relativ messender Sensor ist Odometrie zwar lokal sehr genau, wenn kein starker Schlupf auftritt, kann jedoch über die Zeit driften. Durch die Fusion von Odometrie mit einem absoluten Sensor wie einem GNSS kann der globale Fehler bei der Lokalisierung begrenzt und somit die Drift korrigiert werden [Emt08].

Die Lokalisierung in einer Karte, welche simultan aufgenommen wird, unterliegt ähnlichen Problemen wie die Odometrie: Wird über lange Zeit unbekanntes Gebiet exploriert, vergrößert sich die Unsicherheit der Robo-terpose und das Schließen von Schleifen wird schwieriger. Deswegen sind

vor allem große Schleifen bei SLAM eine besondere Herausforderung. Durch die Integration von Multi-Sensor-Fusion und SLAM sollen nun die Vorteile der Multi-Sensor-Fusion zur Lokalisierung mit SLAM kombiniert werden, d. h., es werden zusätzliche absolute Sensoren integriert, um die Unsicherheit global zu begrenzen und damit das Schließen von Schleifen zu vereinfachen.

5.1 RBPF-SLAM in Kombination mit Multi-Sensor-Fusion

Je größer eine zu schließende Schleife ist, desto mehr Partikel werden i. A. bei den RBPF-SLAM-Algorithmen benötigt. Dies rührt daher, dass während der ersten Befahrung der Schleife ständig neues Gebiet erkundet wird und damit die Lokalisierungsunsicherheit anwächst. Da die Partikelzahl begrenzt ist und durch das Resampling in dem wahrscheinlichsten Bereich des Zustandsraums gehalten wird, kann die anwachsende Unsicherheit nicht mehr adäquat abgedeckt werden, was in der englischen Literatur auch als *particle depletion* bekannt ist [Mon07].

Durch einen neuartigen Ansatz mit der Kombination von SLAM und Multi-Sensor-Fusion zur Integration eines absoluten Positionssensors soll die Lokalisierungsunsicherheit für ein robusteres Schließen von Schleifen begrenzt werden. Als Verfahren wird der modulare RBPF-SLAM-Algorithmus eingesetzt, welcher auf der FastSLAM-2.0-Methodik basiert und bereits in den vorangegangenen Abschnitten vorgestellt wurde. So wurde der modulare RBPF-SLAM-Algorithmus beispielsweise in Abschnitt 4.2 für die Kartierung anhand von Rad-Odometrie- und 2D-LiDAR-Sensoren mit 2D-Rasterkarten vorgestellt und auf dessen Entwicklung in Hinblick auf eine Erweiterbarkeit mit zusätzlichen Sensoren und Lokalisierungsmethoden eingegangen.

Durch die Integration eines absoluten Positionssensors kann die Unsicherheit selbst bei großen Schleifen global begrenzt werden. Während

Multi-Sensor-Fusion im Allgemeinen zu genaueren Schätzungen führt, d. h., die Genauigkeit der Lokalisierung und der Karte erhöht wird, kann die Begrenzung der Unsicherheit bei SLAM darüber hinaus das Schließen von Schleifen vereinfachen und damit auch die Konsistenz der Kartierung verbessern. Da bei Partikelfilter-SLAM umso mehr Partikel benötigt werden, je größer die zu schließende Schleife ist, kann durch die Begrenzung der Unsicherheit eine Reduzierung der benötigten Anzahl der Partikel erzielt werden [Emt12a].

In Abschnitt 4.3 wurde darauf eingegangen, wie sich die Kombination verschiedener Kartenarten dazu eignet, die Lokalisierung und Kartierung sowohl in dicht besetzten und unstrukturierten Umgebungen als auch in spärlich besetzten Umgebungen mit nur wenigen Landmarkenobjekten zu erlauben. Sind weder Landmarken noch andere beobachtbare Objekte in der Umgebung, kann durch die Kombination mit einem absoluten Positionssensor mittels Multi-Sensor-Fusion die Lokalisierung weiterhin gestützt werden. In Outdoor-Umgebungen liegt der Einsatz eines GNSS als absolutem Positionssensor nahe, da es global verfügbar ist und sich sehr gut mit den Umgebungssensoren ergänzt. Seine Genauigkeit ist in offenen Gebieten mit freiem Himmel besonders hoch und kann die hier eingeschränkte Lokalisierung in der Karte kompensieren. In modernen GPS- bzw. GNSS-Empfängern können zwar viele Fehlerquellen, wie z. B. Störungen der Ausbreitung in Iono- und Troposphäre oder Uhrenfehler, durch differentielle Verfahren oder die Verwendung von Korrektursignalen minimiert werden, jedoch nicht lokale Phänomene wie die an hohen Objekten entstehende Mehrwegeausbreitung oder Abschattungen [Bra01; Bra17]. D. h., besonders in Gebieten mit hohen Bäumen oder urbanen Gegenden mit hohen Häusern ist bei einem GNSS mit größeren Fehlern zu rechnen [Zha14a]. Gerade in diesen dicht besetzten Gebieten ist wiederum zu erwarten, dass die Lokalisierung in der Karte sehr genau ist.

Es wird zunächst für jedes Partikel eine Prädiktion mit den Messungen der Rad-Odometrie und einem entsprechenden Bewegungsmodell durchgeführt. Die Fusion der Prädiktion aus der Bewegungsschätzung mit der

Messung des GNSS erfolgt anschließend über ein erweitertes Kalman-Filter pro Partikel im Korrekturschritt:

$$\hat{\mathbf{y}}_k^{[i]} = \mathbf{z}_{\text{GNSS},k} - h\left(\hat{\mathbf{x}}_{k|k-1}^{[i]}\right), \quad (5.1)$$

$$\mathbf{K}_k^{[i]} = \mathbf{P}_{k|k-1}^{[i]} \mathbf{H}_k^\top \left(\mathbf{H}_k \mathbf{P}_{k|k-1}^{[i]} \mathbf{H}_k^\top + \mathbf{R}_{\text{GNSS},k} \right)^{-1}, \quad (5.2)$$

$$\hat{\mathbf{x}}_{k|k}^{[i]} = \hat{\mathbf{x}}_{k|k-1}^{[i]} + \mathbf{K}_k^{[i]} \hat{\mathbf{y}}_k^{[i]}, \quad (5.3)$$

$$\mathbf{P}_{k|k}^{[i]} = \left(\mathbf{I} - \mathbf{K}_k^{[i]} \mathbf{H}_k \right) \mathbf{P}_{k|k-1}^{[i]} \left(\mathbf{I} - \mathbf{K}_k^{[i]} \mathbf{H}_k \right)^\top + \mathbf{K}_k^{[i]} \mathbf{R}_{\text{GNSS},k} \mathbf{K}_k^{[i],\top}. \quad (5.4)$$

Die Messfehlerkovarianzmatrix \mathbf{R}_k wird für die meisten Sensoren als konstant modelliert und der Index k könnte entfallen. Bei einem modernen GNSS-Empfänger ist $\mathbf{R}_{\text{GNSS},k}$ jedoch von k abhängig, da der Empfänger neben der eigentlichen Positionsschätzung auch eine aktuelle Messunsicherheit mit ausgibt. In diese gehen neben der momentanen Satellitenkonstellation u. A. auch die Signal-zu-Rauschabstände der Satellitensignale mit ein.

Erst nach der Fusion mit den Messungen des GNSS wird für jedes Partikel das Scan-Matching in der jeweiligen Karte durchgeführt, vgl. Abschnitt 4.2. Da hier durch die Fusion mit dem absoluten Positionssensor bereits eine Stützung der relativen Rad-Odometrie erfolgt ist, kann dies u. U. schon das Scan-Matching verbessern, da es sich dabei um ein lokales Verfahren handelt und eine falsche Startpose dazu führen kann, dass nur ein lokales Optimum gefunden wird. Anschließend erfolgt für jedes Partikel die Fusion der Lokalisierung in der jeweiligen Rasterkarte mittels EKF, wie in den Gleichungen von (4.34) bis (4.37) erläutert.

Die Gewichte der Partikel lassen sich schließlich wie folgt berechnen:

$$w_k^{[i]} = w_{k-1}^{[i]} \frac{p\left(\mathbf{x}_k^{[i]} \mid \mathbf{z}_{\text{GNSS},k}, \mathbf{x}_{k-1}^{[i]}, \mathbf{u}_k\right)}{p\left(\mathbf{x}_k^{[i]} \mid \mathbf{x}^{k-1,[i]}, \mathbf{z}^k, \mathbf{u}^k, \mathbf{n}^k, [i]\right)} \chi^{[i]}, \quad (5.5)$$

wobei $p(\mathbf{x}_k^{[i]} | \mathbf{z}_{\text{GNSS},k}, \mathbf{x}_{k-1}^{[i]}, \mathbf{u}_k)$ aus der Fusion des Bewegungsmodells mit dem GNSS nach Gleichungen (5.3) und (5.4) berechnet wird. Die restlichen Komponenten werden analog zu Gleichung (4.38) berechnet.

Durch die Einbeziehung eines globalen Positionssensors kann auch direkt eine globale Referenzierung der Karte erfolgen, d. h., die entstehende Karte ist geo-referenziert.

5.1.1 Zusammenfassung

Durch die konsequente Auslegung in probabilistisch konsistenter Weise des in Abschnitt 4.2 vorgestellten modularen SLAM-Algorithmus können zusätzliche Lokalisierungssensoren integriert werden. Hierbei hat vor allem die Integration absoluter Sensoren den Vorteil, die Unsicherheit global zu begrenzen und damit das Schließen von Schleifen zu vereinfachen, was letztendlich das Erstellen konsistenter Karten erleichtert.

In [Gui02a] wurde bereits ein hochgenaues RTK-GPS mit EKF-SLAM und Landmarkenkarte kombiniert. Der hier vorgestellte Ansatz integriert im Vergleich dazu ein GNSS in einen RBPF-SLAM-Algorithmus mit 2D-Rasterkarte und ist dabei nicht darauf ausgelegt, dass das GNSS eine hohe Genauigkeit aufweisen muss. Auch in [Sha18a] wurde ein hochgenaues RTK-GPS in einen RBPF-SLAM-Algorithmus und 2D-Rasterkarte integriert, wobei das GPS nur in die Gewichtung der Partikel eingeht, d. h., das GPS trägt nicht zur Verbesserung der vorgeschlagenen Verteilung bei und es müssen u. U. viele Partikel beim Resampling eliminiert werden, vgl. Abschnitt 3.2.1. Bei dem hier vorgestellten Ansatz nach der FastSLAM-2.0-Methodik geht das GNSS auch in die vorgeschlagene Verteilung ein und trägt somit zu ihrer Verbesserung bei.

5.2 Stochastisches Klonen

In den bisher vorgestellten Verfahren kommt genau ein relativer Sensor – Rad-Odometrie – zum Einsatz und dieser geht in den Prädiktions-

schritt ein. Sollen weitere relative Sensoren integriert werden, so können diese nur über einen Korrekturschritt einbezogen werden. Die Standard-Formulierung des EKF reicht dann zwar nicht mehr aus, da bei relativen Messungen der aktuelle Zustand von einem vorhergehenden Zustand abhängt. Mit der Methode des sogenannten stochastischen Klonens können jedoch weitere relative Sensoren formal korrekt in das EKF integriert werden, siehe Abschnitt 3.1.3.

5.2.1 Implizites stochastisches Klonen

Beim stochastischen Klonen wird der zu einer relativen Messung zugehörige Zustand mit dessen Klon erweitert. Für zwei relative Sensoren, einer mit höherer Rate (m -fach) für die Prädiktion und einer für den Korrekturschritt mit stochastischem Klonen, kann gezeigt werden, dass ein explizites Klonen nicht notwendig ist [Rou02]. D. h., die augmentierte Systemkovarianz $\check{\mathbf{P}}_{k+m|k}$ kann aus der geklonten Systemkovarianzmatrix $\mathbf{P}_{k|k}$ und den Zustandsübergangsmatrizen $\mathbf{F}_{k+1} \dots \mathbf{F}_{k+m}$ rekonstruiert werden, vgl. Gleichung (3.23). Die geklonte Zustandsschätzung $\hat{\mathbf{x}}_{k|k}$ wird nicht durch die Prädiktionen beeinflusst und deshalb auch als statische Zustandsschätzung bezeichnet. Bei der Korrektur wird dann nur die aktuelle Zustandsschätzung $\hat{\mathbf{x}}_{k+m|k+m}$ aktualisiert, vgl. Gleichung (3.31). Da hierbei nicht explizit geklont wird, soll diese Methode im Folgenden als implizites Klonen bezeichnet werden.

Soll, wie im vorherigen Abschnitt dargestellt, ein zusätzlicher absoluter Sensor, wie z. B. ein GNSS, integriert werden, ist das implizite Klonen nur unter bestimmten Umständen möglich. Dabei muss die Rate des absoluten Sensors synchronisiert und kleiner oder gleich des zusätzlichen relativen Sensors sein. Außerdem muss die absolute Korrektur vor dem Klonen des aktuellen Zustands erfolgen. Abbildung 5.1 zeigt eine Sequenz mit synchronisierten relativen und absoluten Messungen als dynamisches bayessches Netz (DBN). Relative Messungen zwischen den Zuständen \mathbf{x}_k und \mathbf{x}_{k+m} werden mit $\mathbf{z}_{k,k+m}$ bezeichnet und absolute Messungen zum Zeitpunkt k

mit \mathbf{z}_k . Die Zustände \mathbf{x}_k sind hellblau eingefärbt und die absoluten Messungen \mathbf{z}_k in Orange bzw. die relativen Messungen $\mathbf{z}_{k,k+m}$ in Hellgrün.

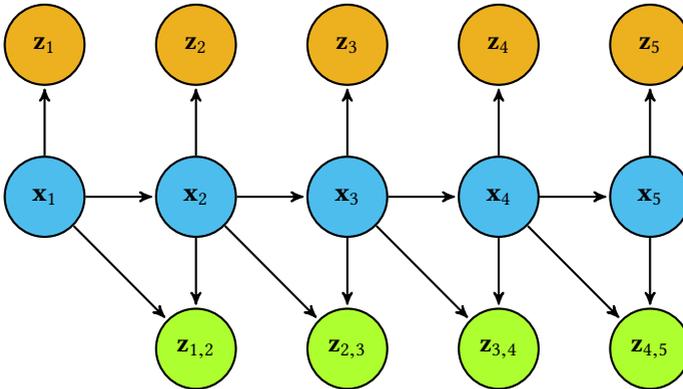


Abbildung 5.1: DBN mit relativen und synchronen absoluten Messungen.

Es kann hier implizit geklont werden, wenn die absoluten Korrekturen nur dann integriert werden, während kein Klon existiert. Am Beispiel von Zustand \mathbf{x}_2 in Abbildung 5.1 bedeutet dies, dass zunächst die Korrektur der relativen Messung $\mathbf{z}_{1,2}$ durchgeführt wird, anschließend die absolute Messung \mathbf{z}_2 integriert wird und erst dann das Klonen des Zustands für die relative Messung $\mathbf{z}_{2,3}$ erfolgt. Diese Bearbeitungssequenz ist mit den Zwischenschritten nochmals als Teilausschnitt in Abbildung 5.2 als DBN dargestellt. Nachdem der Zustand \mathbf{x}_2 durch die relative Messung $\mathbf{z}_{1,2}$ aktualisiert wurde, wird er durch die absolute Messung \mathbf{z}_2 zu \mathbf{x}'_2 aktualisiert. Anschließend erfolgt das Klonen von Zustand \mathbf{x}'_2 , wobei sich \mathbf{x}''_2 gegenüber \mathbf{x}'_2 nicht verändert hat. Mit Zustand \mathbf{x}''_2 erfolgt dann die Aktualisierung durch die relative Messung $\mathbf{z}_{2,3}$ zu \mathbf{x}_3 mittels implizitem stochastischem Klonen. Diese Aktualisierung weist somit die gleiche Struktur auf wie das ursprüngliche stochastische Klonen mit nur einem zusätzlichen relativen Sensor, vgl. Abbildung 3.2.

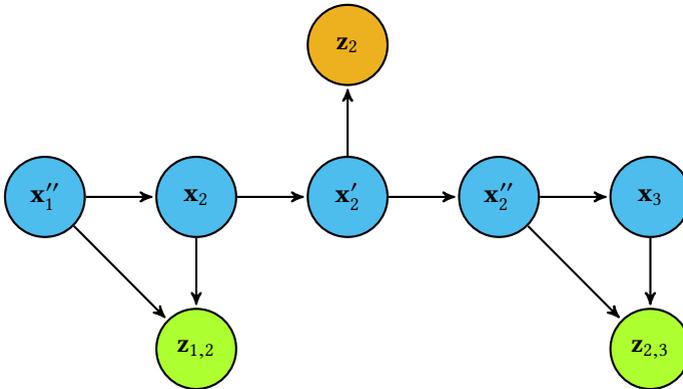


Abbildung 5.2: DBN mit relativen und synchronen absoluten Messungen. Darstellung mit Zwischenschritten.

Abbildung 5.3 zeigt ein DBN einer Sequenz, bei der nun asynchrone absolute Messungen zu verarbeiten sind. Sind die absoluten Messungen nicht mehr synchron, können sie auch zu Zeitpunkten auftreten, welche von relativen Messungen überspannt werden. Hier lässt sich die absolute Korrektur nicht mehr sequentiell in Zwischenschritten integrieren, wie dies am Beispiel von Abbildung 5.2 ausgeführt wurde. Es wäre natürlich möglich, relative Messungen, welche eine absolute Messung überspannen, auszulassen, was jedoch suboptimal ist, da nicht alle verfügbaren Informationen genutzt werden. Im vorliegenden Fall, wie in Abbildung 5.3 dargestellt, könnte keine der relativen Messungen integriert werden.

Abbildung 5.4 zeigt einen Teil der Sequenz in einer alternativen Darstellung mit räumlichen Bezügen über die Zeit mit sich abwechselnden relativen und absoluten Messungen. Auch hier werden die absoluten Messungen zum Zeitpunkt k mit \mathbf{z}_k bezeichnet und relative Messungen mit $\mathbf{z}_{k,k+m}$, wobei $m = 2$ ist. Die prädierten Zustandsschätzungen $\hat{\mathbf{x}}_{k|k-1}$ und die aktualisierten Zustandsschätzungen $\hat{\mathbf{x}}_{k|k}$ sind hellblau eingefärbt. Die absoluten Messungen \mathbf{z}_k sind in Orange bzw. die relativen Messungen $\mathbf{z}_{k,k+m}$ in Hellgrün eingefärbt. Die Pfeile zeigen hier die Abhängigkeiten

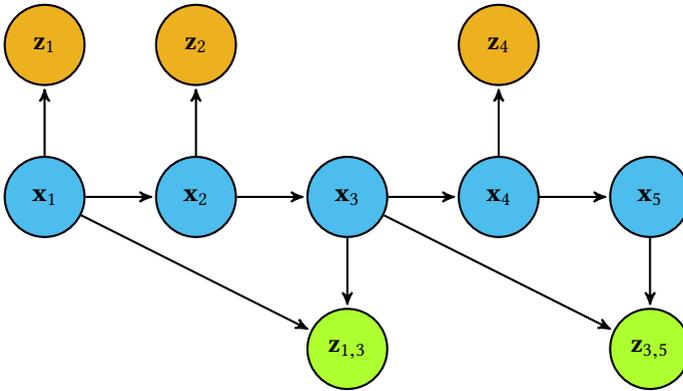


Abbildung 5.3: DBN mit relativen und asynchronen absoluten Messungen.

von den Messungen und die Zustandsübergänge. Die Prädiktionen von $\hat{\mathbf{x}}_{k|k}$ zu $\hat{\mathbf{x}}_{k+1|k}$ sind mit „prääd.“ und die Korrekturschritte von $\hat{\mathbf{x}}_{k+1|k}$ zu $\hat{\mathbf{x}}_{k+1|k+1}$ mit „korr.“ gekennzeichnet.

Die absoluten Messungen treten in Abbildung 5.4 nicht zeitgleich mit den relativen Messungen auf. So korrigiert in einem Korrekturschritt die absolute Messung z_2 eine potentielle Drift von $\hat{\mathbf{x}}_{2|2}$. Der Korrekturschritt mit einer absoluten Messung hat keinen Einfluss auf vorherige Zustandsschätzungen und korrigiert damit auch nicht die Drift der zurückliegenden Schätzung $\hat{\mathbf{x}}_{1|1}$. Wird zum nächsten Zeitpunkt der Korrekturschritt mit der relativen Messung $z_{1,3}$ mittels stochastischem Klonen durchgeführt, wird diese Drift nach $\hat{\mathbf{x}}_{3|3}$ weiterpropagiert und potentiell vergrößert. Mit fortschreitender Zeit könnte sich diese Drift trotz Vorhandensein eines absoluten Sensors unbegrenzt vergrößern und letztendlich zu einem instabilen Filter führen [Eml18b]. Um diesem entgegenzuwirken, sollten die Korrekturen des absoluten Sensors auch den geklonten Zustand aktualisieren, und somit das Degenerieren des Filters unterbinden.

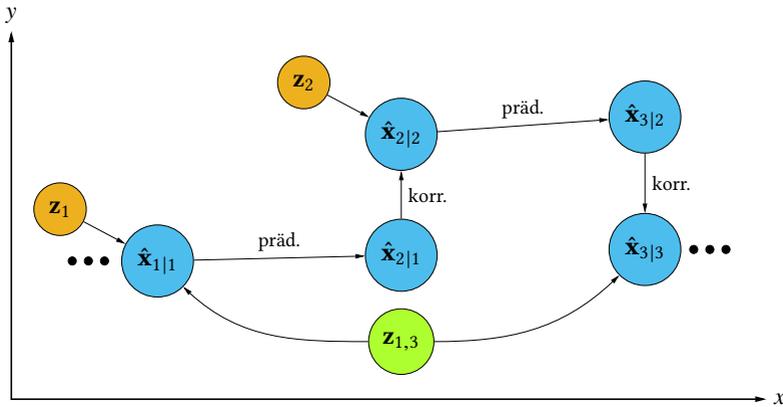


Abbildung 5.4: Illustration von sich abwechselnden absoluten (z_k) und relativen ($z_{k,k+m}$) Messungen mit Ortsbezug.

5.2.2 Explizites Klonen

In [Mou07] wurde die Behandlung von zusätzlichen Messungen als mögliche Erweiterung zu dem in Abschnitt 5.2.1 beschriebenen Ansatz des impliziten Klonens erwähnt. Wenn über die zwei relativen Sensoren hinaus zusätzliche Sensoren zu integrieren sind, ist danach die Vereinfachung mit der Rekonstruktion der augmentierten Matrix $\check{\mathbf{P}}_{k+m|k}$ mit (3.24) und (3.23) nicht mehr gültig. Es muss dann explizit geklont werden und die augmentierte Systemkovarianzmatrix durch alle Filterschritte propagiert werden, es existiert jedoch – sofern bekannt – keine weiterführende Literatur hierzu. In Abgrenzung zu der Bezeichnung implizites Klonen soll diese Methode im Folgenden mit explizitem Klonen bezeichnet werden.

Dass die reine Anwendung des impliziten Klonens bei zusätzlichen asynchronen Sensoren nicht mehr gültig ist, wurde auch im vorherigen Abschnitt dargestellt. Es wurde gezeigt, dass bei absoluten Messungen, welche in von relativen Messungen überspannten Zeiträumen integriert werden, die Anwendung des impliziten Klonens zu divergierenden Schätzungen führen kann. Dass beim expliziten Klonen die augmentierte Systemko-

varianzmatrix durch alle Filterschritte propagiert werden muss, damit auch die geklonte Zustandsschätzung $\hat{\mathbf{x}}_{k|k}$ aktualisiert wird, soll im Folgenden dargestellt werden.

Weil seit dem Klonen weitere Sensormessungen hinzugekommen sind, ist die augmentierte Systemkovarianzmatrix nach der letzten Prädiktion vor dem Korrekturschritt mittels stochastischem Klonen nicht mehr $\check{\mathbf{P}}_{k+m|k}$, sondern wird vom vorherigen Zeitschritt $k+n-1$ im Prädiktionschritt zu $\check{\mathbf{P}}_{k+n|k+n-1}$, wobei n in $]1, m[$ liegt.

Im Gegensatz zur Messmatrix beim Korrekturschritt mit einer relativen Messung mittels stochastischem Klonen (vgl. Gleichung (3.28)) ist die linke Submatrix von $\check{\mathbf{H}}_{k+n}$ für eine absolute Korrektur $\mathbf{0}$, da die absolute Messung nur vom aktuellen Zustand abhängt:

$$\check{\mathbf{H}}_{k+n} = [\mathbf{0} \quad \mathbf{H}_{k+n}], \quad (5.6)$$

wobei $\mathbf{0}$ die Nullmatrix mit passender Dimension ist.

Während die Submatrix \mathbf{K}_k der augmentierten Kalman-Gain-Matrix bei der relativen Korrektur $\mathbf{0}$ war (vgl. Gleichung (3.29)), gilt dies für eine absolute Korrektur nicht mehr: \mathbf{K}_k kann nicht mehr zu $\mathbf{0}$ gesetzt werden, da die Messung, wie im vorherigen Abschnitt beschrieben, auch den geklonten Zustand beeinflusst.

Die Berechnung der augmentierten Systemkovarianzmatrix $\check{\mathbf{P}}_{k+n|k+n}$ für eine absolute Messung zum Zeitpunkt $k+n$ erfolgt wieder nach der Joseph-Form:

$$\check{\mathbf{P}}_{k+n|k+n} = \left(\mathbf{I} - \check{\mathbf{K}}_{k+n} \check{\mathbf{H}}_{k+n} \right) \check{\mathbf{P}}_{k+n|k+n-1} \left(\mathbf{I} - \check{\mathbf{K}}_{k+n} \check{\mathbf{H}}_{k+n} \right)^\top + \check{\mathbf{K}}_{k+n} \mathbf{R}_{k+n} \check{\mathbf{K}}_{k+n}^\top, \quad (5.7)$$

mit der Identitätsmatrix \mathbf{I} und der Messfehlerkovarianz \mathbf{R}_{k+n} .

$\check{\mathbf{K}}_{k+n} \check{\mathbf{H}}_{k+n}$ wird mit der augmentierten Kalman-Gain-Matrix nach Gleichung (3.29) und der augmentierten Messmatrix nach Gleichung (5.6)

berechnet:

$$\check{\mathbf{K}}_{k+n}\check{\mathbf{H}}_{k+n} = \begin{bmatrix} \mathbf{K}_k \\ \mathbf{K}_{k+n} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{H}_{k+n} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{K}_k \mathbf{H}_{k+n} \\ \mathbf{0} & \mathbf{K}_{k+n} \mathbf{H}_{k+n} \end{bmatrix}, \quad (5.8)$$

woraus

$$\mathbf{I} - \check{\mathbf{K}}_{k+n}\check{\mathbf{H}}_{k+n} = \begin{bmatrix} \mathbf{I} & -\mathbf{K}_k \mathbf{H}_{k+n} \\ \mathbf{0} & \mathbf{I} - \mathbf{K}_{k+n} \mathbf{H}_{k+n} \end{bmatrix} \quad (5.9)$$

folgt.

Aus Gleichung (5.9) wird ersichtlich, dass die linke obere Submatrix von $\check{\mathbf{P}}$ durch eine absolute Korrektur verändert wird, und das Einsetzen von Gleichung (5.9) in Gleichung (5.7) ergibt

$$\check{\mathbf{P}}_{k+n|k+n} = \begin{bmatrix} \mathbf{P}_{gg} & \mathbf{P}_{ga} \\ \mathbf{P}_{ag} & \mathbf{P}_{aa} \end{bmatrix} \quad (5.10)$$

mit den Teilmatrizen von $\check{\mathbf{P}}_{k+n|k+n}$, welche für eine bessere Nachvollziehbarkeit durch ihre Indizes „g“ für den geklonten Zustand und „a“ für den aktuellen Zustand gekennzeichnet sind:

$$\begin{aligned} \mathbf{P}_{gg} &= \mathbf{\Psi}_{gg} - \mathbf{K}_k \mathbf{H}_{k+n} \mathbf{\Psi}_{ag} - \mathbf{\Psi}_{ga} (\mathbf{K}_k \mathbf{H}_{k+n})^\top \\ &\quad + \mathbf{K}_k \mathbf{H}_{k+n} \mathbf{\Psi}_{aa} (\mathbf{K}_k \mathbf{H}_{k+n})^\top + \mathbf{K}_k \mathbf{R}_{k+n} \mathbf{K}_k^\top, \end{aligned} \quad (5.11)$$

$$\mathbf{P}_{ag} = (\mathbf{I} - \mathbf{K}_{k+n} \mathbf{H}_{k+n}) (\mathbf{\Psi}_{ag} - \mathbf{\Psi}_{aa} (\mathbf{K}_k \mathbf{H}_{k+n})^\top) + \mathbf{K}_{k+n} \mathbf{R}_{k+n} \mathbf{K}_k^\top, \quad (5.12)$$

$$\mathbf{P}_{ga} = (\mathbf{\Psi}_{ga} - \mathbf{K}_k \mathbf{H}_{k+n} \mathbf{\Psi}_{aa}) (\mathbf{I} - \mathbf{K}_{k+n} \mathbf{H}_{k+n})^\top + \mathbf{K}_k \mathbf{R}_{k+n} \mathbf{K}_{k+n}^\top, \quad (5.13)$$

$$\mathbf{P}_{aa} = (\mathbf{I} - \mathbf{K}_{k+n} \mathbf{H}_{k+n}) \mathbf{\Psi}_{aa} (\mathbf{I} - \mathbf{K}_{k+n} \mathbf{H}_{k+n})^\top + \mathbf{K}_{k+n} \mathbf{R}_{k+n} \mathbf{K}_{k+n}^\top; \quad (5.14)$$

unter Verwendung der Submatrizen $\mathbf{\Psi}$ der prädierten augmentierten Systemkovarianzmatrix

$$\check{\mathbf{P}}_{k+n|k+n-1} = \begin{bmatrix} \mathbf{\Psi}_{gg} & \mathbf{\Psi}_{ga} \\ \mathbf{\Psi}_{ag} & \mathbf{\Psi}_{aa} \end{bmatrix}. \quad (5.15)$$

Anhand der Gleichungen (5.11) bis (5.14) wird deutlich, dass die augmentierten Systemkovarianzmatrix $\check{\mathbf{P}}$ nicht wie beim impliziten Klonen mittels eines einfachen Produkts wie in Gleichung (3.24) mit Gleichung (3.23) zum Zeitpunkt $k + m$ rekonstruiert werden kann und sie somit explizit durch alle Filterschritte zu propagieren ist.

Dadurch, dass \mathbf{K}_k nicht mehr $\mathbf{0}$ ist, ist der Klon der zur relativen Messung gehörenden Zustandsschätzung $\hat{\mathbf{x}}_{k|k}$ nicht mehr statisch, wie in Gleichung (3.20) beschrieben, sondern wird mit der oberen Submatrix von $\check{\mathbf{K}}_{k+n}$ aktualisiert und damit zu

$$\hat{\mathbf{x}}_{k|k+n} = \hat{\mathbf{x}}_{k|k+n-1} + \mathbf{K}_k \mathbf{y}_{k+n}. \quad (5.16)$$

D. h., $\hat{\mathbf{x}}_{k|k+n}$ ist die geklonte Zustandsschätzung zum Zeitpunkt k mit allen Informationen bis zum Zeitpunkt $k + n$.

Beim Korrekturschritt der relativen Messungen mittels stochastischem Klonen können weiterhin die Teilmatrizen wie in den Gleichungen (3.31) und (3.32) verwendet werden, da der geklonte Zustand danach keine weitere Verwendung findet und ein neuer Klon erzeugt wird.

Abbildung 5.5 zeigt eine hybride Darstellung des expliziten stochastischen Klonens mit absoluten (\mathbf{z}_k) und relativen ($\mathbf{z}_{k,k+m}$) Messungen als DBN im oberen Teil und den Schätzungen im unteren Teil, getrennt durch eine gestrichelte horizontale Linie. Dabei sind die geklonten Zustandsschätzungen lila eingefärbt. Hier wird die Zustandsschätzung $\hat{\mathbf{x}}_{1|1}$ nach der Integration der Messung \mathbf{z}_1 geklont und zum Zeitpunkt $k = 2$ durch die absolute Messung \mathbf{z}_2 zu $\hat{\mathbf{x}}_{1|2}$ aktualisiert. Zusammen mit der relativen Messung $\mathbf{z}_{1,3}$ erfolgt dann zum Zeitpunkt 3 die Aktualisierung zu $\hat{\mathbf{x}}_{3|3}$. Der alte Klon wird hierdurch obsolet und es wird die aktuelle Zustandsschätzung $\hat{\mathbf{x}}_{3|3}$ geklont.

In Abbildung 5.6 ist in einer alternativen Darstellung mit räumlichen Bezügen der Einfluss des expliziten Klonens illustriert. Die Zustandsschätzung $\hat{\mathbf{x}}_{1|1}$ ist gestreift dargestellt, um darzustellen, dass die zu diesem Zeitpunkt aktuelle Zustandsschätzung und ihr Klon identisch sind. Wie in

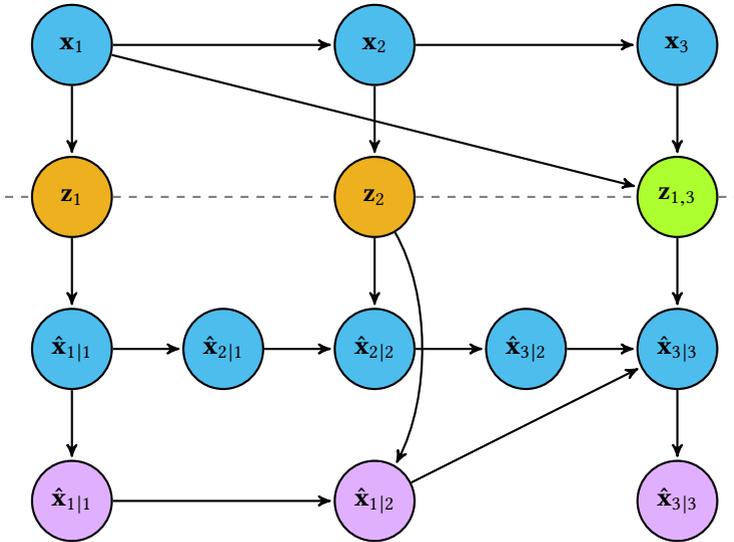


Abbildung 5.5: Hybride Darstellung des expliziten stochastischen Klonens mit absoluten (z_k) und relativen ($z_{k,k+m}$) Messungen als DBN im oberen Teil und den Schätzungen im unteren Teil, getrennt durch eine gestrichelte Linie.

Gleichung (5.16) wird durch die absolute Messung z_2 sowohl $\hat{x}_{2|1}$ zu $\hat{x}_{2|2}$ aktualisiert als auch die geklonte Zustandsschätzung $\hat{x}_{1|1}$ zu $\hat{x}_{1|2}$. Hierdurch wird bei der Aktualisierung mit der relativen Messung $z_{1,3}$ von $\hat{x}_{3|2}$ zu $\hat{x}_{3|3}$ durch den Bezug auf $\hat{x}_{1|2}$ eine geringere Drift erzeugt als in Abbildung 5.4.

5.2.3 Partielles explizites Klonen

In Abschnitt 3.1.3 wurde erwähnt, dass auch Ansätze existieren, welche explizites Klonen nur mit dem relevanten Teil der Zustandsschätzung durchführen, weshalb sie im Folgenden als partielles explizites Klonen bezeichnet werden sollen. Hierzu existieren Veröffentlichungen, welche

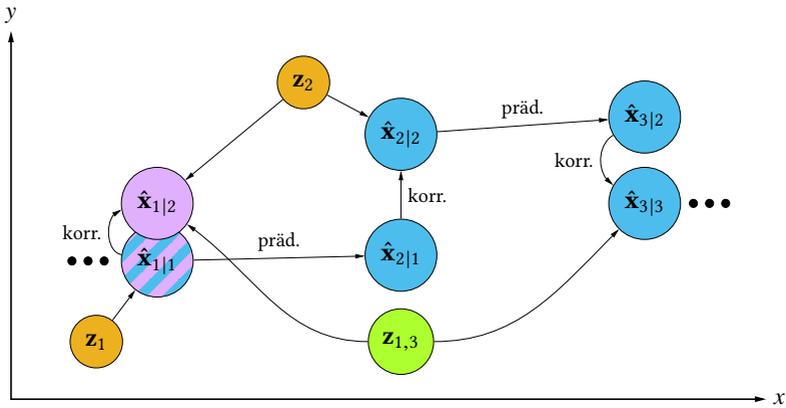


Abbildung 5.6: Illustration von sich abwechselnden absoluten (z_k) und relativen ($z_{k,k+m}$) Messungen mit explizitem stochastischem Klonen.

jedoch keine Analyse beinhalten, welche Auswirkungen die Verwendung des partiellen gegenüber dem kompletten Klonen hat [Chi11; Kle11; Ste12].

Zum besseren Verständnis, welche Implikationen das partielle Klonen gegenüber dem kompletten Klonen für die Zustandsschätzung hat, sollen hier die einzelnen Schritte im Kalman-Filter anhand des kompletten expliziten Klonens dargestellt werden. Zur besseren Übersichtlichkeit wird auf die Notation mit der diskreten Zeit k verzichtet, stattdessen werden Bezeichner für die Submatrizen eingeführt, um diese eindeutig zu identifizieren. Die Bezeichner entsprechen der aktuellen Zustandsschätzung a , der relevanten geklonten Zustandsschätzung c und des ungenutzten geklonten Teils u , welcher beim partiellen expliziten Klonen nicht geklont wird.

Die augmentierte Systemkovarianzmatrix wird dann zu

$$\check{\mathbf{P}} = \begin{bmatrix} \mathbf{P}_{uu} & \mathbf{P}_{uc} & \mathbf{P}_{ua} \\ \mathbf{P}_{cu} & \mathbf{P}_{cc} & \mathbf{P}_{ca} \\ \mathbf{P}_{au} & \mathbf{P}_{ac} & \mathbf{P}_{aa} \end{bmatrix}. \quad (5.17)$$

Im Prädiktionsschritt bleiben die geklonten Teile der Zustandsschätzung nach Gleichung (3.20) unverändert und die augmentierte Zustandsübergangsmatrix $\tilde{\mathbf{F}}$ hat die Form

$$\tilde{\mathbf{F}} = \begin{bmatrix} \mathbf{I}_{uu} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{cc} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F} \end{bmatrix}, \quad (5.18)$$

siehe Gleichung (3.22). Dabei sind \mathbf{I}_{uu} und \mathbf{I}_{cc} Einheitsmatrizen mit den entsprechenden Größen.

Bei der Berechnung der prädizierten Systemkovarianzmatrix durch

$$\check{\check{\mathbf{P}}}\check{\check{\mathbf{F}}} + \check{\check{\mathbf{Q}}} = \begin{bmatrix} \mathbf{P}_{uu} & \mathbf{P}_{uc} & \mathbf{P}_{ua}\mathbf{F}^T \\ \mathbf{P}_{cu} & \mathbf{P}_{cc} & \mathbf{P}_{ca}\mathbf{F}^T \\ \mathbf{F}\mathbf{P}_{au} & \mathbf{F}\mathbf{P}_{ac} & \mathbf{F}\mathbf{P}_{aa}\mathbf{F}^T \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q} \end{bmatrix} \quad (5.19)$$

zeigt sich, dass der ungenutzte Teil weder in den Teil der aktuellen Zustandsschätzung noch in den Teil des relevanten geklonten Teils eingeht, da nur der Block

$$\begin{bmatrix} \mathbf{P}_{cc} & \mathbf{P}_{ca}\mathbf{F}^T \\ \mathbf{F}\mathbf{P}_{ac} & \mathbf{F}\mathbf{P}_{aa}\mathbf{F}^T \end{bmatrix} \quad (5.20)$$

beim partiellen expliziten Klonen vorhanden ist.

Bei dem Korrekturschritt des relativen Sensors mittels stochastischem Klonen werden die Elemente ebenfalls mit den genannten Bezeichnern versehen. Die augmentierte Messmatrix hat die Form

$$\check{\check{\mathbf{H}}} = \begin{bmatrix} \mathbf{H}_u & \mathbf{H}_c & \mathbf{H}_a \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{H}_c & \mathbf{H}_a \end{bmatrix}, \quad (5.21)$$

da der ungenutzte Teil gerade darüber definiert ist, dass er von der relativen Messung unberührt bleibt und somit $\mathbf{H}_u = \mathbf{0}$ gilt. Im Umkehrschluss bedeutet dies, dass die zum relevanten geklonten Teil korrespondierende Submatrix \mathbf{H}_c gerade dem Teil entspricht, an dem die linke Hälfte der Messmatrix nicht $\mathbf{0}$ ist.

Die Elemente der augmentierten Kalman-Gain-Matrix nach Gleichung (3.29) ändern sich mit den neuen Bezeichnern von \mathbf{K}_{k+m} zu \mathbf{K}_a und \mathbf{K}_c wird aufgeteilt in den relevanten Teil \mathbf{K}_c und den ungenutzten Teil \mathbf{K}_u . Die Berechnung der augmentierten Kalman-Gain-Matrix ergibt sich dann zu

$$\check{\mathbf{K}} = \begin{bmatrix} \mathbf{K}_u \\ \mathbf{K}_c \\ \mathbf{K}_a \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{uu} & \mathbf{P}_{uc} & \mathbf{P}_{ua} \\ \mathbf{P}_{cu} & \mathbf{P}_{cc} & \mathbf{P}_{ca} \\ \mathbf{P}_{au} & \mathbf{P}_{ac} & \mathbf{P}_{aa} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{H}_c^\top \\ \mathbf{H}_a^\top \end{bmatrix} \check{\mathbf{S}}^{-1} = \begin{bmatrix} \mathbf{P}_{uc}\mathbf{H}_c^\top + \mathbf{P}_{ua}\mathbf{H}_a^\top \\ \mathbf{P}_{cc}\mathbf{H}_c^\top + \mathbf{P}_{ca}\mathbf{H}_a^\top \\ \mathbf{P}_{ac}\mathbf{H}_c^\top + \mathbf{P}_{aa}\mathbf{H}_a^\top \end{bmatrix} \check{\mathbf{S}}^{-1} \quad (5.22)$$

mit

$$\begin{aligned} \check{\mathbf{S}} &= \check{\mathbf{H}}\check{\mathbf{P}}\check{\mathbf{H}}^\top + \mathbf{R} \\ &= \mathbf{H}_c\mathbf{P}_{cc}\mathbf{H}_c^\top + \mathbf{H}_c\mathbf{P}_{ca}\mathbf{H}_a^\top + \mathbf{H}_a\mathbf{P}_{ac}\mathbf{H}_c^\top + \mathbf{H}_a\mathbf{P}_{aa}\mathbf{H}_a^\top + \mathbf{R}. \end{aligned} \quad (5.23)$$

In Gleichung (5.23) ist keine Teilmatrix mehr von dem ungenutzten Teil vorhanden und in Gleichung (5.22) lediglich in \mathbf{K}_u . Nach Gleichungen (3.31) und (3.32) geht jedoch nur \mathbf{K}_a in den Korrekturschritt mit ein, womit der ungenutzte Teil keinerlei Einfluss auf den Korrekturschritt mittels stochastischem Klonen hat.

Der Korrekturschritt mit der absoluten Messung ist ausführlich in Abschnitt 5.2.2 erläutert. Dort wird anhand der Gleichungen (5.11) bis (5.13) deutlich, dass hier durch vorhandene Kreuzkovarianzen in der Systemkovarianzmatrix auch der ungenutzte Teil einen Einfluss auf den geklonten Teil haben kann. D. h., auch wenn in dem Prädiktionsschritt und in der Korrektur mit dem relativen Sensor der ungenutzte Teil entfallen könnte, ist bei der Korrektur mit dem absoluten Sensor mit einem Unterschied zwischen dem partiellen und dem kompletten Klonen zu rechnen.

5.2.4 Implizites stochastisches Klonen mit Smoothing

Ein alternativer Ansatz, um die in dem von einer relativen Messung überspannten Zeitraum integrierten Messungen zum geklonten Zustand zurückzupropagieren, ist die Kombination mit Smoothing. Damit lassen

sich die Informationen rückwärts in der Zeit propagieren, womit der Vorteil, die augmentierte Systemkovarianzmatrix $\check{\mathbf{P}}_{k+m|k}$ wie beim impliziten Klonen bei Bedarf zu rekonstruieren erhalten bleibt [Emt18b].

Das Smoothing kann mit einem Rauch-Tung-Striebel-*smoother* (RTS-*smoother*) durchgeführt werden [Sär13]. Bei einem RTS-*smoother* werden die prädierte Zustandsschätzung und die zugehörige Systemkovarianzmatrix des Standard-Kalman-Filters

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F}_{k+1} \hat{\mathbf{x}}_{k|k}, \quad (5.24)$$

$$\mathbf{P}_{k+1|k} = \mathbf{F}_{k+1} \mathbf{P}_{k|k} \mathbf{F}_{k+1}^T + \mathbf{Q}_k \quad (5.25)$$

rückwärts in der Zeit propagiert. Die durch Smoothing geglättete Zustandsschätzung $\hat{\mathbf{x}}_{k|k+1}^s$ und die zugehörige geglättete Systemkovarianzmatrix $\mathbf{P}_{k|k+1}^s$ können rekursiv wie folgt berechnet werden:

$$\hat{\mathbf{x}}_{k|k+1}^s = \hat{\mathbf{x}}_{k|k} + \mathbf{A}_k (\hat{\mathbf{x}}_{k+1|k+2}^s - \hat{\mathbf{x}}_{k+1|k}), \quad (5.26)$$

$$\mathbf{P}_{k|k+1}^s = \mathbf{P}_{k|k} + \mathbf{A}_k (\mathbf{P}_{k+1|k+2}^s - \mathbf{P}_{k+1|k}) \mathbf{A}_k^T, \quad (5.27)$$

mit

$$\mathbf{A}_k = \mathbf{P}_{k|k} \mathbf{F}_{k+1}^T \mathbf{P}_{k+1|k}^{-1}. \quad (5.28)$$

Durch die Gleichungen (5.26) und (5.27) wird die Information der Zustandsschätzung zum Zeitpunkt $k + 1$ zur Zustandsschätzung zum Zeitpunkt k zurückpropagiert. Beim rekursiven Ausführen der Gleichungen vom Zeitpunkt $k + n$ aus wird zu Beginn $\hat{\mathbf{x}}_{k+n|k+n+1}^s = \hat{\mathbf{x}}_{k+n|k+n}$ und $\mathbf{P}_{k+n|k+n+1}^s = \mathbf{P}_{k+n|k+n}$ gesetzt.

Für eine relative Messung zwischen dem Zustand zum Zeitpunkt k und dem Zustand zum Zeitpunkt $k + m$ werden somit die Zustandsschätzungen in umgekehrter Reihenfolge rekursiv bis zum Zeitpunkt k durch Smoothing geglättet. Dadurch werden alle Informationen, welche in der Zwischenzeit durch andere Messungen integriert wurden, auf die Schätzung

$\hat{\mathbf{x}}_{k|k+1}^s$ und auch die zugehörige Systemkovarianzmatrix $\mathbf{P}_{k|k+1}^s$ zurückpropagiert. Hierbei ist $k+n = k+m-1$, da zum Zeitpunkt $k+m$, an dem das Smoothing durchgeführt wird, die aktuelle Zustandsschätzung die Prädiktion $\hat{\mathbf{x}}_{k+m|k+m-1}$ ist und noch keine Korrektur durchgeführt wurde.

Die augmentierte Zustandsschätzung wird dann zu

$$\check{\mathbf{x}}_{k+m|k+m-1} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k+1}^s \\ \hat{\mathbf{x}}_{k+m|k+m-1} \end{bmatrix}. \quad (5.29)$$

Durch das Smoothing kann zwar die Information der absoluten Messung zurückpropagiert werden, der Einfluss des Korrekturschritts auf den erweiterten Teil der augmentierten Systemkovarianzmatrix muss jedoch auch bei der Rekonstruktion Berücksichtigung finden. Unter der Annahme, dass die Information der absoluten Messungen durch das Smoothing vollständig in die geklonte Zustandsschätzung und dessen Submatrix der augmentierten Systemkovarianzmatrix eingebracht ist, wird wie beim relativen Korrekturschritt \mathbf{K}_k gleich $\mathbf{0}$ gesetzt, vgl. Abschnitt 3.1.3. Die Berechnung von $\check{\mathbf{K}}_{k+m}\check{\mathbf{H}}_{k+m}$ bei einer absoluten Messung zum Zeitpunkt $k+m$ wird dann zu

$$\check{\mathbf{K}}_{k+m}\check{\mathbf{H}}_{k+m} = \begin{bmatrix} \mathbf{0} \\ \mathbf{K}_{k+m} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{H}_{k+m} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{k+m}\mathbf{H}_{k+m} \end{bmatrix}, \quad (5.30)$$

woraus

$$\mathbf{I} - \check{\mathbf{K}}_{k+m}\check{\mathbf{H}}_{k+m} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - \mathbf{K}_{k+m}\mathbf{H}_{k+m} \end{bmatrix} \quad (5.31)$$

folgt.

Außerdem folgt aus $\mathbf{K}_k = \mathbf{0}$

$$\check{\mathbf{K}}_{k+m}\mathbf{R}_k\check{\mathbf{K}}_{k+m}^\top = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{k+m}\mathbf{R}_k\mathbf{K}_{k+m}^\top \end{bmatrix}. \quad (5.32)$$

Das Einsetzen von Gleichungen (5.31) und (5.32) in Gleichung (5.7) ergibt

$$\check{\mathbf{P}}_{k+m|k+m} = \begin{bmatrix} \mathbf{P}_{gg} & \mathbf{P}_{ga} \\ \mathbf{P}_{ag} & \mathbf{P}_{aa} \end{bmatrix} \quad (5.33)$$

mit

$$\mathbf{P}_{gg} = \mathbf{\Psi}_{gg}, \quad (5.34)$$

$$\mathbf{P}_{ag} = (\mathbf{I} - \mathbf{K}_{k+m}\mathbf{H}_{k+m})\mathbf{\Psi}_{ag}, \quad (5.35)$$

$$\mathbf{P}_{ga} = \mathbf{\Psi}_{ga}(\mathbf{I} - \mathbf{K}_{k+m}\mathbf{H}_{k+m})^T, \quad (5.36)$$

$$\mathbf{P}_{aa} = (\mathbf{I} - \mathbf{K}_{k+m}\mathbf{H}_{k+m})\mathbf{\Psi}_{aa}(\mathbf{I} - \mathbf{K}_{k+m}\mathbf{H}_{k+m})^T + \mathbf{K}_{k+m}\mathbf{R}_k\mathbf{K}_{k+m}^T; \quad (5.37)$$

unter Verwendung der Submatrizen $\mathbf{\Psi}$ der prädizierten augmentierten Systemkovarianzmatrix

$$\check{\mathbf{P}}_{k+m|k+m-1} = \begin{bmatrix} \mathbf{\Psi}_{gg} & \mathbf{\Psi}_{ga} \\ \mathbf{\Psi}_{ag} & \mathbf{\Psi}_{aa} \end{bmatrix}. \quad (5.38)$$

Die augmentierte Systemkovarianzmatrix $\check{\mathbf{P}}_{k+m|k+m}$ nach Gleichung (5.33) setzt sich dann aus den folgenden Teilmatrizen zusammen. Die linke obere Submatrix $\mathbf{P}_{gg} = \mathbf{\Psi}_{gg}$ bleibt wie beim ursprünglichen impliziten Klonen nach Abschnitt 5.2.1 von dem Korrekturschritt unberührt. Die rechte untere Submatrix \mathbf{P}_{aa} entspricht wieder gerade dem Korrekturschritt der normalen Systemkovarianzmatrix in der Joseph-Form. Die linke untere Submatrix \mathbf{P}_{ag} nach Gleichung (5.35) erhält man durch Multiplikation von $\mathbf{\Psi}_{ag}$ mit $(\mathbf{I} - \mathbf{K}_{k+m}\mathbf{H}_{k+m})$ von links und die rechte obere Submatrix \mathbf{P}_{ga} nach Gleichung (5.36) ist die zu ihr transponierte Matrix. Da hier nur eine Multiplikation stattfindet, kann analog zu Gleichung (3.24) ein modifiziertes kumulatives Produkt über mehrere aufeinanderfolgende Prädiktions-

und Korrekturschritte gebildet werden:

$$\begin{aligned}
 \mathcal{F}_{k+m,k} &= \mathbf{F}_{k+m} (\mathbf{I} - \mathbf{K}_{k+m-1} \mathbf{H}_{k+m-1}) \mathbf{F}_{k+m-1} \cdots \\
 &\quad (\mathbf{I} - \mathbf{K}_{k+2} \mathbf{H}_{k+2}) \mathbf{F}_{k+2} (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \mathbf{F}_{k+1} \\
 &= \mathbf{F}_{k+m} \prod_{i=1}^{m-1} (\mathbf{I} - \mathbf{K}_{k+i} \mathbf{H}_{k+i}) \mathbf{F}_{k+i}. \tag{5.39}
 \end{aligned}$$

Hierbei handelt es sich um eine vereinfachte Darstellung, bei der nur ein Korrekturschritt pro Prädiktion stattfindet. Bei mehreren Korrekturschritten können diese natürlich entsprechend mit eingefügt werden.

Die augmentierte Systemkovarianzmatrix für den Korrekturschritt der relativen Messung $\check{\mathbf{P}}_{k+m|k+m-1}$ kann folglich rekonstruiert werden. Die linke obere Submatrix $\mathbf{P}_{k|k+1}^s$ ist die durch Smoothing geglättete Systemkovarianzmatrix zum Zeitpunkt k , die rechte untere Submatrix entspricht der aktuellen Systemkovarianzmatrix und die beiden anderen Teilmatrizen ergeben sich durch Multiplikation der durch Smoothing geglätteten Systemkovarianzmatrix $\mathbf{P}_{k|k+1}^s$ mit dem modifizierten kumulativen Produkt nach Gleichung (5.39):

$$\check{\mathbf{P}}_{k+m|k+m-1} = \begin{bmatrix} \mathbf{P}_{k|k+1}^s & \mathbf{P}_{k|k+1}^s \mathcal{F}_{k+m,k}^\top \\ \mathcal{F}_{k+m,k} \mathbf{P}_{k|k+1}^s & \mathbf{P}_{k+m|k+m-1} \end{bmatrix}. \tag{5.40}$$

Anschließend kann der Korrekturschritt mit der relativen Messung analog zu dem impliziten Klonen nach Gleichung (3.25) bis Gleichung (3.32) durchgeführt werden. D. h., die augmentierte Systemkovarianzmatrix muss auch hier nur temporär für diesen Korrekturschritt rekonstruiert werden. Die Korrekturschritte mit den absoluten Sensoren können mit der einfachen Zustandsschätzung und der normalen Systemkovarianzmatrix durchgeführt werden.

Abbildung 5.7 zeigt eine hybride Darstellung des impliziten stochastischen Klonens mit Smoothing mit absoluten (\mathbf{z}_k) und relativen ($\mathbf{z}_{k,k+m}$) Messungen als DBN im oberen Teil und den Schätzungen im unteren Teil,

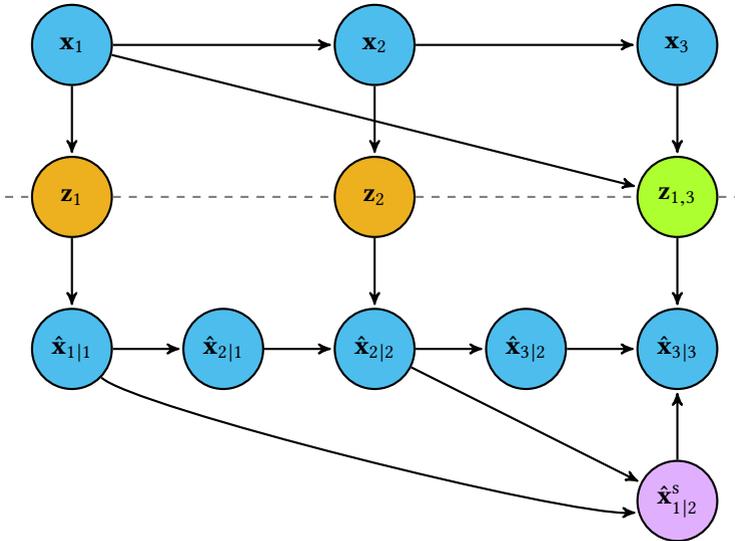


Abbildung 5.7: Hybride Darstellung des impliziten stochastischen Klonens mit Smoothing mit absoluten (z_k) und relativen ($z_{k,k+m}$) Messungen als DBN im oberen Teil und den Schätzungen im unteren Teil, getrennt durch eine gestrichelte Linie.

getrennt durch eine gestrichelte Linie. Hier wird die Zustandsschätzung $\hat{x}_{1|1}$ nicht explizit geklont, sondern zum Zeitpunkt 3 der benötigte Klon $\hat{x}_{1|2}^s$ durch Smoothing rekonstruiert. Zusammen mit der relativen Messung $z_{1,3}$ erfolgt dann die Aktualisierung zu $\hat{x}_{3|3}$.

In Abbildung 5.8 ist in einer alternativen Darstellung mit räumlichen Bezügen der Einfluss des impliziten Klonens mit zusätzlichem Smoothing illustriert. Das Smoothing von $\hat{x}_{k|k}$ zu $\hat{x}_{k|k+1}$ werden mit „sm.“ bezeichnet. Bei der Integration der relativen Messung $z_{1,3}$ wird zunächst das Smoothing durchgeführt, wodurch $\hat{x}_{1|1}$ zu $\hat{x}_{1|2}^s$ aktualisiert wird. Hierdurch wird die Information der absoluten Messung z_2 zu $\hat{x}_{1|2}^s$ zurückgeführt. Bei der Aktualisierung mit der relativen Messung $z_{1,3}$ von $\hat{x}_{3|2}$ zu $\hat{x}_{3|3}$ wird durch den Bezug auf $\hat{x}_{1|2}^s$ auch hier eine geringere Drift erzeugt als in Abbildung 5.4.

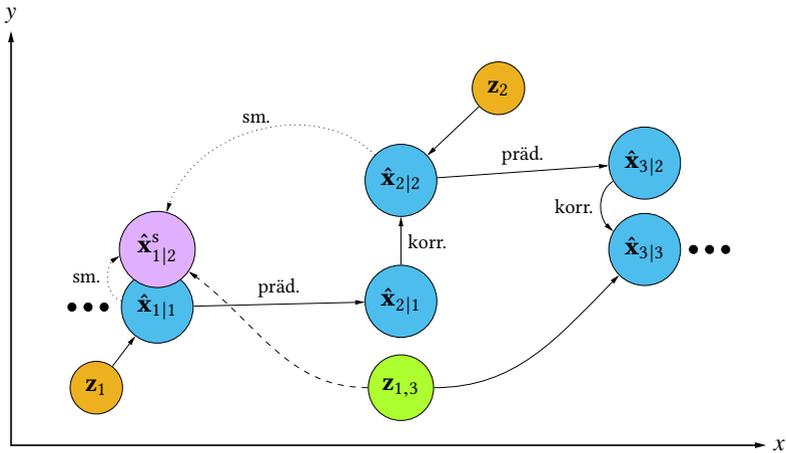


Abbildung 5.8: Illustration von sich abwechselnden absoluten (\mathbf{z}_k) und relativen ($\mathbf{z}_{k,k+m}$) Messungen mit implizitem stochastischem Klonen mit Smoothing.

Smoothing wird oft eingesetzt, um die aktuelle Schätzung mit einer konstanten Anzahl von zukünftigen Messungen zu verbessern, wobei die Ausgabe der aktuellen Schätzung dann entsprechend verzögert geschieht. Hierbei wird dann wegen der Verzögerung in der englischen Fachliteratur von *fixed-lag smoothing* gesprochen. Bei dem hier vorgestellten Ansatz mit der Kombination von implizitem Klonen und dem Smoothing kann letztere durchaus mit *fixed-lag smoothing* verglichen werden, da hier bzgl. der Zustandsschätzung $\hat{\mathbf{x}}_{k|k}$ mit der festgelegten Anzahl $m - 1$ an folgenden Messungen geglättet wird. Es soll jedoch darauf hingewiesen werden, dass hierdurch keine verzögerte Ausgabe der aktuellen Zustandsschätzung entsteht, da das Smoothing hier zum aktuellen Zeitpunkt $k + m$ zurück in der Zeit zu k erfolgt.

Der Ansatz mit explizitem Klonen kann für den Klon als *fixed-point smoother* bzgl. $\hat{\mathbf{x}}_{k|k}$ interpretiert werden [Sch19]. Bei dem *fixed-point smoother* wird der Ansatz verfolgt, dass ein Zustand zu einem festen Zeitpunkt geschätzt werden soll und auch alle zukünftigen Messungen zur Verbesse-

rung der Schätzung zu diesem Zeitpunkt beitragen [And79; Sim06]. Beim expliziten Klonen wird folglich mit der aktuellen Information aus einer absoluten Messung die zurückliegende geklonte Zustandsschätzung verbessert und damit die in Abschnitt 5.2.1 angesprochene Drift vermindert. Die fortlaufende Filterung mittels explizitem Klonen und absoluten Korrekturen kann somit als eine Kombination aus dem stochastischen Klonen und einem sich mitbewegenden *fixed-point smoother* bzgl. der geklonten Zustandsschätzung interpretiert werden. Der *fixed-point smoother* wird hier als sich mitbewegend bezeichnet, da der Bezugszeitpunkt immer nur zwischen k und $k + m$ fest ist, weil nach jedem abgeschlossenen relativen Korrekturschritt neu geklont wird und sich damit auch der Bezugszeitpunkt verschiebt.

5.2.5 Partielles implizites Klonen mit Smoothing

In Anlehnung an die in Abschnitt 3.1.3 erwähnte Methode des partiellen expliziten Klonens, bei dem nur der relevante Teil des Zustands geklont wird, kann auch bei dem im vorherigen Abschnitt vorgestellten impliziten Klonen mit Smoothing partiell geklont werden. Hierbei wird lediglich der von dem relativen Sensor betroffene Teil durch Smoothing geglättet und die rekonstruierte augmentierte Zustandsschätzung wird zu

$$\check{\mathbf{x}}_{k+m|k} = \begin{bmatrix} \hat{\mathbf{x}}_{c,k|k+1}^s \\ \hat{\mathbf{x}}_{k+m|k+m-1} \end{bmatrix}, \quad (5.41)$$

wobei $\hat{\mathbf{x}}_c^s$ den durch Smoothing geglätteten relevanten Teil der Zustandsschätzung $\hat{\mathbf{x}}$ darstellt.

Die erweiterte Systemkovarianzmatrix lässt sich nur approximativ rekonstruieren, da nur der relevante Teil durch Smoothing geglättet wird. Somit können diejenigen Teilmatrizen, welche die Kreuzkovarianzen zwischen der aktuellen Zustandsschätzung und dem geklonten Teil enthalten, nur näherungsweise berechnet werden. Die Teilmatrix $\mathbf{P}_{ac,k|k}$ wird mit dem Verhältnis aus dem geklonten Teil der Systemkovarianzmatrix $\mathbf{P}_{cc,k|k}$

und dem geglätteten geklonten Teil der Systemkovarianzmatrix $\mathbf{P}_{cc,k|k+1}^s$ gewichtet:

$$\mathbf{P}_{ac,k|k+1}^s = \mathbf{P}_{ac,k|k} [\mathbf{P}_{cc,k|k}]^{-1} \mathbf{P}_{cc,k|k+1}^s, \quad (5.42)$$

wobei es sich bei $\mathbf{P}_{ac,k|k}$ um eine $\dim(\hat{\mathbf{x}}) \times \dim(\hat{\mathbf{x}}_c)$ -Matrix handelt und a wieder für die aktuelle Zustandsschätzung steht. Durch diese Gewichtung entsprechen die Kreuzkovarianzen zwischen dem geklonten relevanten Teil und dem aktualisierten relevanten Teil in $\mathbf{P}_{ac,k|k+1}^s$ dem durch Smoothing geglätteten geklonten Teil der Systemkovarianzmatrix $\mathbf{P}_{cc,k|k+1}^s$.

Die approximativ rekonstruierte augmentierte Systemkovarianzmatrix ist dann

$$\check{\mathbf{P}}_{k+m|k+m-1} = \begin{bmatrix} \mathbf{P}_{cc,k|k+1}^s & \left[\mathbf{P}_{ac,k|k+1}^s \right]^\top \mathcal{F}_{k+m,k}^\top \\ \mathcal{F}_{k+m,k} \mathbf{P}_{ac,k|k+1}^s & \mathbf{P}_{k+m|k+m-1} \end{bmatrix}. \quad (5.43)$$

Während beim partiellen expliziten Klonen nur die Kreuzkovarianzen zwischen dem nicht geklonten Teil und der aktuellen Zustandsschätzung vernachlässigt werden, werden hier zusätzlich die Kreuzkovarianzen zwischen dem geklonten Teil der Zustandsschätzung und dem nicht geklonten Teil der aktuellen Zustandsschätzung approximiert.

5.2.6 Zusammenfassung

Es wurde eine ausführliche Analyse der theoretischen Zusammenhänge des stochastischen Klonens durchgeführt und die Voraussetzungen zur Anwendbarkeit des impliziten Klonens mit zusätzlichen absoluten Messungen dargestellt. Darauf aufbauend wurden das explizite Klonen und das davon abgeleitete partielle explizite Klonen eingehend analysiert.

Mit der neuartigen Methode des impliziten Klonens mit Smoothing für das Zurückpropagieren von Information können zusätzliche relative und absolute Messungen ebenfalls robust integriert werden. Sie behält dabei den Vorteil des ursprünglichen impliziten Klonens, dass die augmentierte Systemkovarianzmatrix bei Bedarf rekonstruiert werden kann und

nicht durch alle Filterschritte propagiert werden muss. Letzteres ist bei der in [Mou07] als mögliche Erweiterung vorgeschlagenen Methode des expliziten Klonens und den partiellen expliziten Ansätzen der Fall. Bei diesen Methoden muss sowohl die Zustandsschätzung als auch die Systemkovarianzmatrix explizit erweitert und durch alle Berechnungsschritte des Filters verarbeitet werden.

5.3 3D-Lokalisierung und -Kartierung

Für die Integration von Multi-Sensor-Fusion und SLAM wurde in Abschnitt 5.1 ein Verfahren für die 2D-Lokalisierung und -Kartierung mittels 2D-LiDAR vorgestellt. Dies ist für die Lokalisierung in Indoor-Umgebungen und Outdoor-Umgebungen mit flachem Gelände ausreichend und kann auch in unwegsamem Gelände noch bis zu einem gewissen Grad funktionieren. Wird der Untergrund allerdings zu uneben oder die Höhenunterschiede während der Aufnahme der Karte zu groß, ist eine 3D-Lokalisierung mit allen sechs Freiheitsgraden (6DoF) und die Kartierung mittels 3D-LiDAR notwendig.

5.3.1 3D-Lokalisierung

Für die 6DoF-Posenschätzung wurde ein auf dem in Abschnitt 5.2 vorgestellten SC-EKF basierendes Fusionsframework entwickelt, welches die Integration mehrerer absoluter und relativer Sensoren erlaubt. Das Filter ist modular aufgebaut und verarbeitet unter anderem die asynchronen Sensordaten einer inertialen Messeinheit (englisch: *inertial measurement unit* – IMU), welche aus drei Beschleunigungssensoren und drei Drehratensensoren besteht, einem GNSS, Rad-Odometrie und kann relative 3D-Posen mittels stochastischem Klonen integrieren.

Um lokale Diskontinuitäten aufgrund von geodätischen Projektionen zu vermeiden, wird die globale Position in geographischer Breite, geographische Länge und Höhe über dem Referenzellipsoid (WGS84) geschätzt

(englisch: *latitude, longitude, altitude* – LLA). Die Orientierung wird von einer Quaternion repräsentiert. Zusätzlich zu der 6DoF-Pose enthält der Zustand noch Meta-Variablen, wie z. B. systematische Fehler der IMU-Sensoren (Biase). Der Filterzustand wird im Fehlerzustandsraum formuliert (englisch: *error state space*), auch bekannt als indirektes Kalman-Filter [Rou99; Wen11]:

$$\Delta \mathbf{x} = [\Delta \mathbf{s}^T \ \Delta \Theta^T \ \Delta \mathbf{v}^T \ \Delta \mathbf{b}_\omega^T \ \Delta \mathbf{b}_a^T \ \Delta s_{\text{odo}}]^T, \quad (5.44)$$

mit dem 3D-Positionsfehler $\Delta \mathbf{s} = [\Delta x \Delta y \Delta z]^T$ in kartesischen Koordinaten, dem Lagefehler $\Delta \Theta = [\Delta \phi \Delta \theta \Delta \psi]^T$, dem 3D-Geschwindigkeitsfehler $\Delta \mathbf{v}$, den Bias-Fehlern der drei Drehratensensoren $\Delta \mathbf{b}_\omega$ und der drei Beschleunigungssensoren $\Delta \mathbf{b}_a$ der IMU sowie dem Skalenfaktorfehler für die Rad-Odometrie Δs_{odo} , was einen Zustandsvektor mit 16 Variablen ergibt. Bei der Formulierung im Fehlerzustandsraum werden nicht die Zustandsvariablen direkt geschätzt, sondern deren Fehler, weshalb diese Form auch als indirektes Filter bezeichnet wird. Dies hat für die Positionsschätzung den Vorteil, dass der Zustand zwar mit der geographischen Breite und Länge aufgebaut wird, die Korrektur jedoch in einem lokalen euklidischen Raum berechnet werden kann. Zum einen vermeidet man hierdurch, wie eingangs schon erwähnt, lokale Diskontinuitäten aufgrund von geodätischen Projektionen, zum anderen sind die Größenordnungen der einzelnen Zustandsvariablen nicht zu weit auseinander, was numerische Vorteile hat [Wen11]. Durch die Verwendung des Fehlerzustandsraum kann die Schätzung anhand von Eulerwinkel Fehlern erfolgen und dennoch die Orientierung durch eine Quaternion repräsentiert werden. Bei einer Quaternion kann es nicht wie bei der Repräsentation mittels Eulerwinkeln zu Singularitäten durch Mehrdeutigkeiten für einen Nick-Winkel von $\pm 90^\circ$ kommen, was auch als *gimbal-lock* bekannt ist [Wen11]. Gleichzeitig werden numerische Probleme in der Zustandsschätzung vermieden, welche bei der Verwendung der Quaternion durch deren Überbestimmtheit auftreten können [Lef82].

Im Prädiktionsschritt des Filters wird mit einem Strapdown-Algorithmus aus den Messungen der IMU eine 6DoF-Bewegung berechnet. Die Beschleunigungsmesser der IMU dienen noch zusätzlich durch eine Korrektur zur Stützung der horizontalen Lage, d. h. des Roll- und Nick-Winkels. Des Weiteren wird eine nicht-holonome Beschränkung durch einen Korrekturschritt mit einer virtuellen Messung hinzugefügt. Sie beruht auf dem kinematischen Modell des Fahrzeugs und trägt zu einer Stabilisierung quer zur Fahrtrichtung und in vertikaler Richtung bei. Die Messungen der Rad-Odometrie gehen als Vorwärtsgeschwindigkeit in eine Korrektur ein. Da sich das Abrollverhalten der Räder über deren Geometrie mit der Zeit und auch dynamisch bei Fahrmanövern ändern kann, wird hierfür ein zusätzlicher Skalenfaktor mitgeschätzt. Die Messungen des GNSS gehen in einen Korrekturschritt mit der absoluten 3D-Position und der 3D-Geschwindigkeit ein. Der Gierwinkel kann von keinem der genannten Sensoren direkt gemessen werden. Deshalb wird dieser in einem Vorverarbeitungsschritt aus vergangenen absoluten 2D-Positionsmessungen des GNSS geschätzt und geht anschließend als Korrektur ein. Alternativ kann der Gierwinkel mit den Daten aus einem digitalen Kompass oder einem GNSS mit zwei Antennen korrigiert werden, sofern diese zur Verfügung stehen. Die Messungen eines 3D-LiDARs gehen mittels paarweisem Scan-Matching aufeinanderfolgender Scans als relative 6DoF-Differenz mittels stochastischem Klonen in das Filter ein. Diese Art der Verwendung kann als LiDAR-Odometrie bezeichnet werden, analog zum Begriff visuelle Odometrie, welcher bei der Verwendung von Kameras für die Ermittlung einer relativen Bewegung üblich ist. Als Registrierungsverfahren wird der GICP-Algorithmus eingesetzt, da sich dieser speziell für das Scan-Matching von Scans mehrlageriger 3D-LiDAR-Sensoren eignet, siehe Abschnitt 2.2.2.

Durch die Einbeziehung der Scans des 3D-LiDARs über den GICP-Algorithmus als relative Bewegung in den SC-EKF wird keine explizite Karte aufgebaut, es handelt sich vielmehr um eine implizite Kartierung. Bei Graph-basierten SLAM-Methoden werden die Scans eines 3D-LiDARs ebenfalls paarweise zur Berechnung einer relativen Posendifferenz ver-

wendet. Somit gehen sie in den Graph nur als relative Bedingung zwischen zwei Knoten ein und können bei späteren Optimierungen korrigiert werden, was bei einer schritthaltend aufgebauten Karte nicht direkt möglich wäre. Bei einem Filter ist dieser Vorteil zwar nicht vorhanden, da keine zurückliegenden Posen korrigiert werden. Das paarweise Scan-Matching ist jedoch üblicherweise mit weniger Rechenaufwand verbunden als das Scan-Matching mit einer Gesamtkarte, welche sich stetig vergrößert. D. h., diese Form der impliziten Kartierung ist auch für die Kombination mit einem Filter durchaus sinnvoll.

5.3.2 3D-Kartierung

Für die 3D-Kartierung wird die Normal Distributions Transform (NDT) als Repräsentation verwendet [Mag09]. Bei der NDT-Karte handelt es sich um eine Voxelkarte, wobei für jedes belegte Voxel eine lokale Normalverteilung der darin enthaltenen Punkte geschätzt wird. Eine NDT-Karte hat einige Vorteile gegenüber einer reinen Belegtheitskarte, u. a. kann sie die Umgebung mit wesentlich größeren Voxeln adäquat repräsentieren, vgl. Abschnitt 3.2.4.

Für die kontinuierliche Integration neuer Sensordaten ist eine effiziente inkrementelle Updatemöglichkeit unerlässlich. Im Gegensatz zu [Tak06], wo eine Methode für inkrementelle Updates mit nicht-korrigierter Kovarianzschätzung vorgestellt wurde, wird hier analog zum ursprünglichen Ansatz ein effizientes Update mit einer Schätzung der korrigierten Stichprobenkovarianz eingesetzt, vgl. Gleichungen (3.54) und (3.55). Bei der inkrementellen Berechnung ergeben sich der geschätzte Mittelwert $\hat{\boldsymbol{\pi}}_l$ und die geschätzte Kovarianzmatrix $\hat{\boldsymbol{\Pi}}_l$ des l -ten Voxels für die Integration des m -ten Punkts \mathbf{p}_m nach [Tak06] wie folgt:

$$\hat{\boldsymbol{\pi}}_{l,m} = \frac{\boldsymbol{\rho}_{l,m}}{m}, \quad (5.45)$$

$$\hat{\boldsymbol{\Pi}}_{l,m} = \frac{\boldsymbol{\Upsilon}_{l,m} - \boldsymbol{\rho}_{l,m}\boldsymbol{\rho}_{l,m}^\top/m}{m-1}, \quad (5.46)$$

mit der inkrementellen Summe der Punktkoordinaten $\boldsymbol{\rho}_{l,m}$ und der inkrementellen Summe des dyadischen Produkts der Punkte $\boldsymbol{\Upsilon}_{l,m}$:

$$\boldsymbol{\rho}_{l,m} = \boldsymbol{\rho}_{l,m-1} + \mathbf{p}_m, \quad (5.47)$$

$$\boldsymbol{\Upsilon}_{l,m} = \boldsymbol{\Upsilon}_{l,m-1} + \mathbf{p}_m \mathbf{p}_m^\top. \quad (5.48)$$

Für die inkrementelle Berechnung werden hier somit zusätzlich die Summe der Punktkoordinaten $\boldsymbol{\rho}_{l,m}$ und die Summe der dyadischen Produkte der Punktkoordinaten $\boldsymbol{\Upsilon}_{l,m}$ benötigt. Da die Summe der dyadischen Produkte eine symmetrische 3×3 -Matrix mit sechs unterschiedlichen Koeffizienten ist, sind zusammen mit der Summe der Punktkoordinaten und der Anzahl der Punkte m insgesamt zehn zusätzliche Koeffizienten pro Voxel zu speichern. Eine wesentlich speichereffizientere Variante, bei der lediglich die Anzahl der Punkte m pro Voxel zusätzlich zu speichern ist, kann für die rekursive Integration einzelner Punkte \mathbf{p}_m aus [Saa13] hergeleitet werden:

$$\hat{\boldsymbol{\pi}}_{l,m} = \frac{(m-1)\hat{\boldsymbol{\pi}}_{l,m-1} + \mathbf{p}_m}{m}, \quad (5.49)$$

$$\hat{\mathbf{\Pi}}_{l,m} = \frac{m-2}{m-1} \hat{\mathbf{\Pi}}_{l,m-1} + \frac{1}{m} (\hat{\boldsymbol{\pi}}_{l,m-1} - \mathbf{p}_m)(\hat{\boldsymbol{\pi}}_{l,m-1} - \mathbf{p}_m)^\top. \quad (5.50)$$

Da nur zusätzliche Elemente pro Voxel zu speichern sind, hat auch die Karte mit inkrementellem Update einen Speicherbedarf, welcher mit der Kartengröße steigt und nicht mit der Anzahl der aufgenommenen Punkte bzw. der Kartierungsdauer.

In der NDT-Karte können beispielsweise durch verrauschte Sensordaten, eine fehlerhafte Lokalisierung oder auch durch dynamische Objekte fälschlicherweise Voxel instanziiert werden. Um diese auch wieder eliminieren zu können, ist ein Verfahren zur Einbeziehung von freiem Raum nach [Saa13] integriert. Hierfür wird zusätzlich jedem Voxel eine Existenzwahrscheinlichkeit äquivalent zur Belegtheitswahrscheinlichkeit bei den Belegtheitskarten hinzugefügt. Diese Existenzwahrscheinlichkeit wird erhöht, wenn ein Punkt in dieses Voxel fällt, und verringert, wenn ein Laser-

strahl das Voxel durchquert. Hierfür wird jeder Laserstrahl von seinem Ursprung mit dem Bresenham-Strahlverfolgungsalgorithmus bis zum Endpunkt durch die Voxel verfolgt [Bre65; Sch13]. Voxel werden erst instanziiert, wenn deren Existenzwahrscheinlichkeit über einen gewissen Schwellwert steigt. So kann verhindert werden, dass Voxel durch vereinzelte Beobachtungen instanziiert werden. Fällt die Existenzwahrscheinlichkeit hingegen unter einen gewissen Schwellwert, so wird dieses Voxel als unbelegt betrachtet. Es wird jedoch weiterhin aktualisiert, um sicherzustellen, dass keine Information verloren geht, sollte das Voxel bei ausreichender Evidenz für dessen Existenz wieder als belegt betrachtet werden.

5.3.3 Schleifenschluss

Das Schließen von Schleifen ist eine essentielle Eigenschaft von SLAM. Der GICP-Algorithmus kann bei der impliziten Kartierung nicht nur zum Scan-Matching zweier aufeinanderfolgender Scans, sondern auch zum Scan-Matching mit einem älteren Scan für einen Schleifenschluss verwendet werden. Hierfür wird für den aktuellen Scan zunächst durch die Bestimmung der euklidischen Distanz zu den zurückliegenden geschätzten Posen auf einen potentiellen Schleifenschluss überprüft. Im zweiten Schritt wird der GICP-Algorithmus mit dem aktuellen Scan und dem Kandidaten für eine Schleife durchgeführt. Die vom GICP-Algorithmus ausgegebene Bewertung für das Scan-Matching wird zusätzlich für die Verifikation oder Ablehnung der Schleife herangezogen.

Die relative Bewegung aus dem Scan-Matching zum Schließen der Schleife kann allerdings nicht als relatives Update mittels implizitem stochastischem Klonen integriert werden, da zwischen den beiden Zuständen eine zu lange Zeit bzw. ein zu langer Weg liegt und die Annahme der Linearisierung von \mathcal{F} nicht mehr gelten kann. Beim expliziten Klonen hätte die zurückliegende Zustandsschätzung geklont und mitpropagiert werden müssen. Da aber potentiell jeder Zustand zu einem Schleifenschluss gehören könnte, müssten alle Zustandsschätzungen als Klone mitgeführt

werden, was einen immensen Rechenaufwand bedeuten würde, da sich der Zustandsraum stetig vergrößert.

Da das GNSS in der in Abschnitt 5.3.1 vorgestellten Sensorkonfiguration der einzige absolute Positionssensor ist, hat es einen maßgeblichen Einfluss auf die geschätzte Position. In modernen GNSS-Empfängern können mittels zusätzlicher Korrekturservices die meisten Fehler weitestgehend korrigiert werden. Von den verbleibenden Fehlern ist die Mehrwegeausbreitung der dominierende und die durch sie verursachte Abweichung der Positionsschätzung kann immer noch mehrere Meter betragen [Bra01; Bra17]. Die Mehrwegeausbreitung ist stark von der Umgebung und der Satellitenkonstellation abhängig. Da letztere zeitvariant ist, können sich die Fehler an einem Ort über die Zeit ändern, d. h. auch von dem ersten Besuch eines Ortes bis zum Schleifenschluss. Um nun für den SC-EKF mit der impliziten Kartierung einen Schleifenschluss zu realisieren, wird eine zusätzliche Schätzung der Abweichung der GNSS-Positionsmessungen hinzugefügt. Für die Orientierung werden hingegen keine Abweichungen geschätzt, da diese hinreichend durch die Beschleunigungsmesser und den aus der Vorverarbeitung berechneten Gierwinkel stabilisiert sind. Der Zustandsfehleraum aus Gleichung 5.44 wird um den Fehler der 3D-GNSS-Abweichung $\Delta \mathbf{o}_{\text{GNSS}}$ erweitert:

$$\Delta \mathbf{x}_{\text{loop}} = [\Delta \mathbf{x}^T \Delta \mathbf{o}_{\text{GNSS}}^T]^T, \quad (5.51)$$

was dann einen Zustandsraum mit 19 Variablen ergibt. Die Abweichung geht beim Schleifenschluss durch die relative Position aus dem Scan-Matching zwischen dem aktuellen Scan und dem Scan des zurückliegenden Zustands als Messung in das Filter ein. Die Abweichung wird dann im Korrekturschritt des GNSS entsprechend mit einberechnet.

5.3.4 Asynchrone Multi-Sensor-Fusion

Das in Abschnitt 5.3.1 vorgestellte Filter ist durch seine modulare Filterstruktur in der Lage, verschiedene Sensorkombinationen zu verarbeiten.

Dabei können einzelne Sensoren auch im Betrieb zeitweise oder auch ganz ausfallen, ohne dass die Lokalisierung abgebrochen wird. Die Lokalisierung wird vielmehr nahtlos mit weniger Sensoren fortgesetzt. Die inhärente Kompensation des Ausfalls einzelner Sensoren geht jedoch unweigerlich mit einer gleichzeitigen Verschlechterung der Präzision einher, da weniger Informationen zur Verfügung stehen. Diese Art der Ausfalltoleranz kann auch unter dem Begriff *graceful degradation* zusammengefasst werden, da im Gegensatz zum plötzlichen Totalausfall eine Abnahme der erzielbaren Genauigkeit erfolgt [Gu16].

Mit dem Fusionsalgorithmus können außerdem die Daten von asynchronen Sensoren verarbeitet werden. Die asynchrone Sensordatenverarbeitung bezieht sich zum einen darauf, dass die einzelnen Sensoren unterschiedliche Datenraten aufweisen. Zum anderen kommen die Sensordaten selbst bei gleichen Datenraten nicht gleichzeitig an, da die Taktgeber der Sensoren separat laufen. Die Möglichkeit der Verarbeitung von asynchronen Sensordaten bietet den Vorteil, dass keine aufwändige Hardware-Synchronisation der einzelnen Sensoren notwendig ist. So können neue oder alternative Sensoren mit wenig Aufwand integriert werden, was vor allem bei prototypischen Forschungsplattformen ein immenser Vorteil ist [Emt10b].

Zeitstempelschätzung

Die einfachste Methode, um mehrere asynchrone Sensoren zu integrieren, ist deren direkte Verarbeitung bei Empfang der Daten im Rechner. Hierzu wird für die Prädiktion die spezifizierte Rate der IMU angenommen, da diese hier mittels Strapdown-Algorithmus integriert wird. Die Daten der anderen Sensoren werden der zum Zeitpunkt des Empfangs aktuellen Prädiktion zugeordnet, was zu einem zusätzlichen Fehler führt, welcher durch entsprechende Erhöhung der Messfehlerkovarianzen modelliert wird. Da die Rate der IMU meist mindestens eine Größenordnung über den Raten der anderen Sensoren liegt, sind die zusätzlichen, in der Praxis zu erwarten-

den Fehler zwar gering, eine Integration der Sensordaten mit dem genauen Zeitpunkt ist dennoch nach Möglichkeit anzustreben.

Für die Fusion von asynchronen Sensoren zum möglichst genauen Zeitpunkt bietet es sich an, die Sensordaten auf dem verarbeitenden Rechner mit Zeitstempeln zu versehen. Da die Sensoranbindung und -datenverarbeitung meist mit Rechnern durchgeführt wird, deren Betriebssystem keinen harten Echtzeitbedingungen genügt, können jedoch die Latenzen der Sensordaten schwanken. Die Latenz bezeichnet dabei die Zeit, welche von dem Zeitpunkt vergeht, an dem die Daten vom Sensor verschickt werden, bis sie im Speicher des Rechners verfügbar sind und mit einem Zeitstempel versehen werden.

Für die Verbesserung der Genauigkeit der Zeitstempel bietet sich eine Kombination des Taktgebers des Sensors mit der Uhr des verarbeitenden Rechners an. Während der Taktgeber des Sensors zwar sehr geringe Schwankungen in der Rate aufweist, kann er aufgrund der fehlenden Hardware-Synchronisation gegenüber der Rechneruhr und auch den Taktgebern der anderen Sensoren driften. Die Zeitstempel der Rechneruhr hingegen können als absolute Referenz dienen, da hier alle Sensordaten zusammenlaufen, sie weisen jedoch, wie angesprochen, vergleichsweise starke Schwankungen auf.

Für die Fusion des Taktgebers des Sensors und der Zeitstempel der Rechneruhr kommt ein Kalman-Filter zum Einsatz. Da es nur eine zu schätzende Größe – den Zeitstempel – gibt, hat dieses Kalman-Filter keine Matrizen, sondern beinhaltet nur skalare Operationen. Die Prädiktion erfolgt mit der Zeitstempeldifferenz aus dem Taktgeber des Sensors und in dem Korrekturschritt wird der Zeitstempel der Rechneruhr integriert.

Da die Zeitstempel somit sehr genau geschätzt sind, kann nun der Strapdown-Algorithmus in der Prädiktion im Lokalisierungsfilter mit dem aus den Zeitstempeln ermittelten Zeitintervall erfolgen. Dies erhöht vor allem die Genauigkeit, wenn durch Übertragungsfehler vereinzelt Daten der IMU verloren gehen und ohne Berücksichtigung der Zeitstempel zu

falschen Zeitintervallen bei der Propagation der Lage, Geschwindigkeit und Position führen würden.

Auch die Korrekturen können zum genaueren Zeitpunkt integriert werden. Anstatt sie der letzten Prädiktion zuzuordnen, kann die Prädiktion mit den letzten Sensordaten der IMU bis zum Zeitstempel des zu fusionierenden Sensors extrapoliert werden. Bei der nächsten Prädiktion wird das Zeitintervall für den Strapdown-Algorithmus mit den neuen Daten der IMU entsprechend verkürzt [Kle20].

Sind die in das Filter eingespeisten Zeitstempel stabilisiert und weniger Schwankungen unterworfen, so sind auch die Zeitstempel der vom Filter ausgegebenen Lokalisierung genauer. Dies hat auch für andere Algorithmen Vorteile, welche auf die Lokalisierung zurückgreifen, wie z. B. die inertielle Korrektur von LiDAR-Daten, welche durch die Eigenbewegung während der Rotation des LiDARs verzerrt werden. Für die inertielle Korrektur sollte die Lokalisierung bzw. Eigenbewegung zum einen möglichst genau geschätzt werden, zum anderen sollte die zeitliche Zuordnung zu den Einzelmessungen des LiDAR-Scans auch möglichst exakt sein.

Kombinierte Parallelverarbeitung

Die Verarbeitung der Messungen weist je nach Sensordatentyp durch mehr oder wenig aufwändige Vorverarbeitungs- oder Berechnungsschritte unterschiedliche Verarbeitungszeiten auf. Damit die Verarbeitung der Sensordaten in Echtzeit erfolgen kann, müssen die Verarbeitungszeiten jedes Sensors kleiner als die jeweilige Inverse der Rate dieses Sensors sein. Benötigt die Verarbeitung eines Sensors im Vergleich zu den anderen Sensoren jedoch sehr lange, wird durch die sequenzielle Filterung die Verarbeitung der folgenden Sensoren verzögert. Im Schnitt werden zwar weiche Echtzeitbedingungen eingehalten, es kommt dadurch jedoch zu periodischen Verzögerungen der Ausgabe der Lokalisierungsschätzung, was z. B. für einen nachgelagerten Pfad- oder Trajektorienregler nachteilig ist. Ideal wäre, dass die Verarbeitungszeit jedes Sensors kleiner ist als die Inverse der Rate des schnellsten Sensors. Ist dies nicht der Fall, kann durch eine Par-

allelverarbeitung dennoch eine hohe Ausgaberate erzielt werden [Emt14; Emt18b].

Hierzu werden zwei Filter parallel verarbeitet. In einem Filter werden alle Sensordaten verarbeitet, wodurch dessen Lokalisierungsschätzung periodisch verzögert zur Verfügung steht. Die Verzögerung ist in Abbildung 5.9 in der oberen Reihe durch die Verarbeitung der Daten dargestellt, welche mit dem breiten grünen Rechteck gekennzeichnet sind. Sensoren mit kurzen Verarbeitungszeiten sind als schmale orange Rechtecke dargestellt. Im zweiten Filter werden nur Sensordaten verarbeitet, deren Verarbeitungszeit kleiner als die Inverse der Rate des schnellsten Sensors ist. Es kann damit als Ausgabe-Filter für die Lokalisierungsschätzung mit konstant hoher Rate dienen, was in Abbildung 5.9 in der unteren Reihe durch die hellblauen Pfeile visualisiert ist. Da die Verzögerung des ersten Filters nur temporär ist, kann es nach der Verzögerung wieder aufholen und dessen Zustandsschätzung auf das zweite Filter übertragen werden, was in Abbildung 5.9 durch die roten Pfeile dargestellt ist. Bei der Übertragung wird die Zustandsschätzung des zweiten Filters ersetzt, sodass es zu keiner Mehrfachberücksichtigung der Sensordaten, welche in beiden Filtern verarbeitet werden, kommt. Hierdurch kann das zweite Filter die Lokalisierungsschätzung mit einer möglichst konstanten Rate ausgeben und durch die Übertragung dennoch die Information von Sensoren integrieren, welche eigentlich zu einer Verzögerung führen würden. Bedingung für die Übertragung des Zustands ist allerdings, dass die benötigte kumulierte Rechenzeit für die Integration aller Sensoren im ersten Filter gering genug ist, um wieder auf das zweite Filter aufholen zu können.

5.3.5 Zusammenfassung

Zur Implementierung der Verfahren für die reale Anwendung der vorgestellten Filter-Methoden für die 6DoF-Lokalisierung und Kartierung dient ein Filter-Framework auf Basis eines EKF mit stochastischem Klonen. Durch die modulare Struktur können verschiedene Sensorkonfigurationen

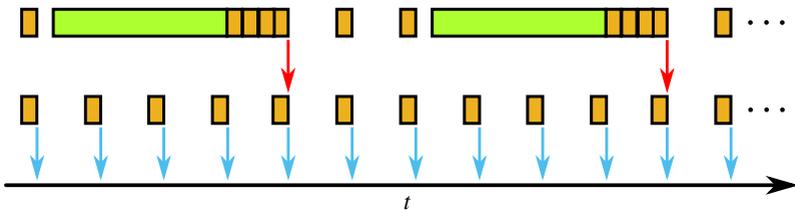


Abbildung 5.9: Schematische Darstellung der parallelen Verarbeitung. Filter mit konstanter Ausgaberate in der unteren Reihe (hellblaue Pfeile). Filter mit Verzögerungen durch lange Berechnungszeiten in der oberen Reihe. Übertragung des Filterzustands als rote Pfeile.

und Algorithmen kombiniert und damit auch evaluiert und miteinander verglichen werden.

Für die 3D-Kartierung wird eine NDT-Karte unter Berücksichtigung der Existenzwahrscheinlichkeit pro Voxel eingesetzt und kann bei der Evaluierung für qualitative Vergleiche herangezogen werden.

Da bei den Filter-Methoden mit stochastischem Klonen keine explizite Kartierung erfolgt, findet für das Schließen von Schleifen eine neuartige Methode mit zusätzlicher Schätzung eines GNSS-Offsets Anwendung.

Bei dem Einsatz auf Systemen ohne Echtzeitbetriebssystem kann die Schwankung der Latenzen beim Einlesen der Sensorwerte durch eine Zeitstempelschätzung verringert werden und eine Parallelverarbeitung erlaubt die Ausgabe der Lokalisierungsschätzung mit hoher Rate, auch wenn einzelne Berechnungen eine längere Verarbeitungsdauer aufweisen.

Kapitel 6

Ergebnisse und Auswertung

Die in den vorherigen Abschnitten vorgestellten Methoden und Verfahren wurden alle als Software implementiert und mit realen Sensordaten evaluiert. Die Umsetzung erfolgte als plattformunabhängige C++-Bibliotheken, um sowohl eine reproduzierbare Evaluation mit aufgenommenen Daten zu ermöglichen als auch den Einsatz auf prototypischen mobilen Robotersystemen für die autonome Navigation zu realisieren.

Bei der Umsetzung wurde auf einige Open-Source-Bibliotheken zurückgegriffen. Die Implementierung der Partikel- und Kalman-Filter wurde unter Verwendung von Eigen realisiert [Eigen]. Eigen ist eine Bibliothek für lineare Algebra und eignet sich besonders für Rechenoperationen mit Matrizen und Vektoren. Zur Verarbeitung der 3D-LiDAR-Scans wurde die Point Cloud Library (PCL) verwendet [Rus11]. Die PCL wurde für die Verarbeitung von Punktwolken entwickelt und bietet bereits eine Vielzahl von Algorithmen, wie z. B. den GICP-Algorithmus für das Scan-Matching. Darüber hinaus lässt sie sich mit eigenen Punkttypen und Verarbeitungsalgorithmen erweitern und findet daher im Forschungsumfeld regen Einsatz.

So wurde auch die Umsetzung der inkrementellen NDT-Karte mittels dieser Bibliothek entwickelt.

Für den Einsatz auf mobilen Robotersystemen wurden die entwickelten Algorithmen als Module für das Robot Operating System (ROS) umgesetzt. Bei ROS handelt es sich um ein Meta-Betriebssystem für Roboter, welches sich in der Robotikforschung als Quasi-Standard etabliert hat [ROS]. Es bietet eine Infrastruktur für die Kommunikation zwischen verschiedenen Softwaremodulen und eignet sich daher sehr gut für die Integration komplexer modularer Systeme [Qui09].

6.1 Erweiterte Landmarken

Die Daten für die Evaluation des in Abschnitt 4.1 vorgestellten SLAM-Algorithmus mit den erweiterten Landmarken stammen von einer Powerbot-Plattform der Firma MobileRobots. Bei dieser Plattform handelt es sich um einen mobilen Roboter mit differentiellm Antrieb, welcher mit Rad-Odometrie-Sensoren ausgestattet ist. Zusätzlich sind ein 2D-LiDAR SICK LMS 200 und eine Kamera PointGrey Flea mit einer Auflösung von 640×480 Pixeln für die Landmarkenextraktion installiert. Die Kamera ist mit einem 1/3-Zoll-Sensor und einem 8-mm-Objektiv ausgestattet, was einen Öffnungswinkel von etwa $33,3^\circ$ ergibt. Dies bedeutet, dass die Kamera eine horizontale Auflösung von ca. $0,05^\circ$ pro Pixel hat, was in etwa einem Zwanzigstel der Winkelinkremente des Laserscanners mit 1° entspricht.

Wie in Abschnitt 4.1 beschrieben, handelt es sich bei dem SLAM-Algorithmus um eine Implementierung nach FastSLAM 2.0 und als Landmarkenmodelle dienen vertikale Strukturen. In dem Testgebiet sind mehrere Laternenpfähle und Bäume vorhanden, die in dieses Landmarkenschema passen. Da keine Referenzlokalisierung oder -karte vorhanden ist, ist die folgende Evaluation vergleichender Art.

6.1.1 Evaluation

Der SLAM-Algorithmus ist flexibel aufgebaut, sodass sowohl das etablierte einfache Punktlandmarkenmodell als auch das neuartige um ortsunabhängige Merkmale erweiterte Modell mit unterschiedlichen Partikelanzahlen miteinander verglichen werden können. Für die Evaluation wurden diese Konfigurationen verwendet:

1. 1 Partikel mit erweitertem Modell,
2. 1 Partikel mit Punktlandmarkenmodell,
3. 25 Partikel mit erweitertem Modell,
4. 25 Partikel mit Punktlandmarkenmodell.

Da es sich um eine Implementierung nach FastSLAM 2.0 handelt und damit die Beobachtungen auch in die vorgeschlagene Verteilung mit eingehen, kann das Filter auch nur mit einem Partikel konvergieren [Mon03b].

Der befahrene Pfad hat in etwa eine Länge von 100 m und es sind drei Schleifen auf der Strecke zu schließen. In dem Areal wurden 12 verschiedene Objekte als Landmarken erkannt. Darunter befinden sich Baumstämme, Laternen und zwei zusätzlich vertikal aufgestellte Rohre. Das kombinierte Extraktionsverfahren mit LiDAR und Kamera ist darauf ausgelegt, möglichst wenig Fehldetektionen zu liefern, und so wurden nur in ca. 20 % der Daten Landmarken erkannt. Hierbei spielt natürlich auch der relativ kleine Öffnungswinkel der Kamera eine Rolle, da dadurch nur selten Landmarken überhaupt beobachtbar sind. Das kombinierte Extraktionsverfahren ist sehr robust, es gab nur eine einzige Fehldetektion, bei der ein Busch einmalig fälschlicherweise als Landmarke erkannt wurde. Vor allem die Messung des Landmarkenradius ist sehr genau mit einer durchschnittlichen Abweichung des geschätzten Radius über alle Landmarken von 7,3 mm. Der jeweilige Referenzradius wurde über den per Messung mit einem Maßband bestimmten Umfang berechnet.

Die folgenden Abbildungen zeigen die Ergebnisse der Evaluation am Ende der jeweiligen Kartierung. Der mobile Roboter wurde von links nach rechts gefahren und in drei Schleifen wieder zurück. Sein Pfad ist in Hellblau und seine Endposition mit einem Kreis dargestellt. Die Landmarken werden durch ihre für die Darstellung vergrößerten σ -Ellipsen in Grün repräsentiert und sind durchnummeriert. Im Falle von mehreren Partikeln ist die Position und die Karte des besten Partikels dargestellt, da davon ausgegangen werden kann, dass das Partikel mit dem höchsten Gewicht auch die beste Karte besitzt.

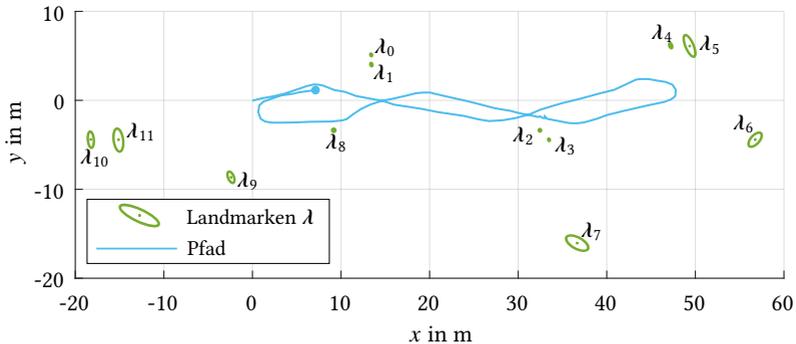


Abbildung 6.1: Kartierungsergebnis der Konfiguration mit einem Partikel und dem erweiterten Landmarkenmodell. Der Pfad ist in Hellblau dargestellt und die Landmarken λ in Grün.

In Abbildung 6.1 ist das Ergebnis der Konfiguration mit einem Partikel und dem erweiterten Landmarkenmodell dargestellt. In der Karte sind genau 12 Landmarken kartiert worden, wobei die Landmarken mit kleinen Fehlerellipsen deutlich öfters beobachtet wurden als die mit den größeren Ellipsen. Landmarke λ_7 ist die oben angesprochene Fehldetektion, bei der ein Busch beobachtet wurde.

Abbildung 6.2 zeigt das Ergebnis mit ebenfalls nur einem Partikel, jedoch ohne erweitertes Landmarkenmodell, was bedeutet, dass die Landmarken nur über ihren Ort definiert sind. Es sind insgesamt 16 Landmarken kartiert worden, wobei vier Landmarken fälschlicherweise neu

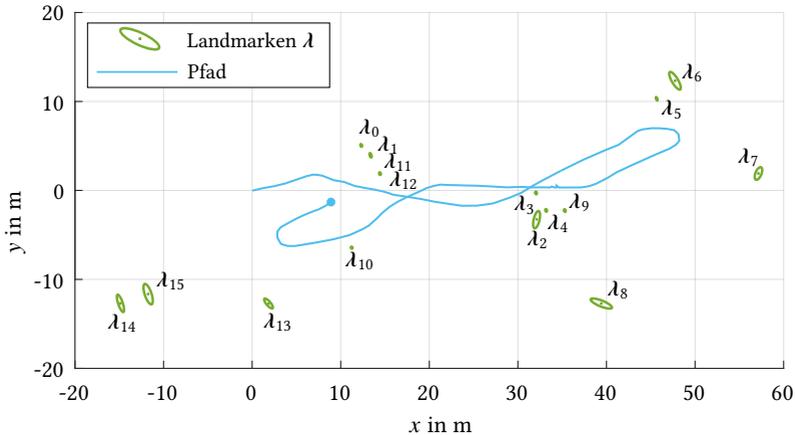


Abbildung 6.2: Kartierungsergebnis der Konfiguration mit einem Partikel und dem einfachen Punktlandmarkenmodell. Der Pfad ist in Hellblau dargestellt und die Landmarken λ in Grün.

instanziiert worden sind, da sie nicht zugeordnet werden konnten. Außerdem traten auch Fehlzuordnungen auf, was aus der Karte jedoch nicht ersichtlich ist. So wurde z. B. direkt am Anfang Landmarke λ_1 nicht gleich instanziiert, sondern deren Beobachtung zunächst mehrmals der nahe bei ihr liegenden Landmarke λ_0 zugeordnet. Im späteren Verlauf konnte die erste Schleife nicht erfolgreich geschlossen werden, was an der zusätzlichen Landmarke λ_9 zu erkennen ist. Auch die zweite Schleife konnte nicht erfolgreich geschlossen werden und es wurden durch den großen Fehler in der Posenschätzung die Beobachtungen den Landmarken λ_0 und λ_1 nicht korrekt zugeordnet, sondern als Landmarken λ_{11} und λ_{12} fälschlicherweise neu instanziiert. Die letzte Schleife wurde mit den Landmarken λ_{11} und λ_{12} geschlossen anstatt λ_0 und λ_1 . Letztendlich ist die erzeugte Karte aufgrund der Fehlzuordnungen und zusätzlichen Landmarken inkonsistent.

Das Ergebnis von 25 Partikeln mit erweitertem Modell wird hier nicht gezeigt, da in der Karte keine Unterschiede zu der Variante mit nur einem Partikel in Abbildung 6.1 zu erkennen sind. Abbildung 6.3 zeigt das Ergeb-

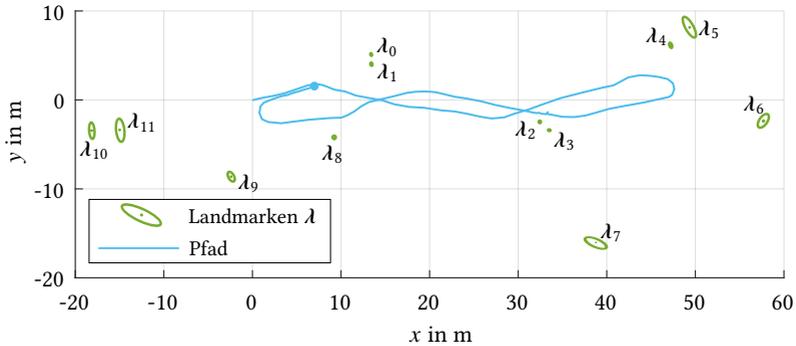


Abbildung 6.3: Kartierungsergebnis der Konfiguration mit 25 Partikeln und dem einfachen Punktlandmarkenmodell. Der Pfad ist in Hellblau dargestellt und die Landmarken λ in Grün.

nis der Konfiguration von 25 Partikeln mit dem einfachen Punktlandmarkenmodell. Die hier erstellte Karte enthält korrekterweise 12 Landmarken inklusive der Fehldetektion und es wurde keine Landmarke fälschlicherweise neu instanziiert. Es traten dennoch einige Fehlzuordnungen auf, wie z. B. wieder gleich zu Beginn bei Beobachtungen der Landmarken λ_0 und λ_1 , was jedoch dank des Partikelfilters im späteren Verlauf korrigiert werden konnte. Auch die Schleifen konnten erfolgreich geschlossen werden. Somit war es möglich, mit mehreren Partikeln auch ohne das erweiterte Landmarkenmodell eine konsistente Karte zu erstellen.

Für die Bewertung der erzielten Genauigkeiten stand keine Referenzlokalisierung oder Karte zur Verfügung. Deshalb konnte nur für einen Teil der Karte eine quantitative Evaluation durchgeführt werden. Bei den Landmarken λ_{11} , λ_8 und λ_2 handelt es sich um Straßenlaternen, welche in einer Reihe entlang einer geraden Straße stehen und es kann davon ausgegangen werden, dass sie entlang einer Gerade ausgerichtet sind. Für die Evaluation wurde jeweils in jeder Karte eine lineare Ausgleichsgerade nach der Methode der kleinsten Quadrate (englisch: *least-squares*) durch diese Landmarken gelegt und die mittlere Distanz der Landmarken von dieser Gerade ermittelt. Eine Überprüfung der Straßenlaternen anhand eines Luftbildes

nach der gleichen Methode ergab einen Mittelwert der Abstände zur Ausgleichsgeraden von 2 mm. Für den Vergleich wurden nur die konsistenten Karten herangezogen und die Ergebnisse sind in Tabelle 6.1 aufgelistet. Anhand von Tabelle 6.1 ist zu erkennen, dass mit dem erweiterten Modell eine höhere Genauigkeit gegenüber dem einfachen Punktlandmarkenmodell erzielt werden konnte.

Tabelle 6.1: Mittelwert der Distanzen zur Geraden in m.

1 Partikel mit erweitertem Modell	0,22
25 Partikel mit Punktlandmarkenmodell	0,51
25 Partikel mit erweitertem Modell	0,19

6.1.2 Zusammenfassung

Die vergleichende Evaluation hat gezeigt, dass ein erweitertes Landmarkenmodell mit ortsunabhängigen Merkmalen zur Robustheit der Datenzuordnung beitragen kann. Durch die verbesserte Datenzuordnung wurden bei dem SLAM-Algorithmus weniger Partikel für das erfolgreiche Schließen der Schleifen benötigt und somit das Erstellen einer konsistenten Karte vereinfacht. Vor allem war bei der Verwendung von nur einem Partikel eine korrekte Kartierung nur mit den zusätzlichen ortsunabhängigen Merkmalen möglich. Weiterhin konnte gezeigt werden, dass bei der Verwendung von mehreren Partikeln die Präzision der Kartierung durch eine gegenüber dem einfachen Punktlandmarkenmodell verbesserte Datenzuordnung gesteigert werden kann.

Ein Grund für das schlechtere Abschneiden des Algorithmus ohne Zusatzmerkmale ist zum Beispiel, dass die Datenzuordnung bereits am Anfang der Fahrt zwei dicht beieinanderliegende Landmarken zeitweise falsch zuordnete, was im Folgenden auch zu einer schlechteren Lokalisierung des Roboters führte. Durch eine zu ungenaue Lokalisierung und weitere Fehlzuordnungen wurde das erfolgreiche Schließen der Schleifen erschwert bzw. im Falle von nur einem Partikel sogar ganz verhindert.

6.2 SLAM – 2D-Rasterkarte

Um den modularen SLAM-Algorithmus mit der 2D-Belegtheitskarte nach Abschnitt 4.2 zu evaluieren, wird dieser zunächst mit einem etablierten Algorithmus anhand von öffentlich verfügbaren Datensätzen verglichen. Die verwendeten Datensätze sind frei erhältlich und können von der Internetseite <http://www2.informatik.uni-freiburg.de/~stachnis/datasets.html> heruntergeladen werden [CS]. Sie beinhalten alle Messungen von der Rad-Odometrie des jeweils verwendeten mobilen Roboters und eines 2D-LiDARs mit 180° Öffnungswinkel. Die Modularität des vorgestellten SLAM-Algorithmus kommt bei diesen Datensätzen noch nicht zum Tragen, da in den Datensätzen keine Messungen von weiteren Sensoren enthalten sind. So soll hier zunächst nur die grundlegende Leistungsfähigkeit evaluiert werden.

Abbildung 6.4 zeigt die 2D-Belegtheitskarte des Datensatzes, welcher von Dirk Hähnel im Intel Research Lab aufgenommen wurde, nur unter Verwendung der Rad-Odometrie. Die Belegtheitswahrscheinlichkeit geht von frei (0 bzw. schwarz) über unbekannt (0,5 bzw. grau) bis belegt (1 bzw. weiß). Zu Beginn werden alle Zellen der Karte mit einer Belegtheitswahrscheinlichkeit von 0,5 initialisiert, da am Anfang noch keinerlei Information über die Umgebung existiert. Der gefahrene Pfad des mobilen Roboters ist in Hellblau dargestellt. Anhand dieser Karte wird deutlich, dass ohne eine simultane Kartierung keine konsistente Karte erzeugt werden kann. Da die Rad-Odometrie nur relative Bewegungen misst und dazu meist auch sehr ungenau ist, können sich Fehler über die Zeit akkumulieren. So entstehen schon über relativ kurze Abstände gekrümmte Gänge und im Gesamten eine fehlerhafte und inkonsistente Karte.

Im Folgenden werden die Ergebnisse des in Abschnitt 4.2 beschriebenen modularen SLAM-Algorithmus vorgestellt und mit Referenzkarten verglichen. Die hier gezeigten Referenzkarten wurden anhand der mit dem GMapping-Algorithmus von Giorgio Grisetti und Cyrill Stachniss erzeugten Lokalisierungsergebnisse erstellt [Gri07]. Diese können ebenfalls von

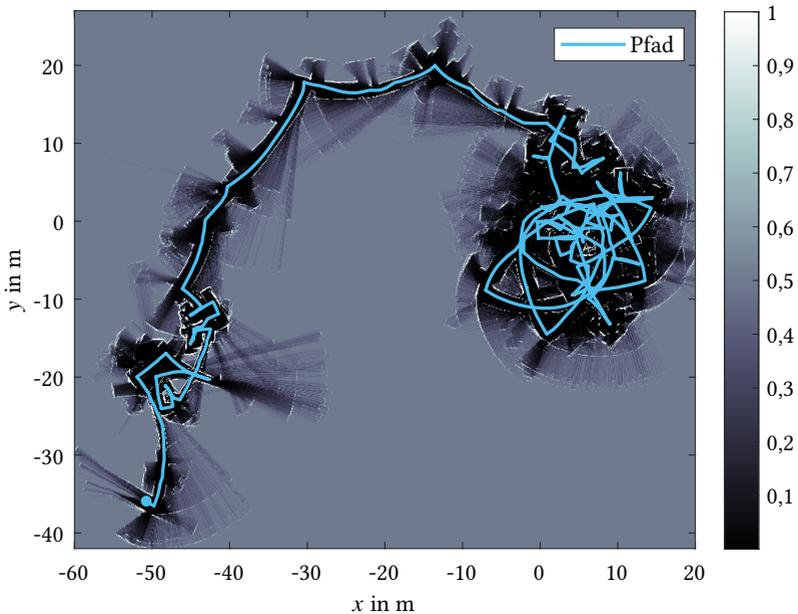


Abbildung 6.4: 2D-Belegtheitskarte des Intel Research Lab und Pfad des mobilen Roboters in Hellblau, aufgezeichnet nur unter Verwendung der Rad-Odometrie. Der Grauwert bildet die Belegheitswahrscheinlichkeit von 0 bis 1 ab.

der o. g. Internetseite heruntergeladen werden. Auf einen quantitativen Vergleich wurde aufgrund von nicht vorhandener Grundwahrheit verzichtet.

6.2.1 Büroumgebung

Der Datensatz vom Intel Research Lab, welcher schon für Abbildung 6.4 herangezogen wurde, beinhaltet eine typische Büroumgebung mit Gängen und vielen kleinen Räumen, jedoch mit einer Besonderheit in Form einer großen Schleife, weshalb der Datensatz für die Evaluation von SLAM-Algorithmus besonders geeignet ist. Die Büroumgebung hat eine Grund-

fläche von in etwa $28 \times 28 \text{ m}^2$ und die Aufnahme weist ein Pfadlänge von etwas über 500 m auf.

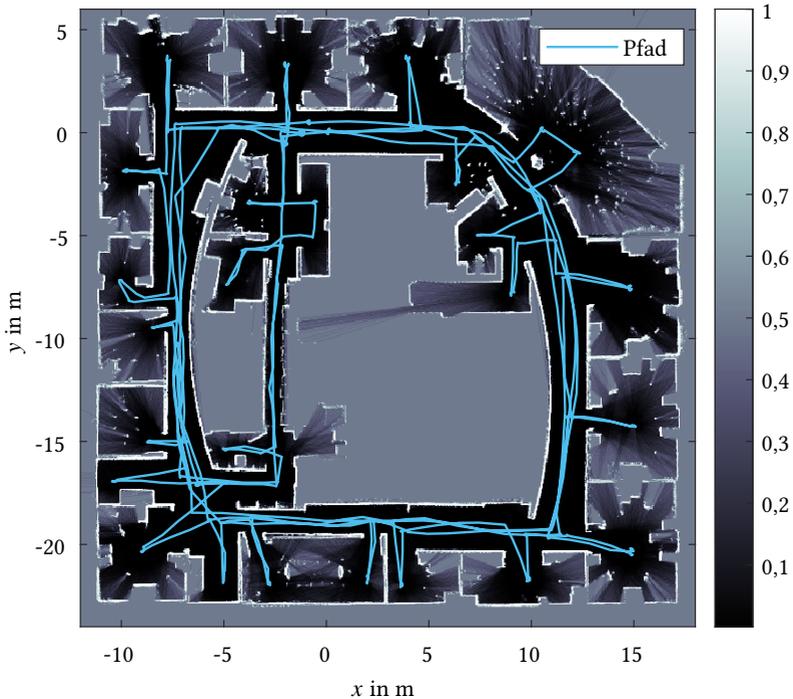


Abbildung 6.5: Referenzkarte des Intel Research Lab und Pfad des mobilen Roboters in Hellblau, berechnet mit dem GMapping-Algorithmus mit 15 Partikeln [CS]. Der Grauwert bildet die Belegheitswahrscheinlichkeit von 0 bis 1 ab.

Abbildung 6.5 zeigt die Referenzkarte. In [Gri07] wurde diese Karte mit 15 Partikeln erzeugt.

Abbildung 6.6 zeigt die mit dem modularen SLAM-Algorithmus mit acht Partikeln erzeugte Karte und den geschätzten Pfad zum Vergleich. Es konnte zwar bereits mit einem Partikel eine konsistente Karte erzeugt werden, bei der jedoch noch leichte Ungenauigkeiten vorhanden sind.

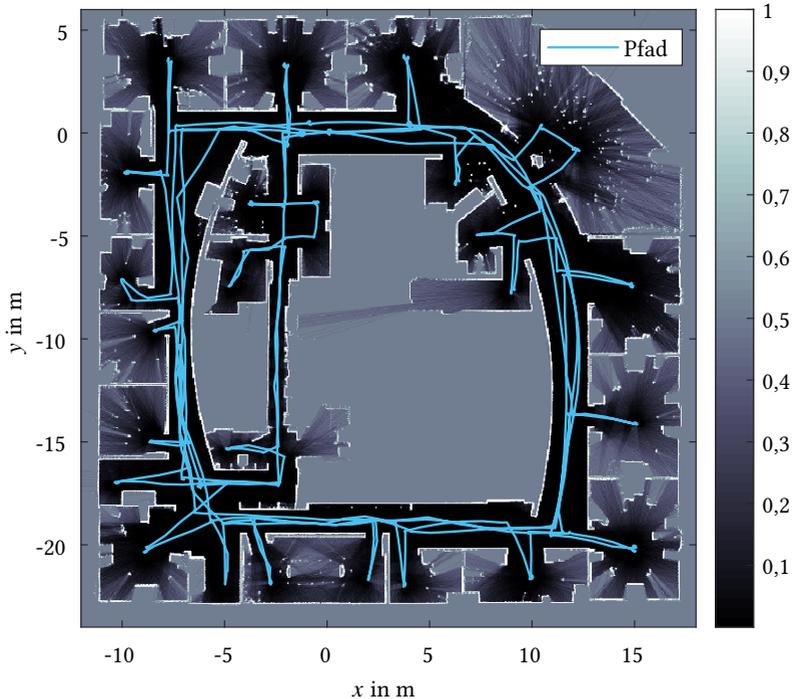


Abbildung 6.6: 2D-Belegtheitskarte des Intel Research Lab und Pfad des mobilen Roboters in Hellblau, berechnet mit dem modularen SLAM-Algorithmus mit acht Partikeln. Der Grauwert bildet die Belegheitswahrscheinlichkeit von 0 bis 1 ab.

6.2.2 Lange Gänge

Der Datensatz des MIT Killian Court wurde von Mike Bosse und John Leonard aufgenommen. Dieser Datensatz enthält eine Umgebung mit vielen langen geraden Gängen auf einer Grundfläche von etwa $250 \times 215 \text{ m}^2$ und die Aufnahme weist ein Pfadlänge von etwas über 1,9 km auf. Der Datensatz ist besonders anspruchsvoll, weil er viele verschachtelte Schleifen beinhaltet und die langen geraden Gänge die longitudinale Lokalisierung mittels Scan-Matching erschweren. Zum einen weist die Lokalisierung

in Längsrichtung kein klares Maximum auf und zum anderen kann es in einem Gang, der neu kartiert wird, dazu kommen, dass die Lokalisierung zurück in den bereits kartierten Teil gezogen wird [Sta08].

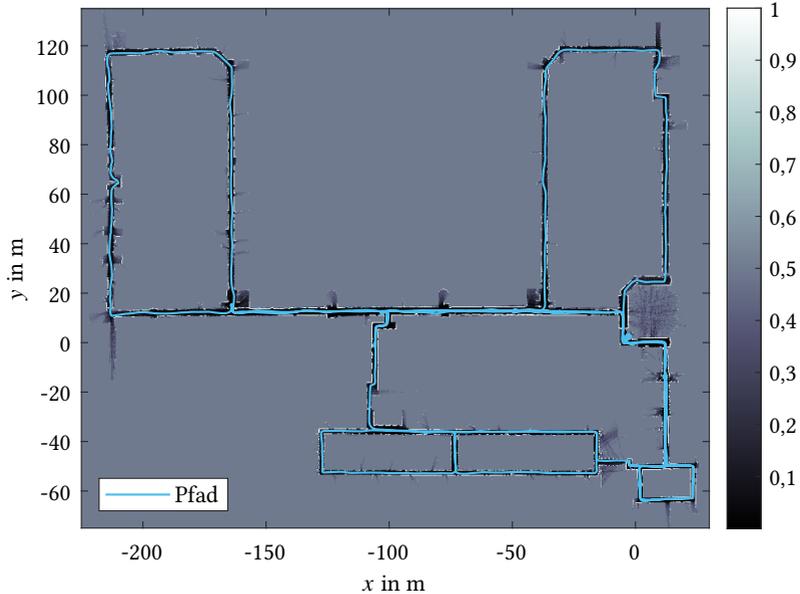


Abbildung 6.7: Referenzkarte des MIT Killian Court und Pfad des mobilen Roboters in Hellblau, berechnet mit dem GMapping-Algorithmus mit 200 Partikeln [CS]. Der Grauwert bildet die Belegtheitswahrscheinlichkeit von 0 bis 1 ab.

Abbildung 6.7 zeigt die Referenzkarte, berechnet mit dem GMapping-Algorithmus mit 200 Partikeln. In [Gri07] konnten mit diesem Datensatz konsistente Karten mit 60 Partikeln erzeugt werden, welche aber noch Ungenauigkeiten wie doppelte Wände aufwiesen. Ab einer höheren Anzahl von 80 Partikeln traten diese Ungenauigkeiten nicht mehr auf.

Abbildung 6.8 zeigt die mit dem modularen SLAM-Algorithmus mit 100 Partikeln erzeugte Karte und den geschätzten Pfad. Dabei konnten alle Schleifen erfolgreich geschlossen und eine konsistente und präzise Karte erzeugt werden.

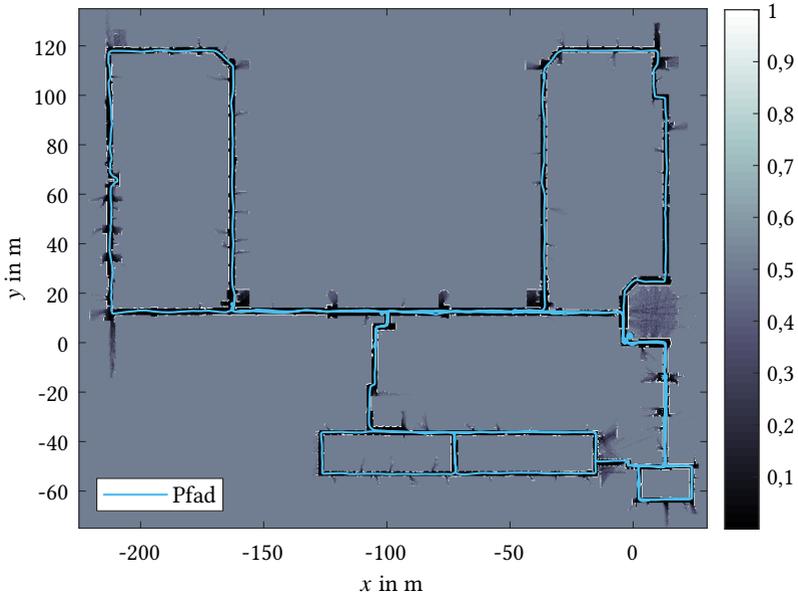


Abbildung 6.8: 2D-Belegtheitskarte des MIT Killian Court und Pfad des mobilen Roboters in Hellblau, berechnet mit dem modularen SLAM-Algorithmus mit 100 Partikeln. Der Grauwert bildet die Belegheitswahrscheinlichkeit von 0 bis 1 ab.

6.2.3 Außengelände

Der Datensatz des Freiburg Campus wurde von Cyrill Stachniss und Giorgio Grisetti aufgenommen. Dieser Datensatz enthält eine Aufnahme eines Außenbereichs mit unterschiedlichen Bereichen. Das Areal erstreckt sich über ca. $250 \times 250 \text{ m}^2$ und die gesamte Pfadlänge beträgt etwa 1,7 km. In dem Areal sind große Gebäude und andere Strukturen vorhanden, aber auch unstrukturierte Objekte, wie Hecken oder Bäume. Außerdem gibt es Bereiche, in denen kaum Objekte vorhanden sind und damit fast keine Beobachtungen möglich sind, da der 2D-LiDAR nur einen Sichtbereich von 180° nach vorne hat. Wie im vorherigen Datensatz sind mehrere verschachtelte Schleifen vorhanden.

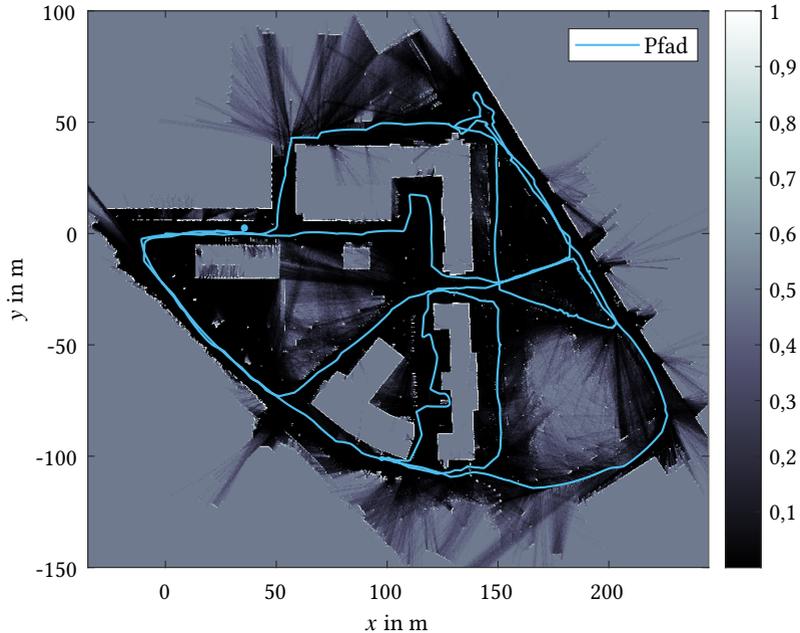


Abbildung 6.9: Referenzkarte des Freiburg Campus und Pfad des mobilen Roboters in Hellblau, berechnet mit dem GMapping-Algorithmus mit 100 Partikeln [CS]. Der Grauwert bildet die Belegtheitswahrscheinlichkeit von 0 bis 1 ab.

Abbildung 6.9 zeigt die Referenzkarte, berechnet mit dem GMapping-Algorithmus mit 100 Partikeln. In [Gri07] konnten mit diesem Datensatz Karten mit guter Qualität bereits ab 30 Partikel erzeugt werden.

Abbildung 6.10 zeigt die mit dem modularen SLAM-Algorithmus erzeugte Karte und den geschätzten Pfad. Die dargestellte Karte wurde mit 100 Partikeln erzeugt. Es konnten alle Schleifen erfolgreich geschlossen werden und eine konsistente und präzise Karte erzeugt werden.

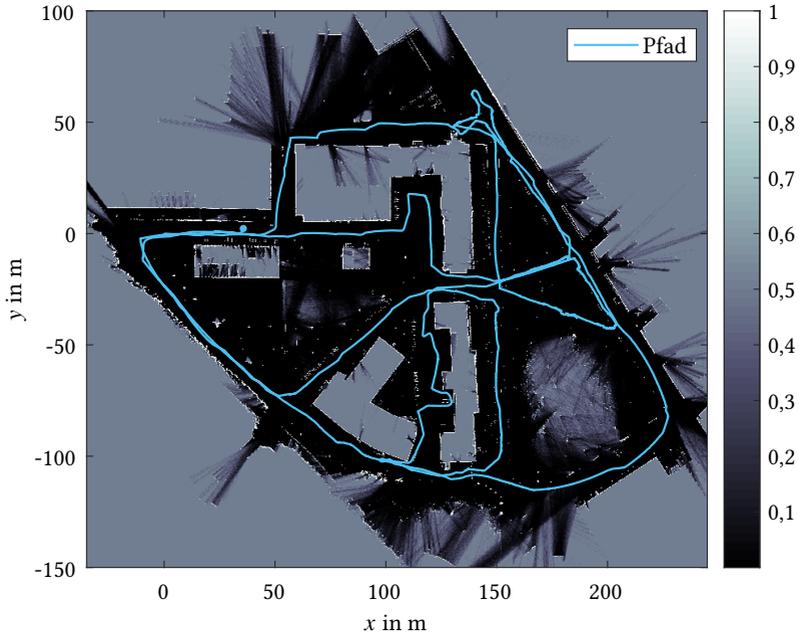


Abbildung 6.10: 2D-Belegtheitskarte des Freiburg Campus und Pfad des mobilen Roboters in Hellblau, berechnet mit dem modularen SLAM-Algorithmus mit 100 Partikeln. Der Grauwert bildet die Belegheitswahrscheinlichkeit von 0 bis 1 ab.

6.2.4 Zusammenfassung

Mit dem modularen SLAM-Algorithmus konnte die erfolgreiche Kartierung mittels Rad-Odometrie und 2D-LiDAR anhand von öffentlich verfügbaren Datensätzen gezeigt werden. Dabei konnten auch sehr anspruchsvolle Datensätze mit großen und mehreren verschachtelten Schleifen in Umgebungen mit langen gerade Gängen oder auch im Außengelände mit unstrukturierten Bereichen kartiert werden.

Die zum Vergleich herangezogenen Referenzkarten stammen von einem etablierten SLAM-Verfahren und in der qualitativen Gegenüber-

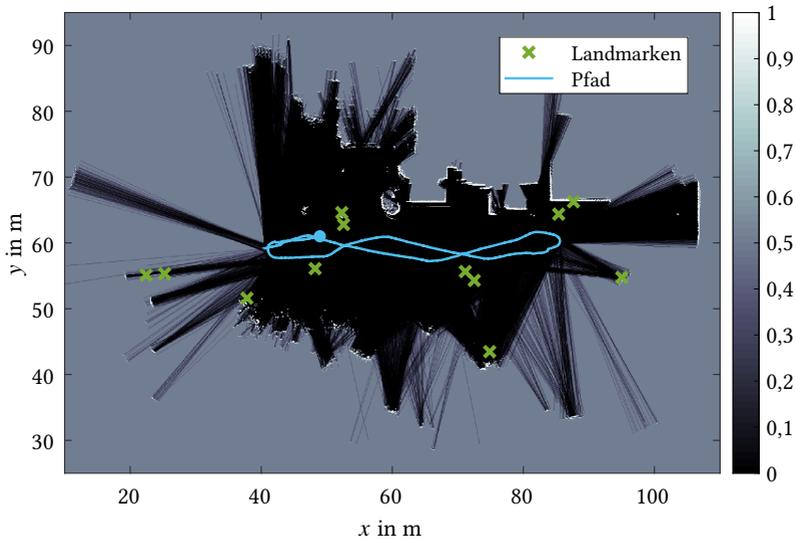


Abbildung 6.11: Kartierungsergebnis, berechnet mit dem modularen SLAM-Algorithmus nur mit erweitertem Landmarkenmodell. Der Pfad ist in Hellblau dargestellt und die Landmarken in Grün. Der Grauwert bildet die Belegheitswahrscheinlichkeit von 0 bis 1 ab.

stellung konnten mit dem modularen SLAM-Algorithmus mit ähnlichen Anzahlen von Partikeln gleichwertige Karten erstellt werden.

Die Modularität des vorgestellten SLAM-Algorithmus kam, wie bereits angemerkt, bei diesen Datensätzen noch nicht zum Tragen, sondern wird in den folgenden Evaluationen mit den erweiterten Methoden relevant.

6.3 Kombination von Landmarken und Rasterkarte

Wie in Abschnitt 4.3 dargestellt, werden hier die beiden Ansätze für die Kartierung mit erweiterten Landmarkenmodellen und für die 2D-Rasterkartierung in einem hybriden SLAM-Algorithmus kombiniert.

Bei dem zur Evaluation herangezogenen Datensatz handelt es sich um den gleichen wie in Abschnitt 6.1. Das Areal, in dem die Daten aufgenommen wurden, beinhaltet neben Laternenmasten und Baumstämmen als Landmarkenobjekte auch geradlinige Strukturen wie Gebäudeteile sowie unstrukturierte Bereiche mit Bäumen und Büschen.

6.3.1 Evaluation

Wie in Abschnitt 4.3 dargestellt, kann der modulare SLAM-Algorithmus flexibel mit unterschiedlichen Kartenarten und auch deren Kombination genutzt werden. Für die Evaluation wurden diese Konfigurationen miteinander verglichen:

1. nur mit Landmarken,
2. nur mit Rasterkarte,
3. Kombination von Landmarken und 2D-Rasterkarte.

Abbildung 6.11 zeigt das Kartierungsergebnis, welches mit dem modularen SLAM-Algorithmus nur mit dem erweiterten Landmarkenmodell für die Kartierung und Lokalisierung berechnet wurde. Der geschätzte Pfad ist in Hellblau dargestellt und die Landmarken als grüne Kreuze. Zusätzlich ist die 2D-Belegtheitskarte angezeigt, wobei diese hier nur dem visuellen Vergleich mit den anderen Konfigurationen dient und in dieser Konfiguration nicht zur Lokalisierung verwendet wurde. Dabei ist die Belegtheitswahrscheinlichkeit wie in Abschnitt 6.2 von frei (0 bzw. schwarz) über unbekannt (0,5 bzw. grau) bis belegt (1 bzw. weiß) dargestellt.

Abbildung 6.12 zeigt das Kartierungsergebnis, berechnet mit dem modularen SLAM-Algorithmus nur mit der 2D-Rasterkarte für die Kartierung und Lokalisierung. Diese Konfiguration entspricht der Evaluation im vorherigen Abschnitt. Hier sind nur die drei Landmarken eingezeichnet, welche schon in Abschnitt 6.1 zur Evaluation verwendet wurden. Die Landmarken fanden in dieser Konfiguration zwar keine Verwendung

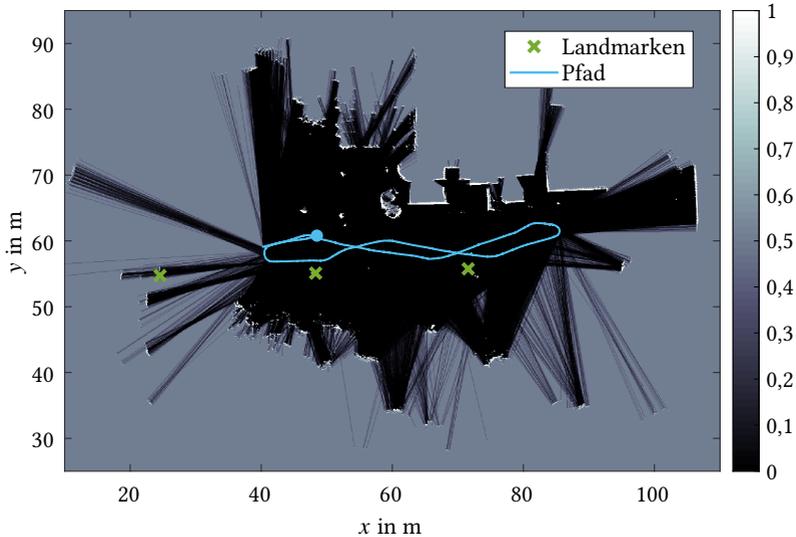


Abbildung 6.12: Kartierungsergebnis, berechnet mit dem modularen SLAM-Algorithmus nur mit 2D-Rasterkarte. Der Pfad ist in Hellblau dargestellt und die Landmarken in Grün. Der Grauwert bildet die Belegheitswahrscheinlichkeit von 0 bis 1 ab.

für die Lokalisierung oder Kartierung, sollen jedoch später ebenfalls zur Evaluation herangezogen werden. Deshalb wurden sie hier anhand der Lokalisierung rekonstruiert.

Abbildung 6.13 zeigt das Kartierungsergebnis, berechnet mit dem modularen SLAM-Algorithmus und der hybriden Konfiguration, d. h. mit der Kombination von erweitertem Landmarkenmodell und 2D-Rasterkarte, wobei in dieser Konfiguration auch beides für die Kartierung und Lokalisierung genutzt wird.

Alle drei Karten sehen sich sehr ähnlich und mit allen Konfigurationen konnten konsistente Karten erstellt werden. In Ermangelung einer Referenzkarte oder -lokalisierung werden im Folgenden qualitative Vergleiche der erzielten Genauigkeiten anhand von Detailbetrachtungen herangezogen.

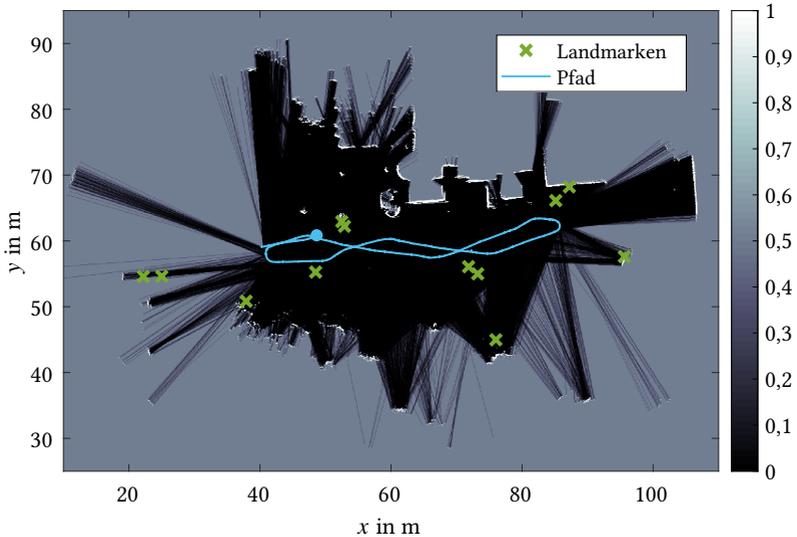


Abbildung 6.13: Kartierungsergebnis, berechnet mit dem modularen SLAM-Algorithmus mit hybrider Konfiguration. Der Pfad ist in Hellblau dargestellt und die Landmarken in Grün. Der Grauwert bildet die Belegheitswahrscheinlichkeit von 0 bis 1 ab.

In Abbildung 6.14 sind drei Detailansichten eines Eingangs zu einem Fahrradunterstand dargestellt, wobei Abbildung 6.14a das Ergebnis der Kartierung nur anhand der erweiterten Landmarken zeigt, Abbildung 6.14b das Ergebnis der Kartierung nur anhand der 2D-Rasterkarte und Abbildung 6.14c das Ergebnis der hybriden Konfiguration. Während bei dem modularen SLAM-Algorithmus nur mit dem erweitertem Landmarkenmodell aufgrund von Ungenauigkeiten die Wände mehrfach als Doppelstrukturen vorhanden sind bzw. nur mit der 2D-Rasterkarte noch leichte Ungenauigkeiten existieren, konnte mit dem SLAM-Algorithmus mit der hybriden Konfiguration eine sehr viel genauere Kartierung erzielt werden und die Wände des Fahrradunterstands sind klarer zu erkennen.

In Abbildung 6.15 sind drei Detailansichten eines Busches dargestellt, wobei Abbildung 6.15a das Ergebnis der Kartierung nur anhand der erwei-

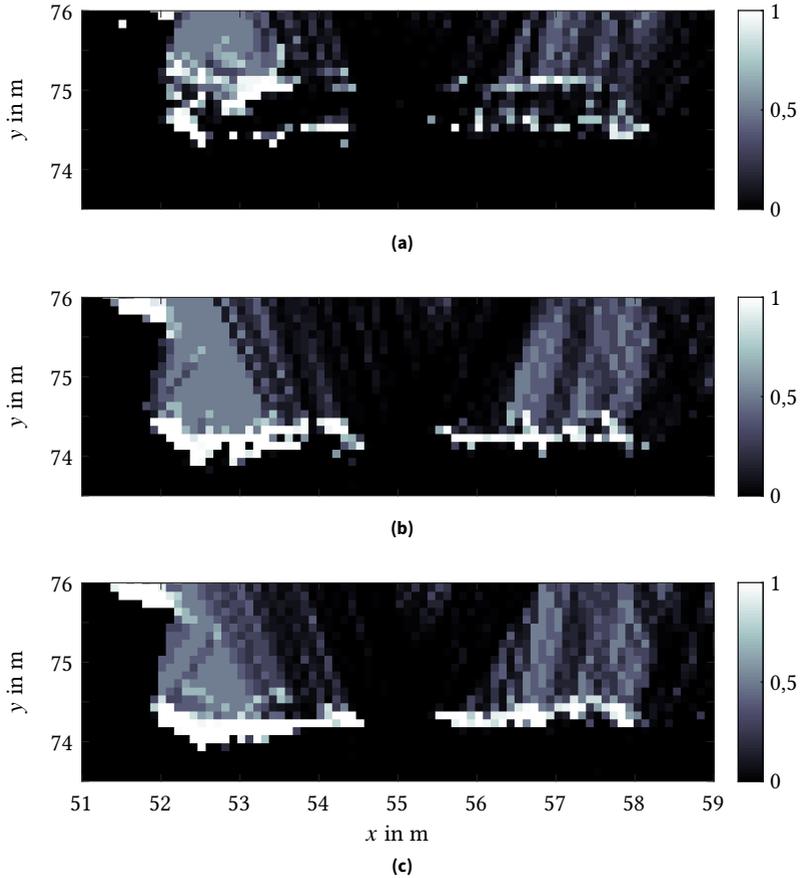


Abbildung 6.14: Detailansicht eines Eingangs: Modularer SLAM-Algorithmus mit erweitertem Landmarkenmodell (a), modularer SLAM-Algorithmus mit 2D-Rasterkarte (b) und modularer SLAM-Algorithmus mit hybrider Konfiguration (c). Der Grauwert bildet die Belegheitswahrscheinlichkeit von 0 bis 1 ab.

terten Landmarken zeigt, 6.15b das Ergebnis der Kartierung nur anhand der 2D-Rasterkarte und 6.15c das Ergebnis der hybriden Konfiguration. Hier ist kein so deutlicher Unterschied zu erkennen wie in Abbildung 6.14. Dennoch sind bei dem modularen SLAM-Algorithmus nur mit dem erweiterten Landmarkenmodell bzw. nur mit der 2D-Rasterkarte leichte Ungenauigkeiten zu erkennen, während der Busch mit dem modularen SLAM-Algorithmus mit der hybriden Konfiguration eine klarere Kontur aufweist.

Für eine quantitative Evaluation wurden auch hier wieder die Landmarken der Straßenlaternen λ_{11} , λ_8 und λ_2 herangezogen (siehe Abschnitt 6.1). Es wurde wieder jeweils in jeder Karte eine lineare Ausgleichsgerade durch diese Landmarken gelegt und die mittlere Distanz der Landmarken von dieser Gerade ermittelt. Die Ergebnisse sind in Tabelle 6.2 aufgelistet. Anhand der Tabelle ist zu erkennen, dass die Genauigkeit mit der hybriden Kartierung gegenüber den beiden Einzelansätzen gesteigert werden konnte, wobei sich gegenüber der Konfiguration nur mit der Rasterkarte lediglich ein leichter Unterschied zeigt.

Tabelle 6.2: Mittelwert der Distanzen zur Geraden in m.

Nur mit Landmarken	0,22
Nur mit Rasterkarte	0,06
Kombination von Landmarken und Rasterkarte	0,04

6.3.2 Zusammenfassung

Mit der Evaluation konnte gezeigt werden, dass mit dem modularen SLAM-Algorithmus mit der hybriden Konfiguration gegenüber den Einzelverfahren eine verbesserte Kartierung und damit auch Lokalisierung erzielt werden konnte. Dies ist besonders deutlich im Vergleich zu dem SLAM-Algorithmus mit den Landmarken. Wie schon in Abschnitt 6.1 erwähnt, wurden nur in ca. 20 % der Daten Landmarken extrahiert, was zum Teil in

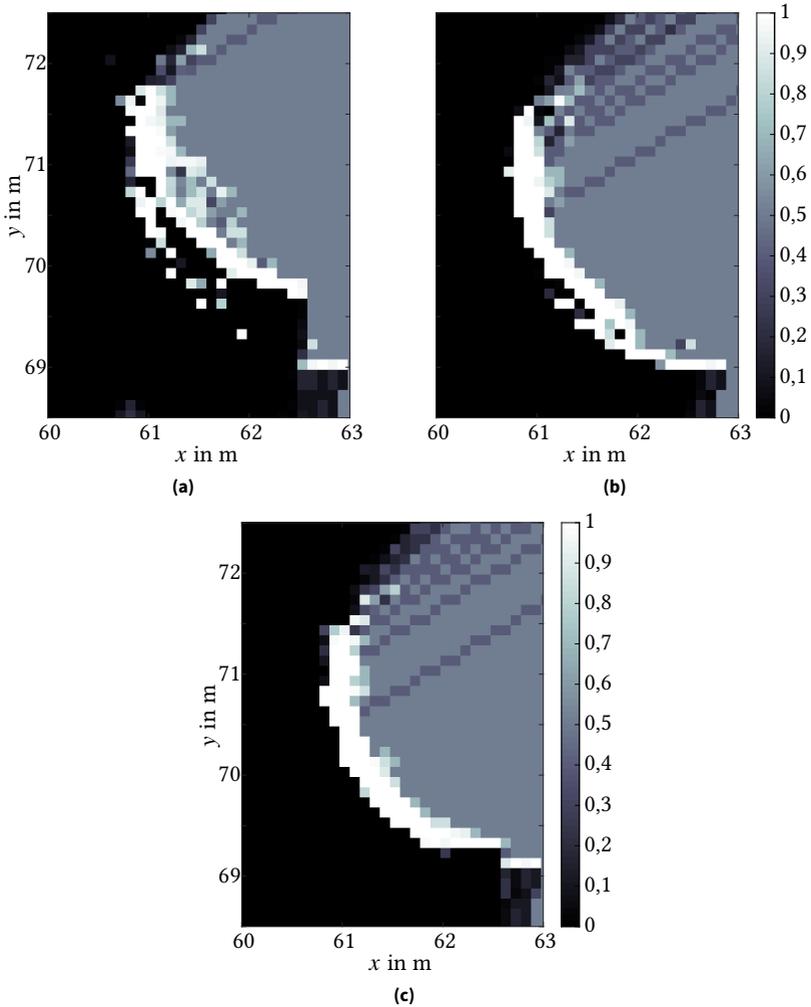


Abbildung 6.15: Detailansicht eines Busches: Modularer SLAM-Algorithmus mit erweitertem Landmarkenmodell (a), modularer SLAM-Algorithmus mit 2D-Rasterkarte (b), modularer SLAM-Algorithmus mit hybrider Konfiguration (c). Der Grauwert bildet die Belegheitswahrscheinlichkeit von 0 bis 1 ab.

dem spärlichen Vorhandensein der Landmarkenobjekte und zum Teil in dem relativ kleinen Öffnungswinkel der verwendeten Kamera-Objektiv-Kombination begründet liegt. Demgegenüber hat der 2D-LiDAR mit 180° einen wesentlich größeren Öffnungswinkel und es stehen in allen Bereichen des Areals valide Messungen zur Kartierung und Lokalisierung zur Verfügung.

Umgekehrt kann jedoch auch der SLAM-Algorithmus mit der hybriden Konfiguration gegenüber dem SLAM-Algorithmus nur mit der 2D-Rasterkarte ebenfalls eine leichte Verbesserung erzielen. Dies liegt darin begründet, dass die Lokalisierung und Kartierung in einer unstrukturierten Umgebung mit einer Rasterkarte höhere Ungenauigkeiten aufweisen, welche durch die Hinzunahme von Landmarkenobjekten leicht verringert werden können.

6.4 Kombination von Multi-Sensor-Fusion und SLAM mit 2D-Rasterkarte

Die in Abschnitt 5.1 vorgestellte Kombination von Multi-Sensor-Fusion und SLAM mit 2D-Rasterkarte soll anhand der Integration eines zusätzlichen absoluten Sensors in Form eines GNSS evaluiert werden. Mit dem schon in den vorherigen Ergebnisabschnitten verwendeten modularen SLAM-Algorithmus können flexibel weitere Sensoren, wie z. B. ein GNSS oder ein Kompass, in die Schätzung integriert werden.

Der für die Evaluierung herangezogene Datensatz wurde mit einer geländegängigen Plattform mit vier angetriebenen Rädern und einer Allradlenkung aufgenommen. Diese Plattform ist mit einem MTi-G-700 von Xsens ausgestattet, welches sowohl eine IMU als auch einen GNSS-Empfänger beinhaltet, wobei für diese Evaluation nur die GNSS-Daten verwendet wurden. Des Weiteren verfügt die Plattform über einen Rad-Odometriesensor und einen 3D-LiDAR Velodyne HDL-64E, wobei von letzterem nur ein horizontaler 2D-Ausschnitt für die Evaluation genutzt

wurde. Da kein absoluter Sensor für die direkte Messung des Gierwinkels vorhanden ist, wird diese in einem Vorverarbeitungsschritt aus vergangenen 2D-Positionsmessungen des GNSS geschätzt und geht als virtueller Kompass-Sensor mit ein. Der Datensatz wurde auf einem Areal aufgenommen, welches neben urbanen Strukturen und Vegetation mit Büschen, Hecken und Bäumen auch Flächen mit geparkten Fahrzeugen beinhaltet. Außerdem befinden sich hohe Gebäude in der Nähe, was mit für GNSS kritischen Abschattungen und Mehrwegeausbreitungssituationen einhergeht.

Die Integration der GNSS-Daten erlaubt auch eine Geo-Referenzierung und die Angabe der globalen Koordinaten erfolgt in dem Universal-Transverse-Mercator-System (UTM-System), welches heute sehr weit verbreitet ist. Die hier dargestellte und alle folgenden Karten befinden sich in der UTM-Zone 32 und Band U. Zur besseren Vergleichbarkeit wurde bei den zum Vergleich herangezogenen Methoden und Konfigurationen nur mit Rad-Odometrie und 2D-LiDAR die Anfangspose auf die globale Startpose gesetzt.

6.4.1 Evaluation

Abbildung 6.16 zeigt das Kartierungsergebnis des modularen SLAM-Algorithmus unter Verwendung der Sensordaten von Rad-Odometrie und 2D-LiDAR mit nur einem Partikel. Im linken Teil der Karte ist ein großes kreuzförmiges Gebäude zu erkennen, während im rechten Teil Vegetation mit Bäumen und Büschen vorherrscht. Das gesamte Gebiet ist durch einen Zaun mit einer dichten Hecke eingeschlossen und im unteren rechten Teil ist ein großer Parkplatz zu finden. Der befahrene Pfad beginnt ca. bei Koordinate (110 m, 150 m) und geht zweimal im Gegenuhrzeigersinn um das große Gebäude, d. h., nach der ersten Runde ist eine große Schleife zu schließen. Nach der zweiten Runde führt der Pfad in mehreren Schleifen durch die Vegetation und anschließend über den Parkplatz, ebenfalls in mehreren Schleifen, um schließlich nahe dem Startpunkt zu enden.

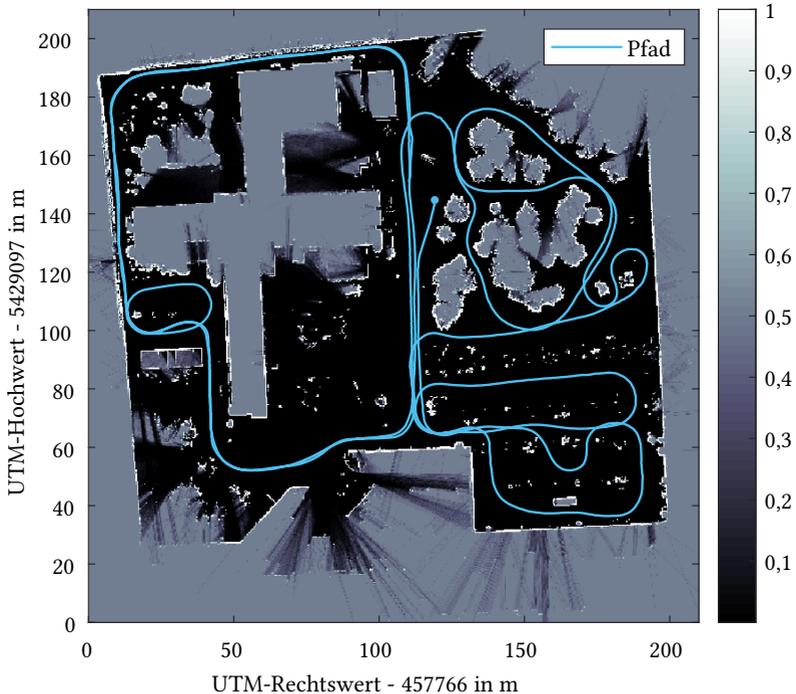


Abbildung 6.16: Kartierungsergebnis des modularen SLAM-Algorithmus mit Rad-Odometrie und 2D-LiDAR mit einem Partikel. Der Pfad ist in Hellblau dargestellt und der Grauwert bildet die Belegheitswahrscheinlichkeit von 0 bis 1 ab.

Auch unter ausschließlicher Verwendung von Rad-Odometrie und 2D-LiDAR konnte eine nahezu konsistente Karte erzeugt werden, jedoch weist sie einige kleine Ungenauigkeiten auf, vor allem beim Schließen der großen Schleife ca. bei Koordinate (100 m, 170 m) sind mehrere Wände doppelt kartiert worden. Dieser Ausschnitt ist in Abbildung 6.17 nochmals in einer Detailansicht gezeigt.

Der in Abschnitt 6.2 zur vergleichenden Evaluation herangezogene GMapping-Algorithmus nach [Gri07] lässt sich hier nicht anwenden, da er keine einfache Integration von zusätzlichen Sensoren wie GNSS

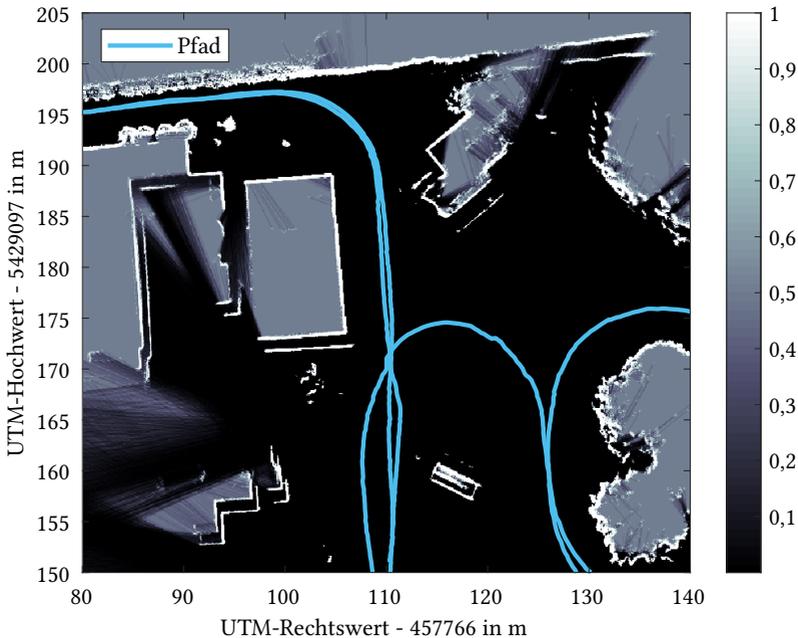


Abbildung 6.17: Detailausschnitt des Kartierungsergebnis des modularen SLAM-Algorithmus mit Rad-Odometrie und 2D-LiDAR mit einem Partikel. Der Pfad ist in Hellblau dargestellt und der Grauwert bildet die Belegtheitswahrscheinlichkeit von 0 bis 1 ab.

erlaubt. In Abschnitt 6.2 wurde außerdem gezeigt, dass mit dem modularen SLAM-Algorithmus gleichwertige Ergebnisse erzielt werden können. Mit Hector-SLAM wurde ein anderer etablierter SLAM-Algorithmus zum Vergleich gewählt, da dieser zwar keine Fusion mit einem zusätzlichen Sensor erlaubt, jedoch eine einfache Möglichkeit bietet, die Information eines globalen Sensors zu integrieren. Die Lokalisierung in Hector-SLAM wird über das Scan-Matching mit der Karte realisiert, wobei die Startpose für das Scan-Matching aus einer zusätzlichen Bewegungsschätzung vorgegeben werden kann. Im Gegensatz zu den Ansätzen mit Partikelfiltern gibt es bei Hector-SLAM lediglich eine einzige Lokalisierung und Karte [Koh11]. Abbildung 6.18 zeigt das Ergebnis von Hector-SLAM, wobei

die Bewegungsschätzung für die Startpose des Scan-Matchings mittels Rad-Odometrie erfolgt. Das Schließen der ersten großen Schleife bei Position (200 m, 200 m) schlägt hier fehl und im weiteren Verlauf kann dies nicht mehr korrigiert werden, sondern führt auf der linken Seite bei Position (125 m, 175 m) letztendlich dazu, dass die Kartierung scheitert.

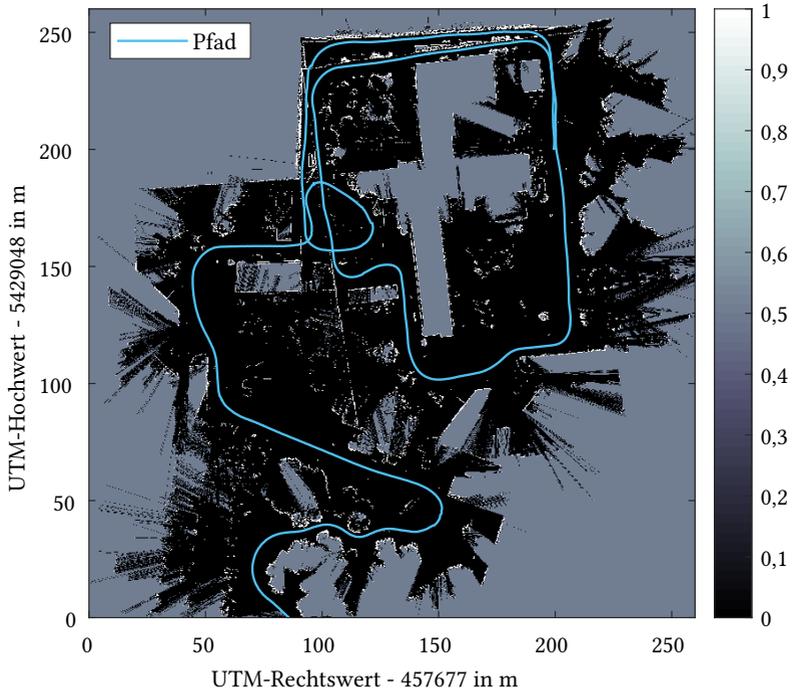


Abbildung 6.18: Kartierungsergebnis von Hector-SLAM mit Rad-Odometrie und 2D-LiDAR. Der Pfad ist in Hellblau dargestellt und der Grauwert bildet die Belegtheitswahrscheinlichkeit von 0 bis 1 ab.

Abbildung 6.19 zeigt das Ergebnis des Hector-SLAM-Algorithmus, wenn für die Startposition des Scan-Matchings jeweils die aktuelle GNSS-Position und Ausrichtung aus der virtuellen Kompass-Messung mit eingeht. D. h., hier geht nun eine absolute Pose in die Startpose für das Scan-

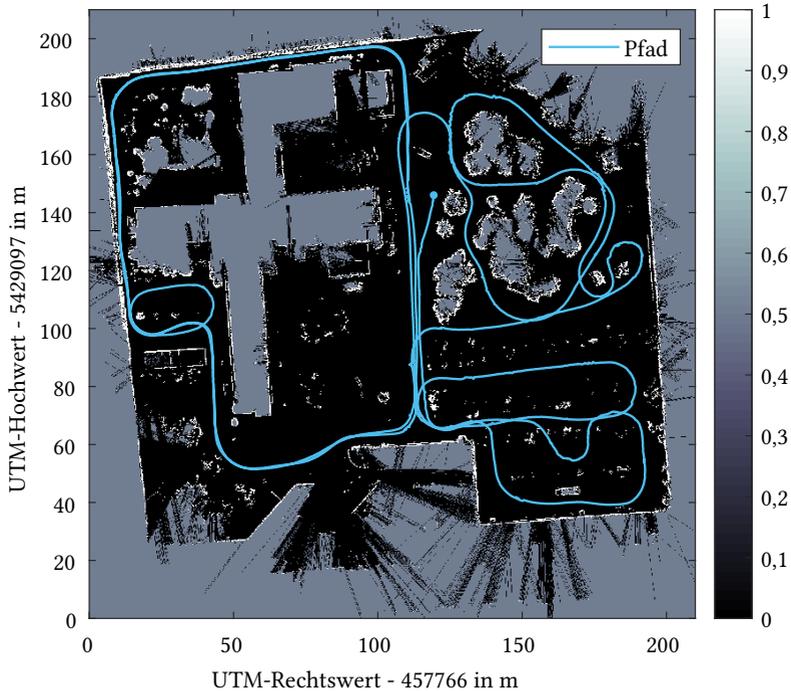


Abbildung 6.19: Kartierungsergebnis von Hector-SLAM mit Rad-Odometrie, GNSS und 2D-LiDAR. Der Pfad ist in Hellblau dargestellt und der Grauwert bildet die Belegheitswahrscheinlichkeit von 0 bis 1 ab.

Matching ein, wodurch lokale Fehler korrigiert werden können und sich nicht mehr wie in Abbildung 6.18 katastrophal auswirken. Dennoch sind Ungenauigkeiten beim Schließen der ersten großen Schleife ca. bei Koordinate (100 m, 170 m) zu erkennen.

Das Ergebnis des modularen SLAM-Algorithmus mit einem Partikel mit zusätzlich integrierten GNSS-Daten und Messungen des virtuellen Kompasses ist in Abbildung 6.20 dargestellt. Gegenüber dem Ergebnis des modularen SLAM-Algorithmus nur mit Rad-Odometrie und 2D-LiDAR ist eine Verbesserung deutlich zu erkennen. Die Karte weist keine Ungenau-

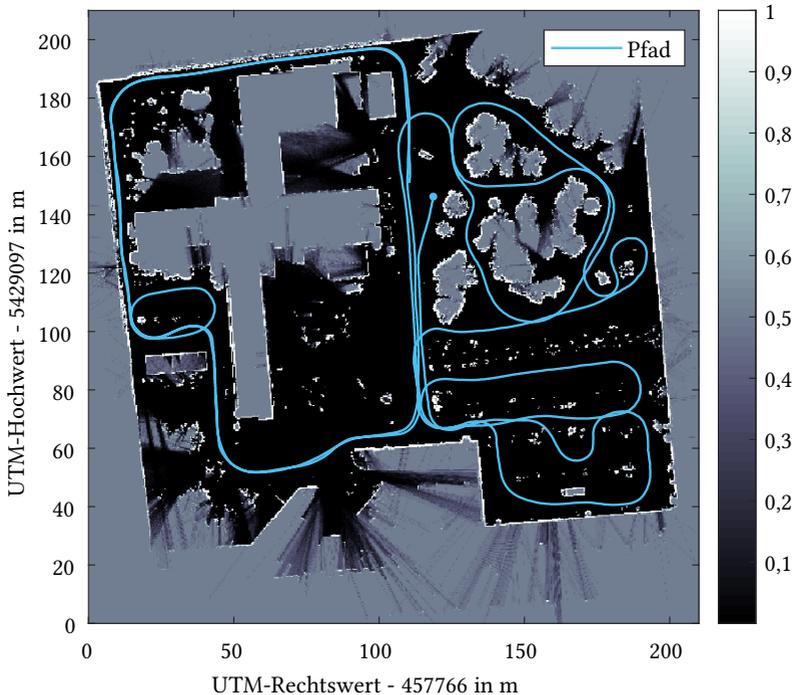


Abbildung 6.20: Kartierungsergebnis des modularen SLAM-Algorithmus mit Rad-Odometrie, GNSS und 2D-LiDAR mit einem Partikel. Der Pfad ist in Hellblau dargestellt und der Grauwert bildet die Belegheitswahrscheinlichkeit von 0 bis 1 ab.

igkeiten mehr im oberen Teil auf. D. h. die erste große Schleife konnte mit verbesserter Genauigkeit geschlossen werden. Dieser Ausschnitt ist in Abbildung 6.21 nochmals in einer Detailansicht gezeigt.

Abbildung 6.22 zeigt das Kartierungsergebnis des modularen SLAM-Algorithmus wieder ohne zusätzliche Integration von GNSS-Daten und Kompass-Messungen, jedoch mit einer Anzahl von 80 Partikeln. Auch hier konnte eine Karte erstellt werden, welche keine Ungenauigkeiten mehr aufweist. D. h., es konnten über die Diversität der Partikel nach dem ersten

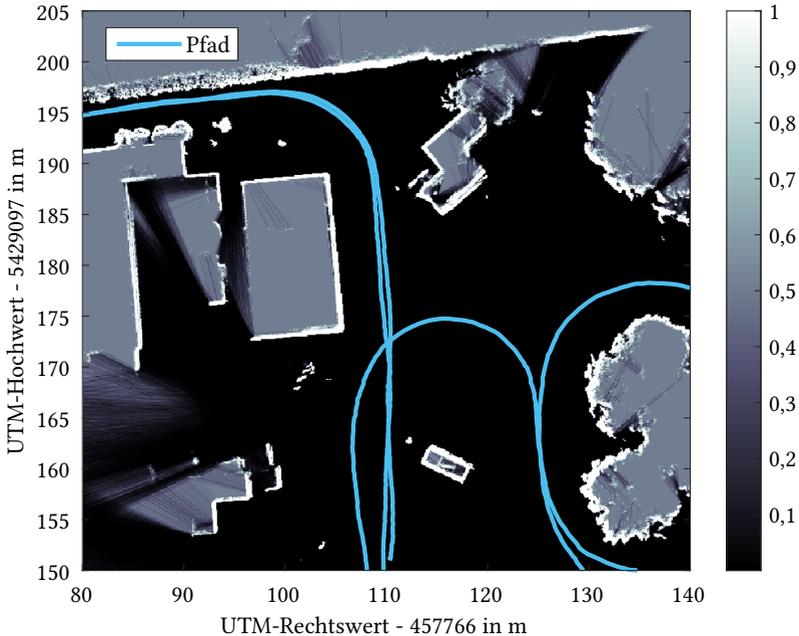


Abbildung 6.21: Detailansicht des Kartierungsergebnis des modularen SLAM-Algorithmus mit Rad-Odometrie, GNSS und 2D-LiDAR mit einem Partikel. Der Pfad ist in Hellblau dargestellt und der Grauwert bildet die Belegheitswahrscheinlichkeit von 0 bis 1 ab.

Schleifenschluss die Partikel mit den Ungenauigkeiten in der jeweiligen Karte beim Resampling eliminiert werden.

6.4.2 Resampling und Laufzeit

Der modulare SLAM-Algorithmus setzt eine adaptive Resampling-Strategie ein, welche das Resampling nur ausführt, wenn die Partikel die Zielverteilung nicht mehr gut approximieren, siehe auch Abschnitt 3.2.1. D. h., je besser die Zielverteilung approximiert wird, desto seltener wird das Resampling durchgeführt.

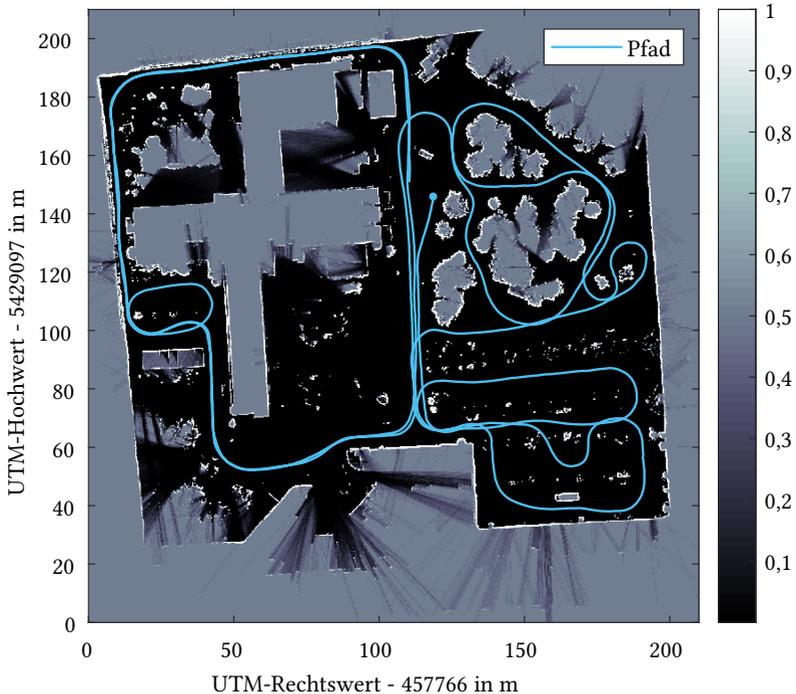


Abbildung 6.22: Kartierungsergebnis des modularen SLAM-Algorithmus mit Rad-Odometrie und 2D-LiDAR mit 80 Partikeln. Der Pfad ist in Hellblau dargestellt und der Grauwert bildet die Belegheitswahrscheinlichkeit von 0 bis 1 ab.

Der Datensatz hat insgesamt 5854 LiDAR-Messungen und zum Vergleich wurden zehn Durchläufe mit 100 Partikeln jeweils mit und ohne GNSS-Messungen durchgeführt. Bei dem modularen SLAM-Algorithmus mit lediglich Rad-Odometrie und 2D-LiDAR wurde bei diesem Datensatz zwischen 14 und 18 Mal (im Schnitt 16,8 Mal) Resampling durchgeführt. Mit zusätzlicher Integration der GNSS- und Kompass-Messungen waren es nur noch acht bis neun Mal (im Schnitt 8,3 Mal). Die höhere Anzahl von Resampling-Operationen, bei dem Partikel mit den zugehörigen Karten kopiert werden müssen, wirkt sich auch auf die Laufzeit aus. Eben-

falls einen Einfluss auf die benötigte Rechenzeit hat die Lokalisierung in der Karte mittels Scan-Matching, welche bei einem besseren Startwert schneller konvergiert. So benötigte der modulare SLAM-Algorithmus mit lediglich Rad-Odometrie und 2D-LiDAR für den kompletten Datensatz auf einem Intel Xeon E5-2687W v2 mit 3,4 GHz und acht Kernen im Schnitt 7528 s und mit zusätzlicher Integration der GNSS-Messungen nur 6701 s.

Für eine Analyse der Laufzeiten bei unterschiedlichen Anzahlen an Partikeln wurden weitere Durchläufe durchgeführt. Auch dieses Mal wurde wieder über jeweils zehn Durchläufe gemittelt, wobei nur ein Ausschnitt aus dem Datensatz ohne Schleifen verwendet wurde. Es kam ebenfalls ein Intel Xeon E5-2687W v2 mit 3,4 GHz und acht Kernen zum Einsatz. Die Ergebnisse sind in Tabelle 6.3 mit den jeweiligen Rechenzeiten pro Aktualisierungsschritt in ms aufgeführt.

Tabelle 6.3: Mittlere Rechenzeit pro Aktualisierungsschritt in ms

Partikelzahl	1	2	4	8	16	32	64	128
Rechenzeit in ms	29	54	58	69	127	239	459	907

Da die Partikel parallel berechnet werden, wäre im Idealfall bei einem Prozessor mit acht Kernen mit bis zu acht Partikeln eine in etwa gleichbleibende Laufzeit zu erwarten und darüber ein proportionaler Anstieg. Bei einem realen Prozessor spielen jedoch noch weitere Faktoren, wie z. B. Cache-Größe oder dynamische Taktanpassung, eine Rolle. Außerdem gibt es Teile in dem Partikelfilteralgorithmus, welche nicht parallelisierbar sind, wie z. B. die Berechnung des effektiven Partikelgewichts und das Resampling. Von nur einem auf zwei Partikel ist ein Anstieg der Laufzeit zu beobachten, da bei nur einem Partikel die genannten und weitere Berechnungen, wie beispielsweise die Berechnung der Gewichte der Partikel, nicht anfallen. Zwischen zwei und vier Partikeln ist die Laufzeit zwar nahezu gleichbleibend, bei acht Partikeln steigt die Laufzeit jedoch etwas an, was darauf zurückzuführen ist, dass der Prozessor über eine dynamische Anpassung der Taktfrequenz verfügt und bei Auslastung aller Kerne nicht mehr

mit dem maximalen Takt läuft. Ab acht Partikeln und mehr ist bei Verdoppelung der Anzahl der Partikel beim Anstieg der Laufzeit in etwa eine Verdoppelung zu beobachten. Angesichts dessen, dass ein gewisser Teil nicht parallelisierbar ist, entspricht die Skalierung in etwa der theoretisch zu erwartenden Laufzeit proportional zur Anzahl der Partikel.

6.4.3 Zusammenfassung

Die Evaluation hat gezeigt, dass durch die Hinzunahme eines absoluten Sensors die Robustheit und Präzision von SLAM-Algorithmen hinsichtlich des Schließens großer Schleifen effektiv gesteigert werden kann. Durch die Fusion mit zusätzlichen GNSS-Messungen konnte bei dem modularen SLAM-Algorithmus selbst mit einem Partikel eine sehr genaue Karte erzeugt werden, wofür ohne die Integration der GNSS-Messungen 80 Partikel notwendig waren. Aber auch bei der gleichen Anzahl von Partikeln konnten Vorteile für die Integration des GNSS-Empfängers aufgezeigt werden. Das adaptive Resampling musste seltener durchgeführt werden, was bedeutet, dass die Zielverteilung besser approximiert wird. Dies führt einerseits zu einer geringeren Rechenzeit und wirkt sich andererseits positiv auf die Diversität der Partikel aus, was letztendlich zu einer robusteren Kartierung führt.

Des Weiteren wurde gezeigt, dass auch bei anderen Methoden wie Hector-SLAM die Integration der Information eines absoluten Sensors hinsichtlich der Robustheit unterstützend wirken kann.

6.5 SC-EKF

Für die Evaluation der Filter mit stochastischem Klonen werden die neuartigen Methoden des impliziten Klonens mit Smoothing nach Abschnitt 5.2.4 und des partiellen impliziten Klonens mit Smoothing nach Abschnitt 5.2.5 mit den bestehenden Methoden des stochastischen Klonens verglichen und Faktor-Graph-Methoden gegenübergestellt.

Der für die Evaluierung herangezogene Datensatz wurde wieder mit der in Abschnitt 6.2 erwähnten geländegängigen Plattform aufgenommen. Der Datensatz wurde außerdem wieder auf demselben Areal aufgenommen, welches neben urbanen Strukturen und Vegetation mit Büschen, Hecken und Bäumen auch Flächen mit geparkten Fahrzeugen umfasst. Verwendet wurden die Sensordaten eines MTi-G-700 von Xsens, welches sowohl eine IMU als auch einen GNSS-Empfänger beinhaltet, sowie die Messungen eines Rad-Odometriesensors und eines 3D-LiDARs Velodyne HDL-64E. Der 3D-LiDAR hat 64 vertikal von $-24,8^\circ$ bis 2° fächerförmig angeordnete Laser und eine Drehrate vom 10 Hz, d. h., eine Umdrehung benötigt 100 ms. Währenddessen findet eine Bewegung des Fahrzeugs statt, was zu einer Verzerrung des Scans führt. Diese Verzerrung kann mittels inertialer Korrektur kompensiert werden, indem der Scan anhand der Eigenbewegungsschätzung des Filters entzerrt wird. Normalerweise wird diese Eigenbewegungsschätzung schritthaltend aus dem Filter herangezogen. Für diese Evaluation wurde sie jedoch vorausberechnet, um eine bessere Vergleichbarkeit der Algorithmen zu gewährleisten.

Die Sensordaten gehen wie in Abschnitt 5.3.1 beschrieben in das SC-EKF ein. Die IMU-Daten haben eine Rate von 100 Hz und werden im Prädiktionsschritt und einem Korrekturschritt für die Lagekorrektur des Nick- und Rollwinkels verwendet. Nach der Lagekorrektur wird auch die nicht-holonome Korrektur angewendet. Die GNSS-Daten haben eine Rate von 4 Hz und gehen als absolute Korrektur für die 3D-Position und die linearen Geschwindigkeiten ein. Zeitgleich wird der Gierwinkel in einem Vorverarbeitungsschritt aus den 2D-Positionsmessungen des GNSS geschätzt und geht als weitere Korrektur mit ein. Die Rad-Odometrie und der 3D-LiDAR haben beide eine Rate von 10 Hz und während die Rad-Odometrie als Messung der Vorwärtsgeschwindigkeit eingeht, wird aus dem 3D-LiDAR-Scan und dem vorhergehenden Scan mittels Scan-Matching eine 6DoF-Differenzpose ermittelt. Diese 6DoF-Differenzpose wird als relative Messung per Korrektur mit stochastischem Klonen integriert. Für das Scan-Matching mittels GICP werden die Scans räumlich unterabgetastet,

da hierdurch der Rechenaufwand deutlich reduziert wird und gleichzeitig die Genauigkeit kaum verringert wird. So kann die Berechnung des Scan-Matching mehr als zwölfmal so schnell erfolgen und die Abweichungen zum Scan-Matching mit dem vollen Scan betragen im Schnitt weniger als 3 cm bzw. $0,06^\circ$. Wie bei der inertialen Korrektur wurde für eine bessere Vergleichbarkeit der Algorithmen die 6DoF-Differenzpose mit dem Scan-Matching für die aufeinanderfolgenden Scan-Paare vorberechnet. Das Scan-Matching beim Schließen von Schleifen wurde dagegen für jeden Ansatz individuell berechnet, da die Schleifenschlüsse auch jeweils unterschiedlich waren.

6.5.1 Implizites Klonen

In Abschnitt 5.2.1 wurde dargestellt, dass die Vereinfachung mit der Rekonstruktion der augmentierten Systemkovarianzmatrix mittels $\mathcal{F}_{k+m,k}$ zwar für eine Konfiguration mit zwei relativen Sensoren gültig ist, jedoch bei der Hinzunahme eines weiteren asynchronen Sensors zu einem instabilen Filterverhalten führen kann.

Dies ist auch in der Evaluation nachweisbar, wie in Abbildung 6.23 zu erkennen ist. Da die Messungen des GNSS nicht zeitgleich mit den relativen Korrekturen des 3D-LiDARs auftreten, können die Schätzungen, welche nur durch die relativen Messungen korrigiert werden, gegenüber den Schätzungen mit den absoluten Korrekturen wegdriften. Da die Korrektur durch das Scan-Matching mit den aufeinanderfolgenden Scans eine hohe Genauigkeit hat, führt dies zu einer hin- und herspringenden Schätzung und letztendlich zu einem instabilen Filter. Wenn z. B. zum Zeitpunkt $k+n$ eine GNSS-Messung als absolute Korrektur in das Filter eingeht, wird nur die aktuelle Lokalisierungsschätzung korrigiert. Eine darauf folgende relative Messung zum Zeitpunkt $k+m$ mit $m > n$ bezieht sich auf die Zustandsschätzung zum Zeitpunkt k , welcher nicht durch die absolute Korrektur zum Zeitpunkt $k+n$ aktualisiert wurde.

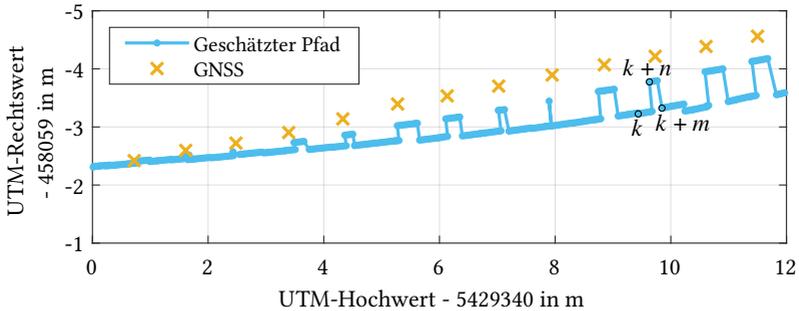


Abbildung 6.23: Divergierende Pfadschätzung mit implizitem Klonen. Der geschätzte Pfad ist in Hellblau dargestellt und die GNSS-Messungen als orange Kreuze.

Abbildung 6.24 zeigt den gleichen Ausschnitt mit der in Abschnitt 5.2.4 vorgestellten Methode mit implizitem Klonen mit Smoothing, welche jeweils vor der Korrektur durch die relative Messung durchgeführt wird. Durch das Smoothing wird die zum Zeitpunkt $k + n$ durch die GNSS-Messung eingebrachte absolute Information zurück zu dem geklonten Zustand des Zeitpunkts k propagiert und dadurch die Drift vermindert und ein Divergieren des Filters unterbunden.

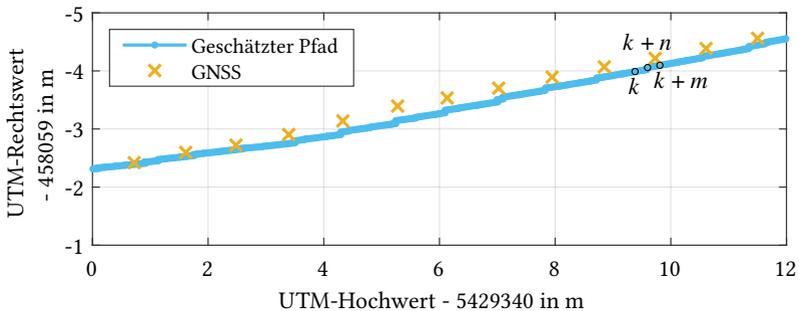


Abbildung 6.24: Pfadschätzung mit implizitem Klonen und Smoothing. Der geschätzte Pfad ist in Hellblau dargestellt und die GNSS-Messungen als orange Kreuze.

Hinsichtlich der Stabilität beim impliziten Klonen mit Smoothing konnte außerdem beobachtet werden, dass es zur Degeneration der Systemkovarianzmatrix kommen kann, wenn nicht das modifizierte kumulative Produkt $\mathcal{F}_{k+m,k}$, sondern lediglich das ursprüngliche kumulative Produkt der Zustandsübergangsmatrizen $\mathcal{F}_{k+m,k}$ verwendet wird. Die Degeneration äußert sich durch negative Diagonalelemente, d. h., die Systemkovarianzmatrix ist dann nicht mehr positiv semidefinit. Dies trat in den Evaluationen allerdings nur auf, wenn für die Messunsicherheit des relativen Sensors extrem niedrige Werte eingesetzt wurden. Mit dem modifizierten kumulativen Produkt $\mathcal{F}_{k+m,k}$ konnten keinerlei Instabilitäten dieser Art beobachtet werden.

6.5.2 Simulierter GNSS-Ausfall

Da wie auch schon bei den Evaluierungen zuvor keine Ground-Truth-Daten von einem Referenzsystem oder Referenzmessungen vorhanden sind, wurde zum Vergleich der Methoden mit stochastischem Klonen ein simulierter GNSS-Ausfall herangezogen. Während des simulierten Ausfalls ist mit einer Verschlechterung der Posenschätzung zu rechnen, da keine absolute Position und kein Gierwinkel zur Stützung zur Verfügung steht. Hiermit kann die in Abschnitt 5.3.4 beschriebene Ausfallsicherheit gezeigt werden und die Methoden können hinsichtlich der auftretenden Verschlechterung miteinander verglichen werden.

Bei der Evaluation wurde ein Ausfall des GNSS für ca. 38 s bzw. über eine Strecke von ca. 130 m simuliert und die sich während des Ausfalls ergebene Drift im Vergleich zu den jeweilig gleichen Methoden ohne den Ausfall des GNSS analysiert. Folgende Methoden wurden für die Evaluation miteinander verglichen:

1. Kein Scan-Matching,
2. Implizites Klonen mit zusätzlicher Offset-Schätzung nach [Lyn13],
3. Implizites Klonen mit Smoothing nach Abschnitt 5.2.4,

4. Partielles implizites Klonen mit Smoothing nach Abschnitt 5.2.5,
5. Explizites Klonen nach Abschnitt 5.2.2,
6. Partielles explizites Klonen nach Abschnitt 5.2.3.

Dabei sind die im Folgenden verwendeten Bezeichnungen unterstrichen. Bei der ersten Methode werden keine relativen Messungen des 3D-LiDARs mit einbezogen, d. h., während des Ausfalls des GNSS stehen als Sensoren nur noch die IMU und die Rad-Odometrie zur Verfügung. Dies stellt den ungünstigsten Fall dar und dient daher zur Orientierung. Die zweite Methode zeichnet sich dadurch aus, dass sie implizites Klonen einsetzt und die in Abschnitten 5.2.1 und 6.5.1 aufgezeigte Drift als 6DoF-Offset zur Kompensation mitschätzt, wodurch der Zustandsraum um sechs zusätzliche Variablen erweitert wird [Lyn13]. Bei der dritten Methode handelt es sich um die neuartige in Abschnitt 5.2.4 vorgestellte Methode mit implizitem Klonen und Smoothing. Bei der vierten Methode handelt es sich um das partielle implizite Klonen nach Abschnitt 5.2.5, wobei es sich bei dem relevanten geklonten Teil um die 6DoF-Pose der Zustandsschätzung handelt, da diese von der relativen 6DoF-Pose des Scan-Matchings betroffen ist. Die fünfte Methode setzt das in Abschnitt 5.2.2 vorgestellte explizite Klonen um. Bei der letzten Methode wird ebenfalls ein explizites Klonen umgesetzt, allerdings werden hier wieder nur die sechs Zustandsvariablen der 6DoF-Pose geklont, vgl. Abschnitt 5.2.3. Alle Methoden wurden im selben Filter-Framework umgesetzt und mit allen wurde der gleiche oben erwähnten Datensatz verarbeitet, um so eine adäquate Vergleichbarkeit sicherzustellen.

Für die Evaluation der Methoden wird eine Fehlermetrik definiert, welche den euklidischen Abstand der geschätzten Position zu einer Referenzposition darstellt und im Folgenden als 3D-Positionsfehler bezeichnet wird:

$$e_{3D,k} = \sqrt{(x_k - x_{\text{ref},k})^2 + (y_k - y_{\text{ref},k})^2 + (z_k - z_{\text{ref},k})^2}, \quad (6.1)$$

wobei die Komponenten der jeweiligen Referenzposition ohne GNSS-Ausfall mit dem Index „ref“ gekennzeichnet sind. Da in der Zustandsschätzung die 2D-Position in Längen- und Breitengrad repräsentiert ist, wird sie in UTM-Koordinaten konvertiert, um zusammen mit der Höhe den 3D-Positionsfehler in kartesischen Koordinaten berechnen zu können.

Abbildung 6.25 zeigt die 3D-Positionsfehler im Zeitraum um den GNSS-Ausfall. Der Zeitraum des GNSS-Ausfalls geht von 18 s bis 56 s und ist grau

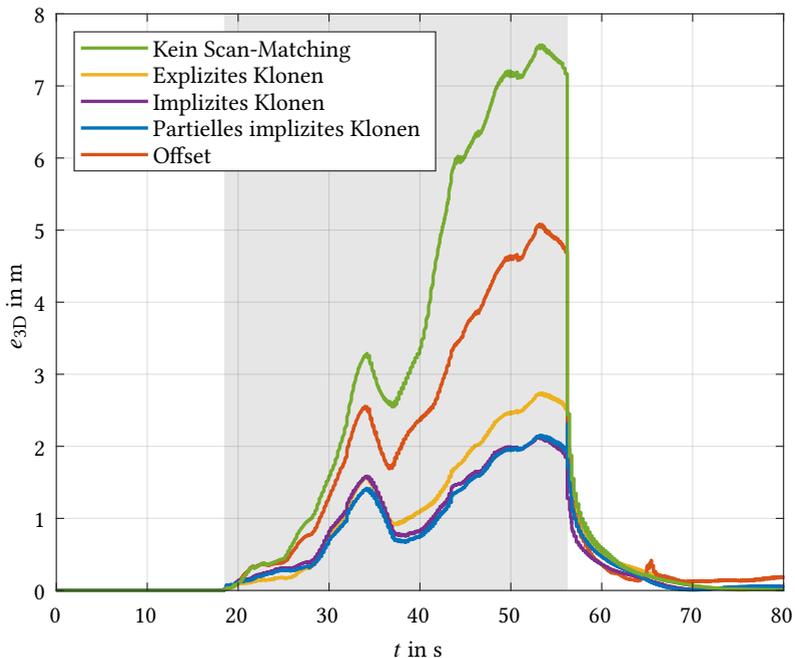


Abbildung 6.25: 3D-Positionsfehler im Zeitraum um den GNSS-Ausfall. Der Zeitraum des GNSS-Ausfalls ist grau hinterlegt.

hinterlegt. Vor dem GNSS-Ausfall ist der Fehler aller Varianten 0 m, da sie gegen sich selbst verglichen werden. Sobald das GNSS ausfällt, steigen bei allen Varianten die Fehler mit ähnlicher Tendenz und werden erwartungsgemäß schnell wieder reduziert, wenn das GNSS wieder verfügbar

ist. Die Methode ohne Scan-Matching weist wie zu erwarten den größten maximalen Fehler von über 7 m auf. Die Offset-Methode ist mit einem maximalen Fehler von um die 5 m besser, jedoch haben die Methoden mit explizitem Klonen und implizitem Klonen deutlich geringere maximale Fehler von unter 3 m bzw. um die 2 m. Das Ergebnis der Methode des partiellen expliziten Klonens ist nicht in die Abbildung mit aufgenommen, da es eine maximale Differenz von weniger als 1 mm zur Methode des kompletten expliziten Klonens aufweist, d.h., die in Abschnitt 5.2.3 diskutierten Einflüsse durch die Kreuzkovarianzen sind hier sehr gering. Während des GNSS-Ausfalls hat das partielle implizite Klonen eine maximale Differenz von 17 cm zum kompletten impliziten Klonen. Der Unterschied zwischen dem partiellen und dem kompletten impliziten Klonen ist im Vergleich zu den expliziten Varianten also deutlich größer, was auf die zusätzlichen Approximationen bei der Rekonstruktion der augmentierten Systemkovarianzmatrix beim partiellen impliziten Klonen zurückzuführen ist, vgl. Abschnitt 5.2.5.

Abbildung 6.26 zeigt die Fehler des Gierwinkels ψ im Zeitraum um den GNSS-Ausfall. Die horizontalen Lagewinkel wurden nicht verglichen, da sie auch während des GNSS-Ausfalls über die Beschleunigungssensoren korrigiert werden. Vor dem GNSS-Ausfall ist der Fehler aller Varianten wieder 0° , da sie gegen sich selbst verglichen werden. Sobald der Ausfall einsetzt, sind für alle Varianten tendenziell ähnliche Verläufe mit unterschiedlichen Amplituden zu beobachten. Wie bei der Position ergeben sich ohne Scan-Matching während des Ausfalls die größten maximalen Fehler von mehr als $\pm 10^\circ$. Die Offset-Methode ist mit einem maximalen Fehler von unter $\pm 7,5^\circ$ etwas besser. Die Methoden mit explizitem Klonen und implizitem Klonen weisen nochmals deutlich geringere maximale Fehler von unter $\pm 2,5^\circ$ auf. Zusammen mit dem Ergebnis des Vergleichs der 3D-Positionsfehler in Abbildung 6.25 ist festzustellen, dass die Methoden mit explizitem Klonen bzw. implizitem Klonen eine vergleichbare Genauigkeit erzielen. Das partielle explizite Klonen ist wiederum nicht in die Abbildung mitaufgenommen, da es eine maximale Differenz von weniger als

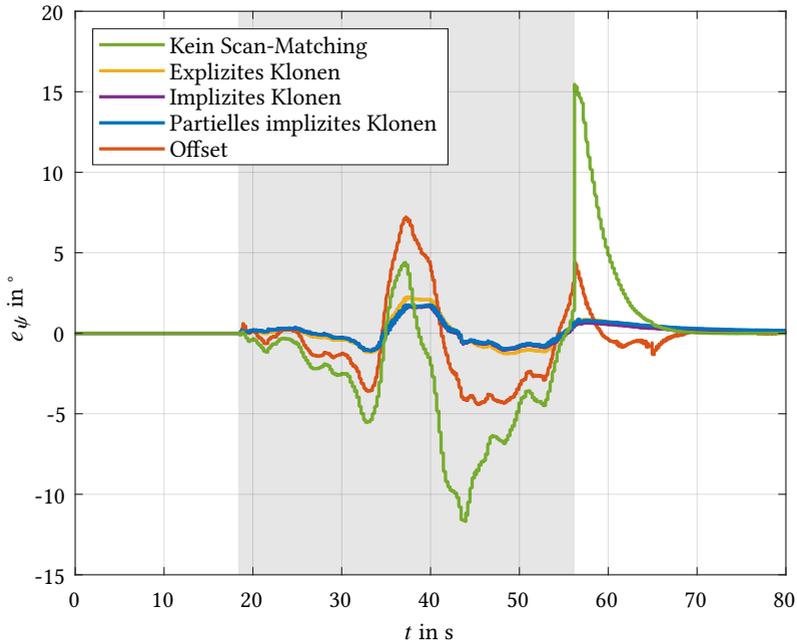


Abbildung 6.26: Fehler des Gierwinkels im Zeitraum um den GNSS-Ausfall. Der Zeitraum des GNSS-Ausfalls ist grau hinterlegt.

0,01° zum kompletten expliziten Klonen aufweist. Das partielle implizite Klonen hat im Vergleich dazu wiederum eine höhere maximale Differenz von 0,17° zum kompletten impliziten Klonen.

Der große Überschwinger beim Wiedereinsetzen des GNSS von mehr als 15° bei der Variante ohne Scan-Matching kann damit erklärt werden, dass zu diesem Zeitpunkt der Gierwinkel noch nicht durch absolute Messungen gestützt wird, da die vorgelagerte Gierwinkelschätzung mehrere GNSS-Messungen benötigt. Die IMU-Messungen liefern lediglich eine Information über die Orientierungsänderung und durch die nicht-holonome Korrektur wird der Gierwinkel deshalb in Richtung der GNSS-Messung gezogen und erst langsam korrigiert, sobald die Gierwinkelschät-

zung mittels GNSS-Messungen kurze Zeit später wieder zur Verfügung steht. Die Filtervarianten mit dem stochastischen Klonen haben noch zusätzlich die sehr genauen relativen Posen des Scan-Matching, welche stärkere Orientierungsänderungen verhindern. Außerdem fallen die notwendigen Positionskorrekturen nicht so stark aus, da die Abweichungen zum Zeitpunkt des Wiedereinsetzens des GNSS nicht so groß sind wie ohne Scan-Matching, vgl. Abbildung 6.25.

Zeitlicher Verlauf der Systemkovarianzen

In diesem Abschnitt werden ausgesuchte zeitliche Verläufe von Elementen der Systemkovarianzmatrizen der unterschiedlichen Methoden miteinander verglichen.

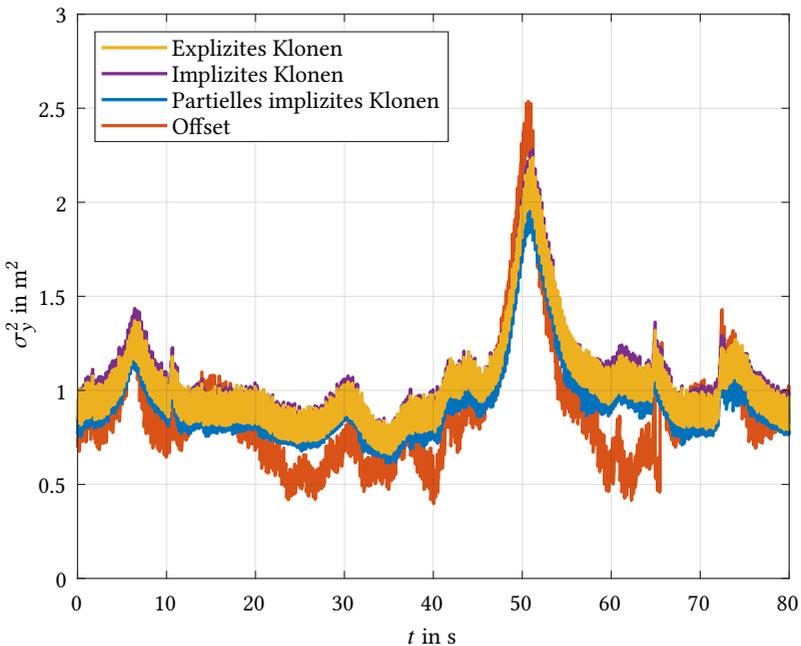


Abbildung 6.27: Autokovarianzen des Hochwerts.

In Abbildung 6.27 sind die Autokovarianzen des Hochwerts y der verschiedenen Methoden im zeitlichen Verlauf ohne Ausfall des GNSS dargestellt und die Methoden mit explizitem Klonen und implizitem Klonen weisen nahezu identische Verläufe auf. Die Methode mit partiellem implizitem Klonen hat tendenziell auch einen sehr ähnlichen Verlauf, jedoch mit einem stets etwas abweichenden Wert, was darauf hindeutet, dass die Kreuzkovarianzen zu den nicht geklonten Zustandsvariablen einen nicht vernachlässigbaren Einfluss haben. Der Ansatz mit dem Offset zeigt deutlichere Abweichungen zu den anderen Methoden, was daran liegt, dass die Lokalisierungsunsicherheit zum Teil auch in dem zusätzlichen 6DoF-Offset modelliert ist. Die anderen Autokovarianzen der Zustandsvariablen der Pose zeigen ähnliche Verhältnisse.

Abbildung 6.28 zeigt die Kreuzkovarianzen zwischen Hochwert und Gierwinkel. Die Methoden mit explizitem Klonen und implizitem Klonen haben wiederum nahezu identische Verläufe, was zusammen mit der Betrachtung der Autokovarianzen in Abbildung 6.27 zeigt, dass das Zurückpropagieren der Information durch Smoothing beim impliziten Klonen nahezu äquivalente Ergebnisse erzielt, während die zusätzlichen Approximationen beim partiellen impliziten Klonen wieder zu leichten Abweichungen führen. Der Ansatz mit dem Offset hingegen zeigt wieder deutlichere Abweichungen.

In Abbildung 6.29 sind die Varianzen der 3D-Position im Zeitraum um den GNSS-Ausfall dargestellt. Zum Vergleich ist zusätzlich die Varianz der 3D-Position der Methode mit explizitem Klonen ohne GNSS-Ausfall abgebildet. Die Varianz der 3D-Position ist dabei die Varianz der Summe der Komponenten und wird über das Aufsummieren der jeweilige Auto- und Kreuzkovarianzen von Rechtswert x , Hochwert y und Höhe z berechnet:

$$\sigma_{3D}^2 = \sum_i \sigma_i^2 + 2 \sum_{i < j} \sigma_{ij} \quad \text{mit } i, j \in \{x, y, z\}. \quad (6.2)$$

Es ist deutlich zu erkennen, wie die Varianz während des Ausfalls bei allen Methoden wie erwartet stetig ansteigt, während sich die Varianz ohne

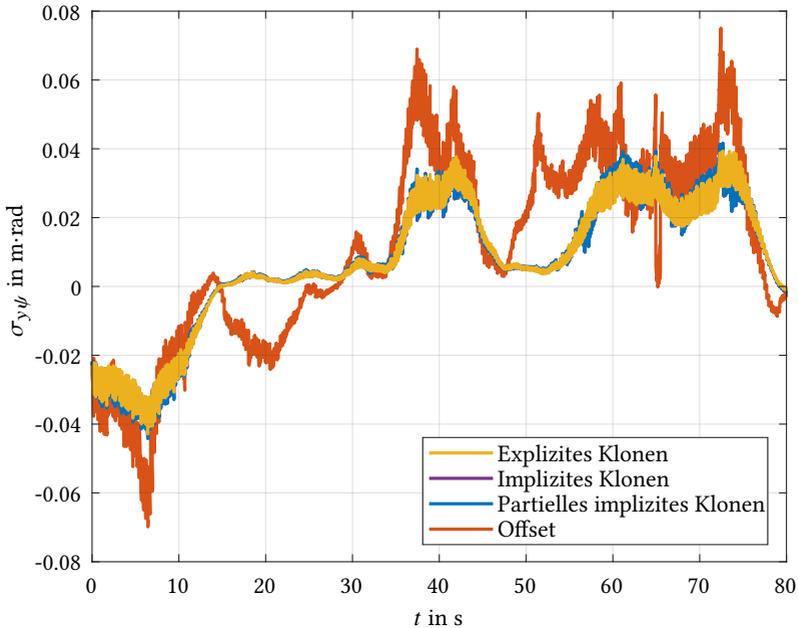


Abbildung 6.28: Kreuzkovarianzen zwischen Hochwert und Gierwinkel.

GNSS-Ausfall im Vergleich kaum ändert. Sobald das GNSS wieder verfügbar ist, werden die Unsicherheiten durch die absoluten Messungen in kurzer Zeit wieder reduziert. Der stetige Anstieg während des Ausfalls wird durch die Integration des Scan-Matching als relativer Sensor durch stochastisches Klonen gewährleistet und die Abweichungen in Abbildung 6.25 sind dabei stets kleiner als die Quadratwurzel der jeweiligen Varianz, d. h., die Varianzen der Filter werden nicht zu optimistisch berechnet. So können die absoluten Messungen des GNSS bei Verfügbarkeit auch die Schätzung in kurzer Zeit korrigieren, wie in Abbildungen 6.25 und 6.26 gezeigt wurde. Bei einer Modellierung der relativen Messungen beispielsweise als pseudo-absolute Messungen wäre die Varianz nicht angestiegen. D. h., die Messungen des GNSS hätten bei Wiederverfügbarkeit einen nur geringen Einfluss

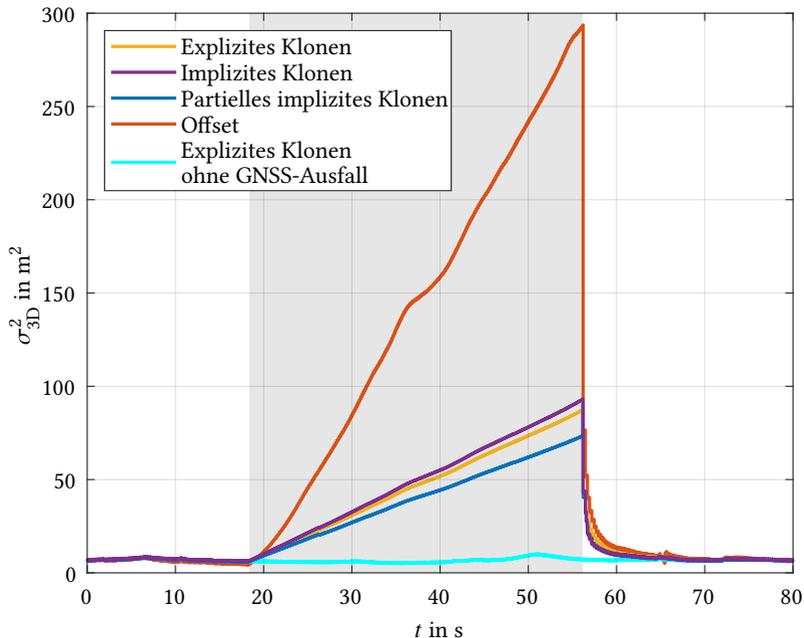


Abbildung 6.29: Varianzen der 3D-Position im Zeitraum um den GNSS-Ausfall. Der Zeitraum des GNSS-Ausfalls ist grau hinterlegt.

gehabt und somit hätte die Korrektur der Schätzung sehr viel länger gedauert.

Qualitativer Vergleich

Wie bereits in Abschnitt 5.3.1 findet bei der Einbeziehung des 3D-LiDARs als relative 6DoF-Bewegung über Scan-Matching keine explizite Kartierung statt. Für einen qualitativen Vergleich anhand von erstellten Karten können diese dennoch aufgezeichnet werden, ohne sie jedoch für die Lokalisierung zu verwenden. Als Kartenformat kommt hier die in Abschnitt 5.3.2 vorgestellte NDT-Karte mit inkrementeller Aktualisierung zum Einsatz. Während für das Scan-Matching zur Lokalisierung unterab-

getastete Scans Verwendung finden, werden für die Kartierung die vollen Scans verwendet.

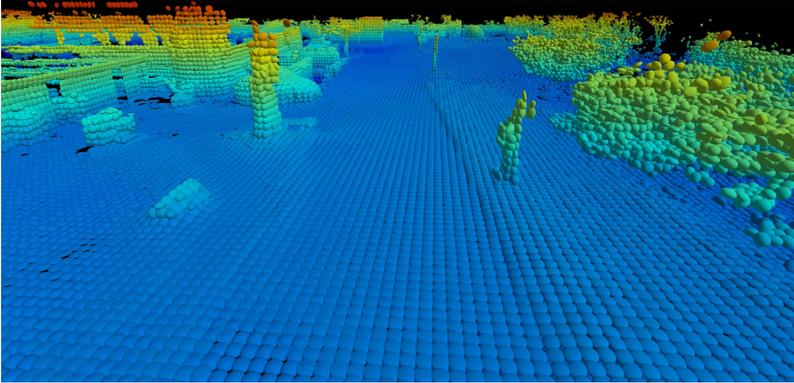


Abbildung 6.30: Beispiel einer NDT-Karte. Die Höhe ist farbkodiert von blau (niedrig) über grün zu rot (hoch).

Abbildung 6.30 zeigt ein Beispiel für eine NDT-Karte einer gemischten Umgebung. Auf der linken Seite befinden sich Gebäudestrukturen, bei denen Ellipsoide vorherrschen, welche flach und vertikal ausgerichtet sind. In der Mitte ist eine Straße mit Ellipsoiden, welche wie zu erwarten flach und horizontal ausgerichtet sind. Auf der rechten Seite ist ein unstrukturierter Bereich mit Vegetation in Form von Bäumen und Büschen, erkennbar an unregelmäßig verteilten Ellipsoiden, welche in alle Richtungen ähnliche Ausdehnungen ohne Vorzugsrichtung aufweisen, vgl. Abschnitt 3.2.4.

Abbildung 6.31a zeigt einen Ausschnitt aus der Karte, welche mit explizitem Klonen ohne GNSS-Ausfall aufgenommen wurde. Diese dient als Referenzkarte, da hier alle zur Verfügung stehenden Sensorinformationen eingegangen sind. In der Mitte des Ausschnitts steht ein Block von Garagen, auf der linken Seite befindet sich eine lange gerade Hecke und auf der rechten Seite sind Teile eines großen Gebäudes sichtbar. In den anderen Karten ist der gleiche Ausschnitt für die verschiedenen Methoden mit stochastischem Klonen bei GNSS-Ausfall gegenübergestellt. Abbil-

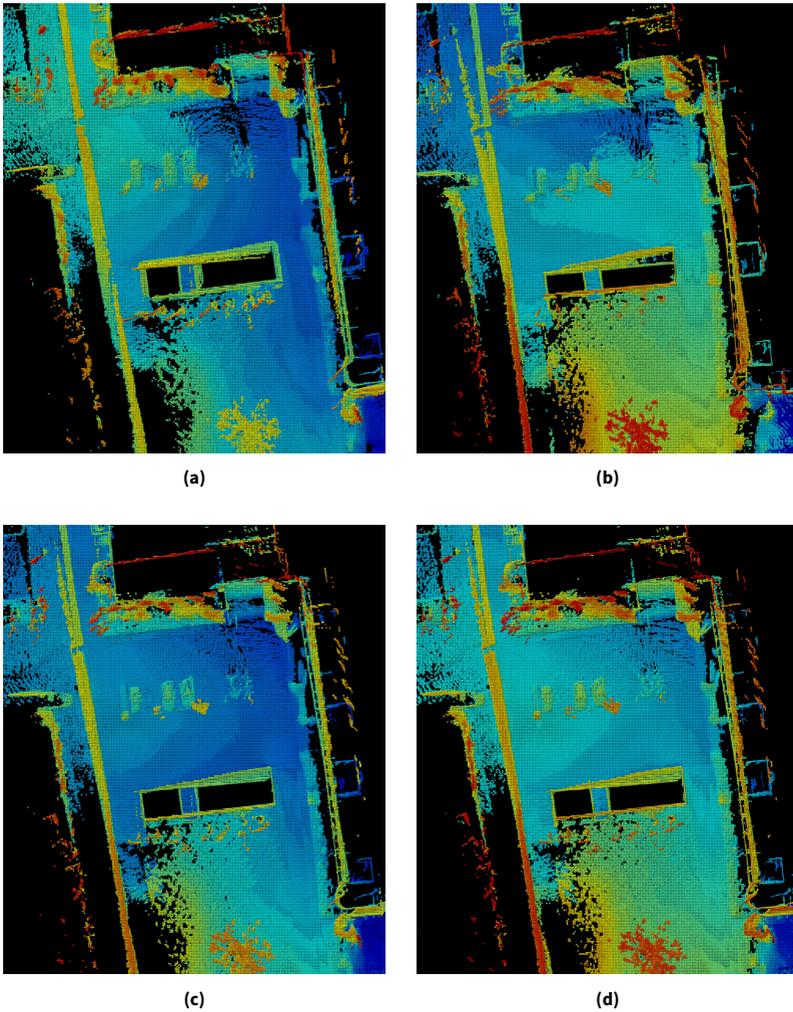


Abbildung 6.31: Kartierungsergebnis ohne GNSS-Ausfall mit explizitem Klonen (a), bei GNSS-Ausfall mit Offset (b), bei GNSS-Ausfall mit implizitem Klonen (c), bei GNSS-Ausfall mit explizitem Klonen (d). Die Höhe ist farbkodiert von blau (niedrig) über grün zu rot (hoch).

Abbildung 6.31b zeigt das Kartierungsergebnis der Methode mit Offset und es sind deutliche Ungenauigkeiten an den Garagen und der Hecke in Form von Verdrehungen zu erkennen. In den Abbildungen 6.31c und 6.31d, den Kartierungsergebnissen mit implizitem Klonen bzw. explizitem Klonen, sind zwar auch Ungenauigkeiten an den gleichen Stellen zu erkennen, sie sind jedoch wesentlich weniger ausgeprägt. Diese Beobachtungen decken sich auch mit den Ergebnissen in den Abbildungen 6.25 und 6.26, bei denen die Methode mit Offset ebenfalls höhere Fehler aufweist.

6.5.3 Vergleich mit Faktor-Graph-SLAM

Um trotz der fehlenden Referenzlokalisierung eine Vergleichsmöglichkeit gegenüber einer einzigen Referenz zu realisieren, wurde ein SLAM-Algorithmus auf Basis eines Faktor-Graphen implementiert. Faktor-Graph-Methoden gehören momentan zu den leistungsfähigsten SLAM-Algorithmen, haben jedoch den Nachteil, dass sie sich nur bedingt für Echtzeitanwendungen eignen, siehe Abschnitt 2.2.1. Bei den folgenden Untersuchungen ist auch die Methode für das Schließen von Schleifen Teil der Evaluation. Da beim Schließen von Schleifen das Scan-Matching auch mit zurückliegenden Scans durchgeführt wird, müssen diese auch gespeichert werden. Für das Scan-Matching zur Lokalisierung finden jedoch nur unterabgetastete Scans Verwendung, weshalb im realen Einsatz nur diese und nicht die vollen Scans abgespeichert werden müssten, da letztere nur für die Kartierung Verwendung finden, welche inkrementell erfolgt, vgl. Abschnitt 5.3.2.

Der Faktor-Graph-SLAM-Algorithmus wurde mit dem GTSAM-Framework implementiert. Dieses bietet eine flexible Faktor-Graph-Struktur, einige vordefinierte Faktoren für die Lokalisierung und Kartierung und verschiedene Optimierungsalgorithmen [GTSAM]. Zusätzlich bietet das Framework noch einen speziellen Faktor für die Integration von IMU-Messungen. Da die IMU-Messungen mit einer hohen Rate zur Verfügung stehen, werden mehrere aufeinanderfolgende Messungen in einem IMU-

Faktor vorab integriert, damit die Anzahl der benötigten Knoten und damit auch der Rechenaufwand reduziert wird [Car14; For15]. Wie auch in Abschnitt 5.3.4 erläutert, kann die hohe Rate der IMU im Vergleich zu den anderen Sensoren zudem für die Synchronisierung verwendet werden. Die Vorab-Integration mit dem IMU-Faktor wird immer bis zur nächsten Messung eines beliebigen anderen Sensors durchgeführt. Da diese Messung meist nicht exakt auf die IMU-Messungen fällt, wird die letzte Vorab-Integration extrapoliert und das Integrationsintervall der nachfolgenden IMU-Messung entsprechend verkürzt. Die Messungen des GNSS gehen als Positions- und als Geschwindigkeitsfaktor ein, die Rad-Odometrie als 2D-Zwischenfaktor. Die 3D-LiDAR-Messungen gehen über Scan-Matching sowohl zwischen aufeinanderfolgenden Posen als auch für das Schließen von Schleifen als 3D-Zwischenfaktor mit ein, d. h., auch hier findet keine explizite Kartierung statt [Emt18a].

Zusätzlich zur Graph-Optimierung über den gesamten Pfad-Posterior ist es mit GTSAM auch möglich, jederzeit eine Schätzung mit den bisher verfügbaren Informationen durchzuführen. Daher wurden mit dem Faktor-Graph zusätzlich zwei inkrementelle Lokalisierungslösungen berechnet und in den Vergleich mit aufgenommen. Eine der Lösungen wird mittels einer Levenberg-Marquardt-Optimierung berechnet. Diese wird auch für die Berechnung der Referenz über den gesamten Graphen verwendet und muss für jede Zwischenlösung erneut ausgeführt werden. Die zweite Lösung wird mit dem iSAM2-Algorithmus berechnet, welcher inkrementelle Aktualisierungen durchführt, d. h. Lösungen von vorherigen Durchläufen weiter nutzen kann und im Idealfall nur kleine Teile des Graphen optimiert [Kae12].

Wie bereits in der Einleitung des Abschnitts 6.5 erwähnt, wurde der Datensatz in einem Mischgebiet aufgenommen. Der abgefahrte Pfad hat eine Länge von ca. 1,8 km, deckt sowohl urbane Strukturen als auch unstrukturierte Bereiche mit Vegetation ab und beinhaltet mehrere größere und kleinere Schleifen. Das Ergebnis des Faktor-Graph-SLAM-Algorithmus mit einer Optimierung über den vollen Pfad-Posterior ist in

Abbildung 6.32 dargestellt. Da die Erstellung der Karte hier erst mittels des finalen Pfad-Posteriors durchgeführt wird, sind für die Referenzkarte alle Scans zu speichern und es genügt nicht, wie bei den Filtern nur die unterabgetasteten Scans zu speichern. Die mit den optimierten Posenschätzungen

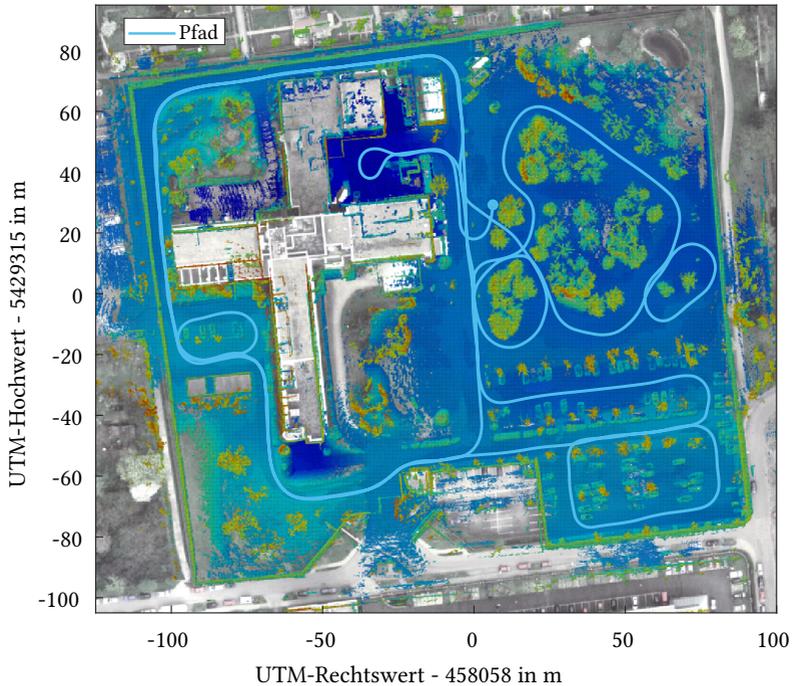


Abbildung 6.32: Kartierungsergebnis des Faktor-Graph-SLAM-Algorithmus mit einer Optimierung über den vollen Pfad-Posterior. Die Karte ist über ein Satellitenbild gelegt und der Pfad ist in Hellblau dargestellt. Die Höhe ist farbkodiert von blau (niedrig) über grün zu rot (hoch).

erzeugte NDT-Karte ist in der Abbildung über ein Satellitenbild gelegt und der Pfad ist in Hellblau dargestellt. Dieser Pfad und die Karte dienen im Folgenden als Referenz für die Evaluierung der unterschiedlichen Methoden mit stochastischem Klonen.

Für eine übersichtlichere Darstellung werden die unterschiedlichen Methoden mit stochastischem Klonen in den kommenden Evaluationen folgendermaßen abgekürzt: Offset (O), explizites Klonen (EK), partielles explizites Klonen (PEK), implizites Klonen mit Smoothing (IK) und partielles implizites Klonen mit Smoothing (PIK), mit (mS) oder ohne dem Schließen von Schleifen nach Abschnitt 5.3.3. Die Abkürzungen für die inkrementellen Faktor-Graph-Methoden sind Levenberg-Marquardt-Optimierung (FGLM) und iSAM2 (FGISAM), wobei diese beiden immer inklusive des Schließens von Schleifen sind.

Abbildung 6.33 zeigt die 3D-Positionsfehler aller Methoden mit stochastischem Klonen und der beiden inkrementellen Faktor-Graph-Methoden bezogen auf den global optimierten Faktor-Graphen. Die Zeitpunkte, an denen Schleifen zu schließen sind, werden durch vertikale gestrichelte Linien angezeigt. Die partiellen Methoden (PEK und PIK) wurden nicht in die Abbildungen übernommen, da sie nur wenige Millimeter bzw. Zentimeter Differenzen zu den jeweiligen Methoden mit komplettem Klonen aufwiesen. Die Schätzung des GNSS-Offsets nach Abschnitt 5.3.3 bei den Ansätzen mit stochastischem Klonen wird erst nach dem ersten Schleifenschluss aktiviert, d. h., die Ergebnisse der Paare mit und ohne Schleifenschluss sind davor identisch. Vor dem ersten Schleifenschluss weisen alle Methoden sehr hohe Fehler auf, insbesondere die Faktor-Graph-Methoden. Letzteres deutet darauf hin, dass die Faktor-Graph-Methoden mehr Verknüpfungen benötigen, um besser zu konvergieren. Nach dem Schleifenschluss sind bei allen Methoden die Fehler unter 4 m und zwischen den Methoden mit stochastischem Klonen ist kein signifikanter Unterschied zu erkennen. Es zeigen sich zwar etwas andere Verläufe, aber keine Vorteile durch die Hinzunahme des GNSS-Offsets für das Schließen von Schleifen. Dies ist auch nicht unbedingt zu erwarten, da bei diesem Ansatz der GNSS-Offset die Lokalisierung der späteren Durchläufe zu der Lokalisierung der ersten Schleife hinzieht und nicht wie bei den Faktor-Graph-Methoden die Korrektur auch rückwirkend erfolgt. D. h., wenn die erste Schleife einen größeren Fehler gegenüber der Referenz aufweist, zeigt sich mit dieser

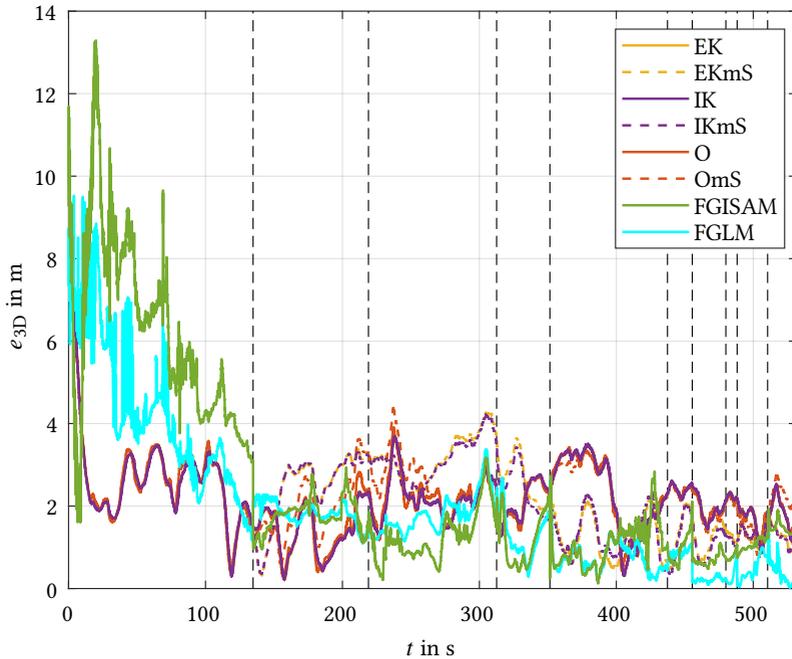


Abbildung 6.33: 3D-Positionsfehler bezogen auf den global optimierten Faktor-Graphen.

Fehlermetrik im späteren Verlauf ein höherer Fehler. Der Fehler der FGLM-Methode geht am Ende auf 0, da der gleiche Optimierungsalgorithmus wie bei der Referenz zum Einsatz kommt. Bei der FGISAM-Methode und den Filtern mit stochastischem Klonen ist am Ende ein Fehler von etwa 1 m zu beobachten.

Da in Abbildung 6.33 die Unterschiede zwischen den Methoden nicht direkt sichtbar sind, ist in Abbildung 6.34 eine Violinengrafik des 3D-Positionsfehlers gezeigt. Bei der Violinengrafik handelt es sich um eine Erweiterung der Kastengrafik, welche zusätzlich zu den Quartilen (schwarzes Rechteck) und dem Median (weißer Balken) ein geglättetes Histogramm als Dichterepräsentation enthält [Hin98]. Hier lässt sich erkennen, dass

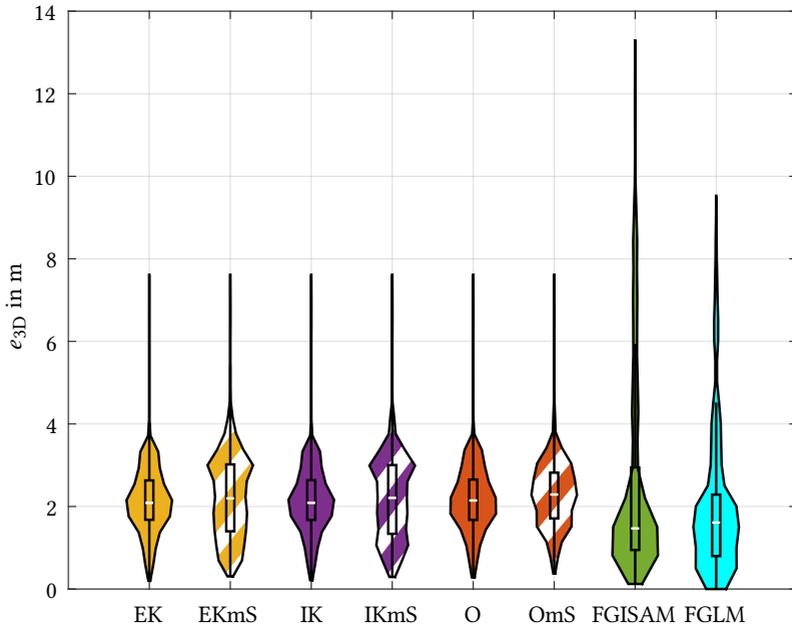


Abbildung 6.34: Violinengrafik des 3D-Positionsfehlers.

sich die Dichten der 3D-Positionsfehler der Methoden mit explizitem (EK) und implizitem Klonen (IK) ohne Schleifenschluss kaum unterscheiden, wohingegen die Methode mit dem Offset (O) hier eine etwas andere Form aufweist. Mit Schleifenschluss ändern sich die Formen der Dichten bei den Methoden mit explizitem (EKmS) und implizitem Klonen (IKmS) und es ist ein leichter Unterschied zwischen beiden zu erkennen. Die Methode mit dem Offset (OmS) ändert sich hingegen mit dem Schleifenschluss kaum und unterscheidet sich daher deutlich von den anderen Methoden. Allgemein zeigt sich auch, dass hier mit dem Schleifenschluss keine Verbesserung erzielt wurde, vor allem bei der Methode mit dem Offset ist kaum eine Veränderung zu erkennen. Beide inkrementellen Faktor-Graph-Methoden weisen gegenüber den Filtern wesentlich höhere Ausreißer auf, der Haupt-

teil der 3D-Positionsfehler ist jedoch hin zu kleineren Werten verschoben und sie weisen geringere Medianwerte auf.

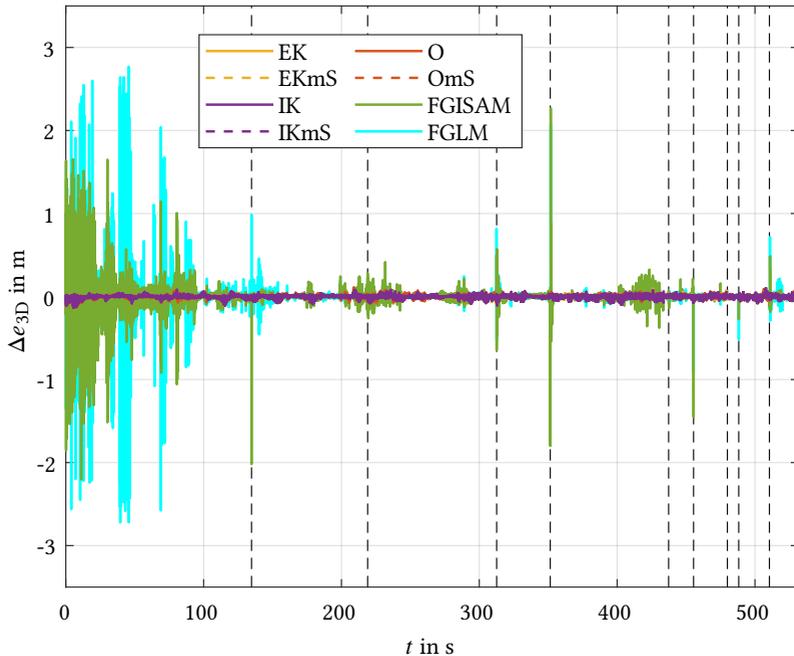


Abbildung 6.35: Änderung des 3D-Positionsfehlers.

Für die Lokalisierung kann neben der absoluten Genauigkeit auch die Glattheit eine bevorzugte Eigenschaft sein, d. h., die Lokalisierung sollte möglichst frei von Sprüngen bzw. Diskontinuitäten sein. Hierfür wurden die Änderungen der 3D-Positionsfehler evaluiert und in Abbildung 6.35 dargestellt. Die Zeitpunkte, an denen Schleifen zu schließen sind, werden wieder durch vertikale gestrichelte Linien angezeigt. Die Fehler der Filter mit stochastischem Klonen ändern sich in einem Zeitschritt nicht mehr als ± 15 cm, bei den Faktor-Graph-Methoden hingegen sind einzelne Spitzen bis zu mehreren Metern zu beobachten. Diese Spitzen treten meist

beim Schließen von Schleifen auf, was darauf zurückzuführen ist, dass zu diesen Zeitpunkten zusätzliche Bedingungen in den Graphen eingefügt werden, die große Bereiche des Graphen und damit die Gesamtlösung beeinflussen. Das Auftreten größerer Sprünge bei inkrementellen Faktor-Graph-Methoden konnte auch in anderen Veröffentlichungen beobachtet werden [Wil14; Emt18a].

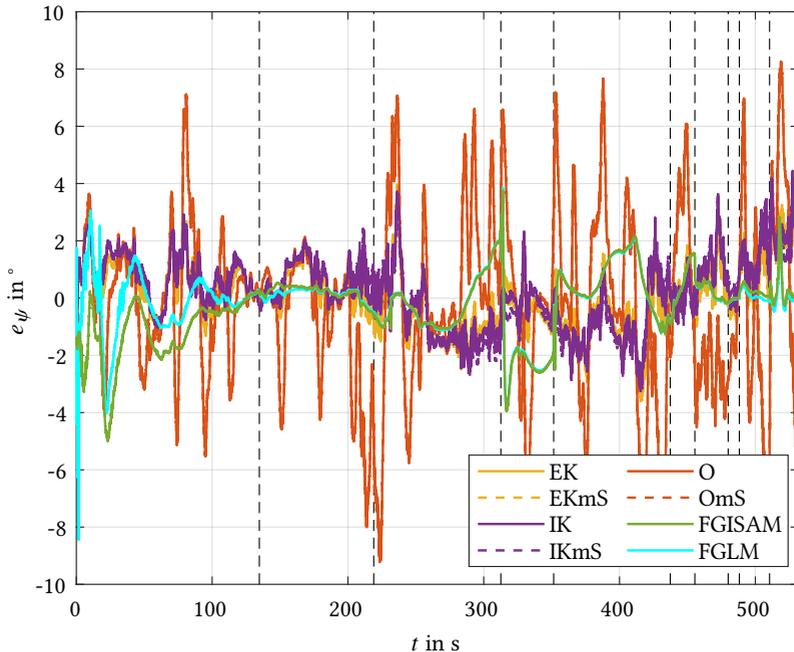


Abbildung 6.36: Gierwinkelfehler.

In Abbildung 6.36 sind die Gierwinkelfehler dargestellt. Auch hier sind die partiellen Methoden PEK und PIK nicht mit abgebildet, da sie lediglich Unterschiede kleiner $0,003^\circ$ bzw. $0,1^\circ$ zu den jeweiligen Methoden mit den kompletten Klonen aufwiesen. Die Filter-Methoden haben bis auf die Offset-Methode (O) einen nahezu gleichen Verlauf mit Fehlern unter $\pm 4^\circ$. Die inkrementellen Faktor-Graph-Methoden weisen vor dem ersten Schlei-

fenschluss ein paar Ausreißer auf und sind danach ebenfalls unter $\pm 4^\circ$. Lediglich die Offset-Methode zeigt durchweg sehr hohe Fehler bis über $\pm 8^\circ$. Generell ist hier erkennbar, dass die Fehler der inkrementellen Faktor-Graph-Methoden bis auf den Anfang und einzelne Sprünge beim Schließen von Schleifen einen glatteren Verlauf aufweisen als die Filter-Methoden, was auf einen glättenden Effekt der Vorab-Integration der IMU-Messungen in dem speziellen IMU-Faktor schließen lässt.

Die Präzision von GNSS-Messungen ist in der Höhe i. A. schlechter als in der horizontalen Ebene [Kou17]. Während die Strecke, welche bei der Aufnahme des Datensatzes abgefahren wurde, nahezu flach ist und nur einen Innenhof beinhaltet, welcher etwa 2 m niedriger liegt, war in den GNSS-Messungen eine Differenz zwischen dem höchsten und dem niedrigsten Wert von 11,5 m zu beobachten. Die Lokalisierungslösung der Referenz weist eine Differenz von 2,4 m in der Höhe auf, was sehr gut mit der Realität übereinstimmt, während die Filter und die inkrementellen Faktor-Graph-Methoden eine Differenz von mehr als 10 m aufweisen. Hier zeigt sich im Vergleich ein Vorteil durch die Integration von Schleifenschlüssen bei den Filtern, welche mit Schleifenschluss durchweg geringere Differenzen zwischen der minimalen und maximalen Höhe aufweisen, siehe Tabelle 6.4. Dabei fallen die Verbesserungen bei den Methoden mit explizitem bzw. implizitem Klonen mit Smoothing wesentlich deutlicher aus als bei der Methode mit Offset.

Tabelle 6.4: Maximale Höhendifferenz in m.

	Ohne Schleifenschluss	mit Schleifenschluss (mS)
EK	12,1	10,4
PEK	12,1	10,4
IK	12,1	10,5
PIK	12,1	10,5
O	12,1	11,6
FGISAM	-	14,3
FGLM	-	11,8

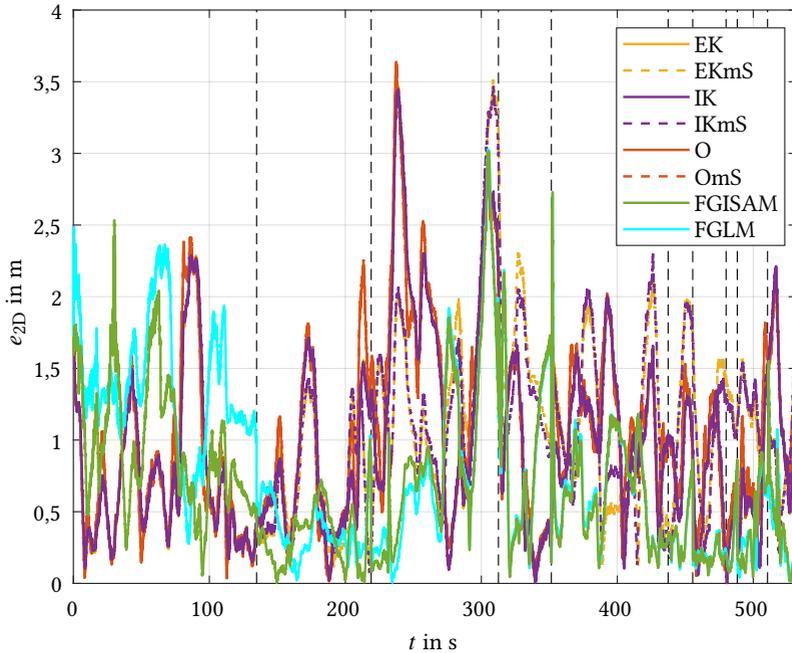


Abbildung 6.37: 2D-Positionsfehler.

Da beim GNSS in der Höhe sehr große Fehler auftreten und sich diese auch auf die Lokalisierungsschätzungen auswirken, wird zusätzlich die horizontale Abweichung, d. h. der 2D-Positionsfehler e_{2D} , ausgewertet und in Abbildung 6.37 dargestellt. Die Schleifenschlüsse sind wieder durch vertikale gestrichelte Linien dargestellt. Die 2D-Positionsfehler sind erwartungsgemäß wesentlich geringer als die 3D-Positionsfehler. Sie liegen größtenteils unter 2,5 m mit ein paar Spitzen bis ca. 3,5 m.

In Tabelle 6.5 sind alle mittleren und maximalen 2D- und 3D-Positionsfehler zum Vergleich aufgelistet. Die mittleren 2D-Positionsfehler sind durchweg weniger als halb so groß wie die mittleren 3D-Positionsfehler. Die Positionsfehler der unterschiedlichen Filtermethoden unterscheiden sich nur geringfügig, im Vergleich zu den inkrementellen Faktor-Graph-

Methoden ergeben sich jedoch einige Unterschiede. Die maximalen 3D-Positionsfehler der Faktor-Graph-Methoden, vor allem der FGLM-Lösung, sind viel höher als die der Filtermethoden, wohingegen die mittleren 3D-Positionsfehler ähnlich sind.

Tabelle 6.5: Mittlerer und maximaler 3D-Positionsfehler in m.

	\bar{e}_{3D}	$\max(e_{3D})$	\bar{e}_{2D}	$\max(e_{2D})$
EK	2,15	7,62	0,97	3,47
EKmS	2,24	7,62	1,03	3,53
PEK	2,15	7,61	0,97	3,47
PEKmS	2,24	7,61	1,03	3,53
IK	2,15	7,62	0,98	3,45
IKmS	2,21	7,62	1,02	3,47
PIK	2,16	7,62	0,99	3,42
PIKmS	2,20	7,62	0,98	3,47
O	2,18	7,62	1,03	3,64
OmS	2,31	7,62	1,03	3,62
FGISAM	2,50	13,29	0,67	3,01
FGLM	1,96	9,51	0,80	3,03

Wie bereits angesprochen, ist nicht zu erwarten, dass das Schließen von Schleifen durch einen GNSS-Offset den absoluten Fehler in der Lokalisierung verringert, da die Lokalisierung in späteren Durchläufen zu der Lokalisierung der ersten Schleife hingezogen wird. Es hat sich jedoch gezeigt, dass durch die Integration von Schleifenschlüssen eine Reduktion der maximalen Höhendifferenzen erreicht wird, siehe Tabelle 6.4. Deshalb wurde noch eine Evaluation mit einer alternativen Fehlermetrik durchgeführt, welche von den Einzelkomponenten des 3D-Positionsfehler jeweils deren mittleren Fehler abzieht, um den Einfluss von etwaig größeren Fehlern während der ersten Schleife zu verringern. Der resultierende relative 3D-Positionsfehler hat eine im Vergleich zu Gleichung (6.1) modifizierte

Berechnung:

$$\tilde{e}_{3D,k} = \sqrt{(x_k - x_{\text{ref},k} - \bar{e}_x)^2 + (y_k - y_{\text{ref},k} - \bar{e}_y)^2 + (z_k - z_{\text{ref},k} - \bar{e}_z)^2}, \quad (6.3)$$

mit

$$\bar{e}_x = \frac{1}{K} \sum_{k=1}^K (x_k - x_{\text{ref},k}), \quad (6.4)$$

$$\bar{e}_y = \frac{1}{K} \sum_{k=1}^K (y_k - y_{\text{ref},k}), \quad (6.5)$$

$$\bar{e}_z = \frac{1}{K} \sum_{k=1}^K (z_k - z_{\text{ref},k}). \quad (6.6)$$

Abbildung 6.38 zeigt die Violinengrafik für den relativen 3D-Positionsfehler. Im Gegensatz zu Abbildung 6.34 zeigt sich hier für das explizite (EKmS) und implizite stochastische Klonen (IKmS) eine deutliche Verbesserung durch den Schleifenschluss. Außerdem sind die Formen bei diesen Ansätzen wieder sehr ähnlich. Bei der Methode mit dem Offset (OmS) ergibt sich auch in dieser Betrachtung keine Verbesserung durch den Schleifenschluss. Mit dem Schleifenschluss mittels GNSS-Offset sind bei diesem Ansatz zwei Offsets zu schätzen, was hier dazu führt, dass sich die Korrektur durch den Schleifenschluss zu gering auswirkt. Die inkrementellen Faktor-Graph-Methoden zeigen gegenüber Abbildung 6.34 nur leichte Verbesserungen, da diese bei der Aktualisierung auch vergangene Schätzungen verbessern und damit auch der Schleifenschluss mit verbesserten Schätzungen erfolgt, was bereits den absoluten 3D-Positionsfehler reduziert.

In Tabelle 6.6 sind die mittleren relativen 3D-Positionsfehler ohne und mit Schleifenschluss gegenübergestellt und auch hier ist außer für die Methode mit dem Offset (O) für alle Filtermethoden eine deutliche Verbesserung mit dem Schließen von Schleifen zu beobachten.

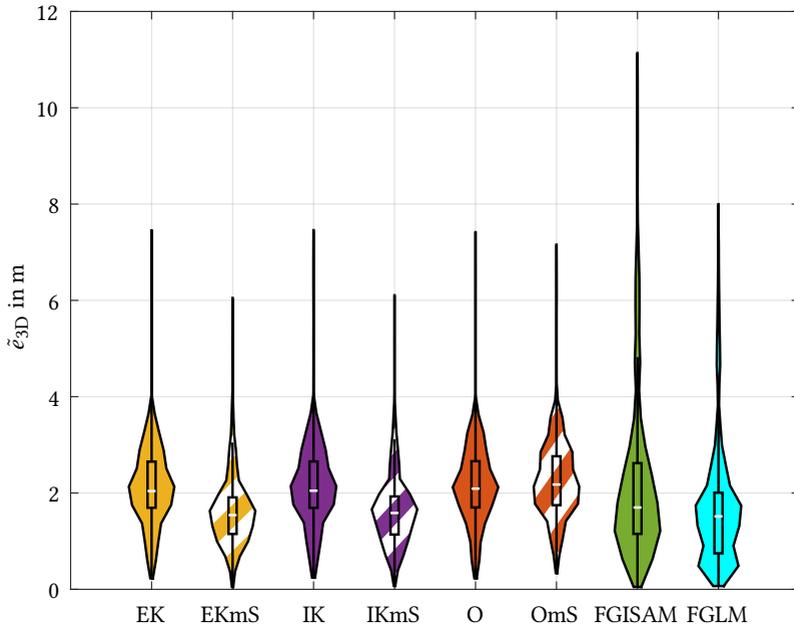


Abbildung 6.38: Violinengrafik des relativen 3D-Positionsfehlers.

In Tabelle 6.7 sind die Mediane der relativen 3D-Positionsfehler ohne und mit Schleifenschluss gegenübergestellt und hier ist ebenso außer für die Methode mit dem Offset (O) für alle Filtermethoden eine deutliche Verbesserung mit dem Schließen von Schleifen zu beobachten. Die Filtermethoden ohne Offset weisen dabei mit Schleifenschluss eine mit den Faktor-Graph-Methoden vergleichbare Präzision auf.

Qualitativer Vergleich

Eines der Entwicklungsziele der Filter-Algorithmen ist auch die simultane Kartierung, wofür der GNSS-Offset nach Abschnitt 5.3.3 eingeführt wurde. In den vorherigen Betrachtungen wurden zwar keine besseren Ergebnisse mit dem Schließen von Schleifen für die absolute Genauigkeit der Lokali-

Tabelle 6.6: Mittlerer relativer 3D-Positionsfehler in m.

	Ohne Schleifenschluss	mit Schleifenschluss (mS)
EK	2,15	1,62
PEK	2,15	1,62
IK	2,15	1,63
PIK	2,15	1,64
O	2,17	2,27
FGISAM	-	2,19
FGLM	-	1,66

Tabelle 6.7: Median des relativen 3D-Positionsfehler in m.

	Ohne Schleifenschluss	mit Schleifenschluss (mS)
EK	2,04	1,54
PEK	2,07	1,54
IK	2,05	1,59
PIK	2,07	1,60
O	2,09	2,18
FGISAM	-	1,70
FGLM	-	1,52

sierung erzielt (siehe Tabelle 6.5), jedoch waren bei den Ergebnissen der relativen Betrachtung und bei der Differenz der maximalen und minimalen Höhe der Lokalisierung geringere Werte für die Fehlermaße zu beobachten, siehe Tabelle 6.6 bzw. 6.4.

Wie in Abschnitt 6.5.2 sollen deshalb auch Karten zu einem qualitativen Vergleich herangezogen werden. Diese sind wieder NDT-Karten, welche mit den berechneten Schätzungen aufgezeichnet wurden, ohne sie jedoch für die Lokalisierung zu verwenden. Die Karten in Abbildung 6.39 zeigen den nord-westlichen Teil des Areals, wobei Abbildung 6.39a den Ausschnitt der Referenzkarte aus Abbildung 6.32 zeigt. In Abbildung 6.39b ist das Kartierungsergebnis mit der inkrementellen Faktor-Graph-Methode FGISAM

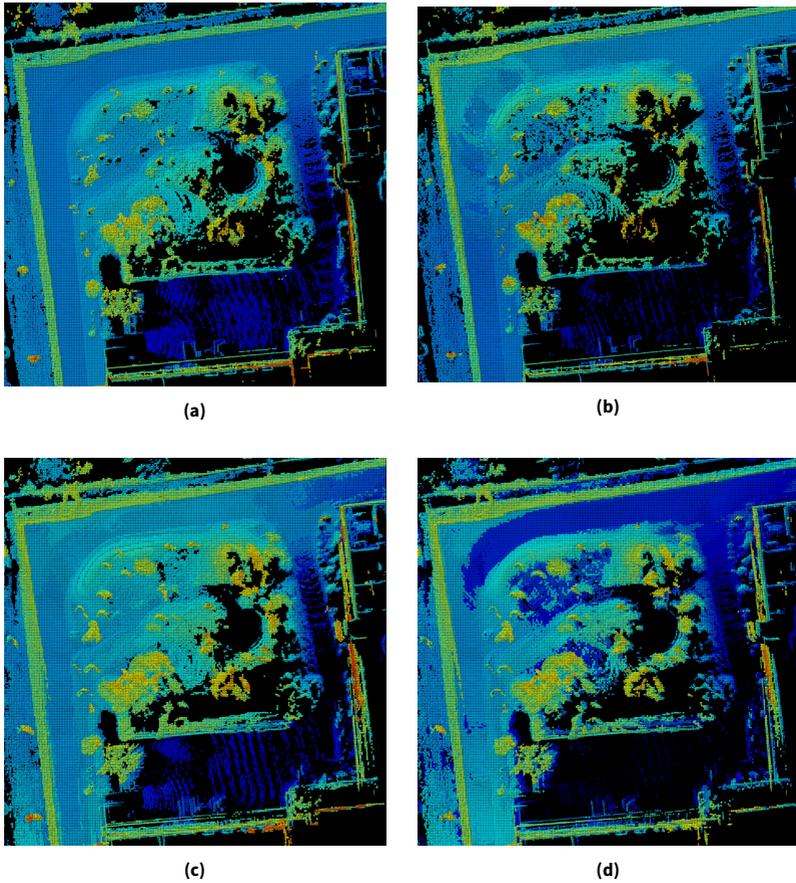


Abbildung 6.39: Ausschnitt aus Kartierungsergebnis des Faktor-Graph-SLAM-Algorithmus mit einer Optimierung über den vollen Pfad-Posterior. (a), mit FGISAM (b), EKmS (c), EK (d). Die Höhe ist farbkodiert von blau (niedrig) über grün zu rot (hoch).

dargestellt. Es zeigt gegenüber der Referenz leichte Ungenauigkeiten auf der Straße in der Kurve. Die Hecke an dem linken und dem oberen Rand und die Gebäudewände am rechten und am unteren Rand sind dagegen sehr genau kartiert worden. Die Kartierungsergebnisse des Filters mit explizitem Klonen mit dem Schließen von Schleifen (EKmS) in Abbildung 6.39c bzw. ohne das Schließen von Schleifen (EK) in Abbildung 6.39d zeigen demgegenüber einige Ungenauigkeiten an den Gebäudewänden. Ansonsten zeigt das Filter mit dem Schließen von Schleifen (EKmS) eine ähnlich gute Karte wie die inkrementelle Faktor-Graph-Methode mit leichten Ungenauigkeiten auf der Straße, wenn auch an anderer Stelle. Das Filter ohne das Schließen von Schleifen (EK) zeigt hingegen sogar Inkonsistenzen bei der Straße, da diese zweimal in verschiedenen Höhen kartiert wurde (dunkelblau und hellblau). Dies liegt an den großen Fehlern in der Höhenschätzung, welche hauptsächlich durch die fehlerbehafteten Messungen des GNSS verursacht werden. Sie konnten durch die zusätzliche Schätzung des GNSS-Offsets in Abbildung 6.39c verringert werden, sodass nur noch leichte Ungenauigkeiten in der Straße sichtbar sind.

6.5.4 Rechenzeit

Da das Filter auch für die echtzeitfähige Lokalisierung und Kartierung auf autonomen mobilen Robotersystemen mit begrenzten Ressourcen eingesetzt werden soll, sind die benötigten Rechenzeiten eine wichtige Eigenschaft.

Die mittleren Berechnungszeiten für die Integration der einzelnen Messungen für den Vergleich der verschiedenen Filter-Methoden sind in Tabelle 6.8 aufgelistet. Die Ergebnisse wurden auf einem Intel i7-7600U Mobilprozessor mit zwei Kernen und 2,8 GHz Basistakt erzielt.

Die Messungen der IMU werden sowohl für den Strapdown-Algorithmus der Prädiktion als auch die Korrektur der horizontalen Lage verwendet. Zusätzlich wird bei der Integration der IMU-Messungen auch die Korrektur durch das nicht-holonome Modell angewendet. Da hier meh-

Tabelle 6.8: Mittlere Rechenzeit in μs .

		EK	PEK	IK	PIK	O
IMU	@ 100 Hz	115,1	71,5	53,7	51,4	99,7
GNSS	@ 4 Hz	98,5	56,4	38,3	36,4	59,8
Rad-Odometrie	@ 10 Hz	37,3	21,7	11,8	10,8	25,9
LiDAR	@ 10 Hz	26,5	31,5	329,9	99,0	112,7

rere Schritte durchgeführt werden, wird mehr Rechenzeit im Vergleich zur Integration von GNSS und Rad-Odometrie benötigt. Die Integration der GNSS-Messungen beinhaltet neben den Positions- und Geschwindigkeitskorrekturen den Vorverarbeitungsschritt der Gierwinkelschätzung und dessen Korrektur. Für die Integration der Rad-Odometrie ist lediglich ein Korrekturschritt im Filter nötig, weshalb sie wesentlich weniger Rechenzeit als die Integration der GNSS-Messungen benötigt. Die Integration der relativen Messung des Scan-Matchings mit den 3D-LiDAR-Daten ist für die verschiedenen Ansätze unterschiedlich. Der Korrekturschritt des 3D-LiDARs bei der Offset-Methode (O) beinhaltet die Akkumulation von $\mathcal{F}_{k+m,k}$, die Rekonstruktion der augmentierten Systemkovarianzmatrix $\check{\mathbf{P}}_{k+m|k+m-1}$ und den eigentlichen Korrekturschritt des Kalman-Filters. Beim impliziten Klonen (IK) und partiellen impliziten Klonen (PIK) sind die gleichen Berechnungen notwendig, es kommt hier jedoch noch zu Beginn das Smoothing hinzu. Bei den beiden Ansätzen mit explizitem Klonen, sei dies komplett (EK) oder nur partiell (PEK), muss nur der Kalman-Korrekturschritt ausgeführt werden. Im Anschluss an den Korrekturschritt der 3D-LiDAR-Messungen wird wieder jeweils der alte Klon durch einen neu erzeugten ersetzt.

Die Aktualisierungen mit den Messungen der IMU, des GNSS und der Rad-Odometrie benötigen bei der Methode des expliziten Klonens (EK) am meisten Rechenzeit, da hier die Systemkovarianzmatrix mit 38×38 am größten ist. Bei den Methoden des impliziten Klonens (IK und PIK) hat sie hier lediglich die einfache Größe von 19×19 . Die Systemkovarianzma-

trizen der Offset-Methode (O) und der Methode des partiellen expliziten Klonens (PEK) haben eine Größe von 25×25 und die Rechenzeiten liegen dazwischen. Bei der Prädiktion benötigt die Methode partiellen expliziten Klonens (PEK) etwas weniger Rechenzeit als die Offset-Methode, da hier nur ein Teil der Systemkovarianzmatrix aktualisiert werden muss, vgl. Gleichung (3.22) bzw. Gleichung (5.19). Die größten Unterschiede ergeben sich bei den Aktualisierungen mit den 3D-LiDAR-Daten. Die Methode des impliziten Klonens mit Smoothing (IK) benötigt hier signifikant länger als die anderen Methoden, da das Smoothing mit ihrer Rückwärtsfilterung einen vielfachen Rechenaufwand bedeutet und $\mathcal{F}_{k+m,k}$ rekonstruiert werden muss. Das partielle implizite Klonen (PIK) benötigt demgegenüber nur geringfügig länger als die anderen Methoden, da hier nur mit einer 6×6 Systemkovarianzmatrix geglättet wird. Die Rechenzeit bei den expliziten Ansätzen ist demgegenüber sehr gering, da hier lediglich der Korrekturschritt des Kalman-Filters anfällt. Die Rechenzeit der Offset-Methode (O) liegt dazwischen, da hier noch zusätzlich die Berechnung von $\mathcal{F}_{k+m,k}$ dazu kommt.

Zur Abschätzung des Rechenaufwands in der realen Anwendung können auch die kumulierten Rechenzeiten herangezogen werden, welche in Tabelle 6.9 aufgelistet sind. Hierfür werden die mittleren Rechenzeiten aller Sensoren mit ihrer jeweiligen Rate multipliziert und addiert, was dann die benötigte Rechenzeit pro Sekunde ergibt. Die kumulierten Rechenzeiten sind alle um die 10 ms, was einer Auslastung der verfügbaren Rechenzeit von 1 % entspricht. Die Methoden des expliziten Klonens (EK) und dem Offset (O) haben mit 12,6 ms bzw. 11,6 ms etwas höhere kumulierte Rechenzeiten im Vergleich zu 7,9 ms und 8,9 ms für die Methoden mit partiellem explizitem Klonen (PEK) bzw. implizitem Klonen (IK). Die geringste kumulierte Rechenzeit weist das partielle implizite Klonen (PIK) mit 6,4 ms auf. D. h., die impliziten Methoden mit Smoothing weisen mit diesem Sensorsetup jeweils einen geringeren Rechenaufwand auf als die entsprechenden expliziten Methoden. Ein weiterer Vorteil der Methoden mit implizitem Klonen mit Smoothing (IK und PIK) ist, dass der zusätzliche Rechenauf-

wand für das stochastische Klonen entfällt, falls der relative Sensor ausfällt oder temporär nicht genutzt werden soll. In der zweiten Zeile von Tabelle 6.9 sind die kumulierten Rechenzeiten ohne die Verarbeitung des LiDARs aufgelistet und hier haben die anderen Methoden alle einen höheren Rechenaufwand als die Methoden mit implizitem Klonen (IK), da hier die augmentierten Matrizen bei den expliziten Methoden (EK und PEK) bzw. die zusätzlichen Offset-Variablen (O) trotzdem zu berechnen sind.

Tabelle 6.9: Kumulierte Rechenzeiten pro s in ms.

	EK	PEK	IK	PIK	O
mit LiDAR	12,6	7,9	8,9	6,4	11,6
ohne LiDAR	12,3	7,6	5,6	5,4	10,5

Ein weiterer Aspekt bei der Betrachtung der Rechenzeit ist die Skalierbarkeit, d. h. der Anstieg der Rechenzeit beim Hinzufügen weiterer relativer Sensoren. Hier ist bei den impliziten Methoden ein geringerer Anstieg zu erwarten, da die zusätzliche Rechenzeit für weitere Korrekturschritte der relativen Sensoren hinzuaddiert wird, wohingegen sich bei den anderen Methoden die Matrizen vergrößern. So ergibt sich durch einen weiteren relativen Sensor für das implizite Klonen (IK) eine kumulierte Rechenzeit von 12,2 ms bzw. von 7,4 ms für das partielle implizite Klonen (PIK). Damit sind beide immer noch geringer als die jeweilige explizite Methode mit nur einem Sensor (EK: 12,6 ms, PEK: 7,9 ms).

Für die Methoden des impliziten Klonens (IK) und des expliziten Klonens (EK) wurde noch eine Evaluation der Skalierung der Rechenzeit auf einem weiteren Testsystem durchgeführt. Die Filter wurden mittels sogenannten C++-Templates neu implementiert, sodass die Anzahl der zu integrierenden relativen Sensoren vor dem Kompilieren angepasst werden kann [Grö19]. Dies hat vor allem bei dem Filter mit explizitem Klonen den Vorteil, dass, falls kein relativer Sensor genutzt wird, die Berechnungen nicht mehr unnötig mit einem augmentierten Systemzustand durchgeführt werden müssen, wie es bei den Ergebnissen in Tabelle 6.9 ohne LiDAR

der Fall war. Allerdings bringt dies bei einem temporären Sensorausfall keine Vorteile hinsichtlich der Rechenzeit wie beim impliziten Klonen. Die kumulierten Rechenzeiten auf einem Intel Xeon E3-1245 mit vier Kernen und 3,3 GHz sind in Tabelle 6.10 aufgelistet. Bei diesem Vergleich wurde kein GNSS-Offset mitgeschätzt, d. h., die Größe der Systemkovarianzmatrix beträgt 16×16 . Beim expliziten Klonen (EK) wird die augmentierte Systemkovarianzmatrix mit einem relativen Sensor (32×32) bzw. mit zwei relativen Sensoren (48×48) entsprechend größer. Ohne zusätzlichen relativen Sensor, also ohne stochastisches Klonen, zeigen beide Varianten die gleichen Rechenzeiten, da sich in dieser Implementierung auch ihre Matrixengrößen und Korrekturschritte nicht unterscheiden. Bei dem impliziten Klonen (IK) steigt die kumulierte Rechenzeit mit den zusätzlichen relativen Sensoren jeweils um ca. 1,9 ms, da für jeden relativen Sensor ein Korrekturschritt hinzukommt und beide Sensoren eine Rate von 10 Hz haben. Beim expliziten Klonen (EK) ist hingegen jeweils mehr als eine Verdoppelung der kumulierten Rechenzeit zu beobachten. Bei der Steigerung von einem auf zwei Sensoren nähert sich der Faktor von 2,5 dabei der theoretisch zu erwartenden Steigerung der Laufzeit bei Matrixmultiplikationen, welche zwischen $1,5^{2,3737} = 2,6$ und $1,5^3 = 3,4$ liegt, vgl. Abschnitt 2.2.1.

Tabelle 6.10: Kumulierte Rechenzeiten pro s in ms.

		IK	EK
ohne		4,1	4,1
ein rel. Sensor	@ 10 Hz	5,9	11,6
zwei rel. Sensoren	@ 10 Hz	7,8	29,1

Die Methoden mit den inkrementellen Faktor-Graph-Algorithmen haben im Vergleich zu den Filter-Methoden signifikant höhere Rechenzeiten. So hat der FGLM-Algorithmus eine mittlere Rechenzeit von 11,7 s mit einzelnen Spitzen bis zu 47,5 s. Wie zu erwarten, benötigt der FGISAM-Algorithmus demgegenüber deutlich weniger Rechenzeit mit durchschnittlich 814,5 ms und Spitzen bis zu 21,1 s, was jedoch gegenüber den Filter-

Methoden immer noch sehr viel höher ist und einer Verwendung unter Echtzeitbedingungen entgegensteht. Vor dem erstem Schleifenschluss sind die benötigten Rechenzeiten deutlich geringer mit durchschnittlich 400,1 ms und Spitzen bis zu 2,2 s beim FGISAM-Algorithmus und durchschnittlich 1,8 s und Spitzen bis zu 6,2 s beim inkrementellen FGLM-Algorithmus. Hier wird deutlich, dass bei den Faktor-Graph-Algorithmen der Rechenaufwand mit der Zeit deutlich größer wird, wobei er wie zu erwarten bei dem FGISAM-Algorithmus weniger stark zunimmt und allgemein auch deutlich niedriger liegt als beim FGLM-Algorithmus.

Wie eingangs erwähnt, wurde das Scan-Matching bei der Evaluation für eine bessere Vergleichbarkeit vorberechnet. Es hat jedoch einen erheblichen Anteil an der Rechenzeit für die reale Anwendung. Die durchschnittliche Rechenzeit für ein Scan-Matching mittels GICP-Algorithmus liegt bei 265,8 ms, was deutlich über dem Messintervall des 3D-LiDARs von 100 ms liegt. D. h., auf dem für die Evaluation verwendeten System könnte in der realen Anwendung nicht für jedes aufeinanderfolgende Scan-Paar das Scan-Matching berechnet werden. Damit auch während der Berechnungsdauer eine Aktualisierung der Lokalisierung für andere Komponenten des autonomen mobilen Robotersystems mit hoher Rate erfolgen kann, ist eine parallele Filter-Struktur nach Abschnitt 5.3.4 notwendig. Die benötigte kumulierte Rechenzeit der Prädiktions- und Korrekturschritte der Filter ist hier in jedem Fall gering genug, damit das Filter, welches die 3D-LiDAR-Daten verarbeitet, nach deren Verarbeitung stets zum Ausgabe-Filter mit der hohen Rate aufschließen kann.

Wird eine inkrementelle Kartierung durchgeführt, sind zusätzlich noch die Rechenzeiten für die Aktualisierung der Karte zu berücksichtigen. Die in Abschnitt 5.3.2 vorgestellte inkrementelle Aktualisierung der NDT-Karte benötigt für die Integration eines Scans des 3D-LiDARs mit durchschnittlich ca. 130.000 Punkten im Mittel 42 ms mit der Aktualisierung der Existenzwahrscheinlichkeiten mit Strahlverfolgung und nur um die 7 ms, wenn die Existenzwahrscheinlichkeiten unberücksichtigt bleiben. Damit liegt die

Rechenzeit der Aktualisierung der Karte in beiden Varianten deutlich unter dem Messintervall des 3D-LiDARs.

6.5.5 Zeitstempelfilter

Zur Evaluation des in Abschnitt 5.3.4 vorgestellten Zeitstempelfilters wurden Sensordaten mit und ohne Filterung der Zeitstempel aufgezeichnet. Abbildung 6.40 zeigt beispielhaft eine Sequenz der Zeitdifferenzen zwischen den Zeitstempeln aufeinanderfolgender Messungen der Xsens-IMU. Die spezifizierte Rate der IMU beträgt 100 Hz, d. h., die Zeitdifferenz sollte 10 ms betragen. Die Zeitdifferenzen der ungefilterten Daten weisen Schwankungen bis zu einigen Millisekunden auf, während bei den gefilterten Daten in der Abbildung keine Schwankungen erkennbar sind. Die Mittelwerte liegen hingegen mit 10,0008 ms bzw. 10,0001 ms beide sehr dicht beieinander und nahe dem spezifizierten Wert.

In Tabelle 6.11 sind die Standardabweichungen der Zeitstempeldifferenzen für die in den vorherigen Abschnitten verwendeten Sensoren im Vergleich zwischen ungefilterter und gefilterter Aufzeichnung aufgelistet. Die hier angegebene Datenrate des LiDARs bezieht sich auf die Rohdatenpakete von denen mehrere zusammen einen Scan mit 10 Hz ergeben. Hier sind bei allen Sensoren Verbesserungen um mehrere Größenordnungen zu erkennen und es ergeben sich für die gefilterten Daten Standardabweichungen im einstelligen Mikrosekundenbereich.

Tabelle 6.11: Standardabweichung σ in μs .

		ungefiltert	gefiltert
IMU	@ 100 Hz	589,2	1,6
GNSS	@ 4 Hz	2121,8	7,4
Rad-Odometrie	@ 10 Hz	1942,9	5,2
LiDAR	@ 347,22 Hz	26,8	0,3

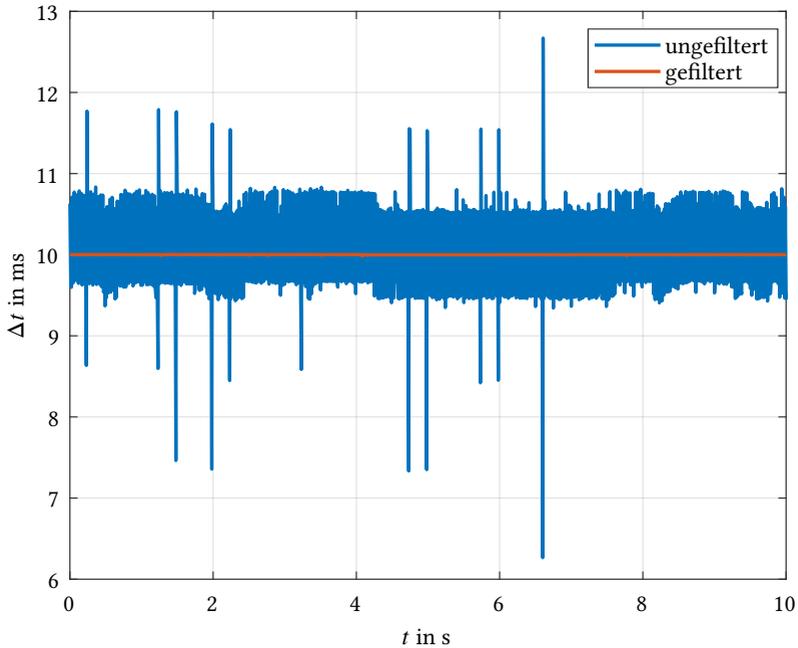


Abbildung 6.40: Zeitstempeldifferenzen der IMU, welche eine Rate von 100 Hz hat.

Bei dem in den vorherigen Abschnitten verwendeten Datensatz betrug die maximale gemessene Abweichung ohne Filterung weniger als 5 ms und mit einer maximalen Geschwindigkeit der mobilen Roboterplattform von 4 m/s ergibt sich für eine Positionsschätzung ein potentieller Fehler von unter ± 2 cm. Die maximale von der IMU gemessene Drehrate betrug unter $40^\circ/\text{s}$ was einen potentiellen Winkelfehler von weniger als $\pm 0,2^\circ$ ergibt. D. h., der Einfluss der Zeitstempelabweichung auf die einzelnen Lokalisierungsschätzungen ist sehr gering.

Die Filterung der Zeitstempel hat dennoch einen positiven Einfluss auf die Fusion für die Lokalisierung und Kartierung, da sich kleine Fehler in der Lage stärker auf die 3D-LiDAR-Messungen auswirken. So ergibt ein Winkelfehler von $0,2^\circ$ in einer Entfernung von 50 m eine Abweichung von ca.

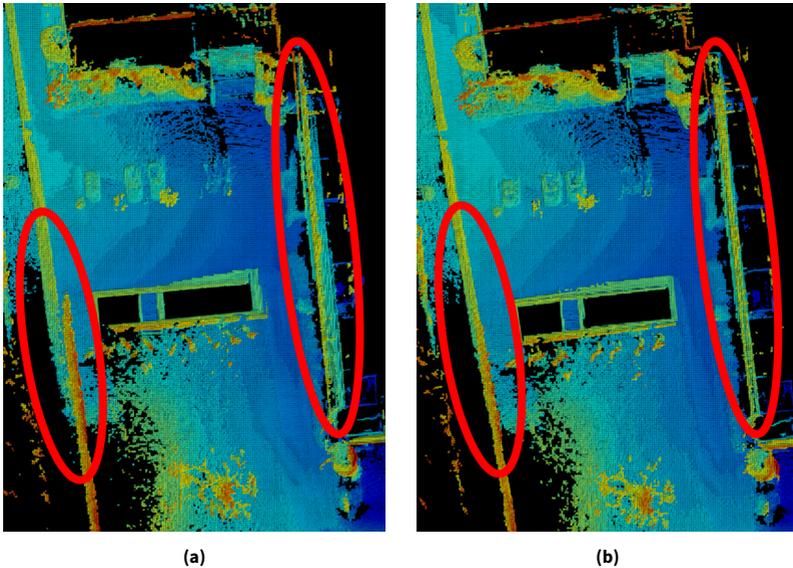


Abbildung 6.41: Kartierungsergebnis ohne Zeitstempelfilter (a) und mit Zeitstempelfilter (b). Die relevanten Bereiche sind mit roten Ellipsen gekennzeichnet. Die Höhe ist farbkodiert von blau (niedrig) über grün zu rot (hoch).

17 cm. Außerdem können sich bei SLAM auch kleine Abweichungen über die Zeit stärker auswirken. Wie in Abbildung 6.41 in einigen Bereichen, welche durch rote Ellipsen gekennzeichnet sind, zu erkennen ist, kann die Genauigkeit der Kartierung durch die Zeitstempelfilterung erhöht werden. Auf der linken Seite befindet sich ein lange gerade Hecke, welche in der Karte ohne die Zeitstempelfilterung eine Unterbrechung aufweist, wohingegen in der Karte mit Zeitstempelfilterung keine Unterbrechung der Hecke erkennbar ist. Auf der rechten Seite befindet sich ein Gebäude, welches in der Karte mit der Zeitstempelfilterung deutlich präziser kartiert wurde, erkennbar an der schmalere Kontur.

6.5.6 Alternativer Datensatz

Als alternativer Datensatz zur Verifikation der Vergleiche zwischen den Filter-Methoden mit stochastischem Klonen wurden Datensätze der KITTI Vision Benchmark Suite verwendet [Gei12; Gei13]. Die Datensätze sind unter <http://www.cvlibs.net/datasets/kitti/> erhältlich. Die Datensätze sind im Rahmen der Forschung für autonomes Fahren im Straßenverkehr aufgenommen worden und beinhalten somit urbane Szenarien und keine unstrukturierten Umgebungen.

Die Ausgangsdaten enthalten Messungen eines INS/GNSS von OxTS, welches eine Lokalisierung mit hoher Rate von 100 Hz ausgibt. Hiervon wurden jedoch lediglich die Rohdaten verwendet, welche auch aus einem normalen GNSS und einer IMU kommen würden. Die zur Evaluation verwendeten GNSS-Daten umfassten den Längengrad, den Breitengrad, die Höhe und die 3D-Geschwindigkeit mit 10 Hz. Die IMU-Daten umfassten die 3D-Drehraten und die 3D-Beschleunigungen mit 100 Hz. Der verwendete Datensatz wies auch einige zeitweilige Ausfälle der Sensordaten auf. Da die Prädiktion im Filter mit den IMU-Daten durchgeführt wird, wurden die IMU-Daten für die Zeiten der Ausfälle linear interpoliert, die GNSS-Daten hingegen wurden mit Lücken verwendet. Als Umgebungssensor kommt ebenfalls ein Velodyne HDL-64 zum Einsatz und die Datensätze enthalten bereits inertial korrigierte 3D-Scans.

Es wurde zunächst der Datensatz „Sequenz 00“ ausgewählt, da für diesen eine Referenzlokalisierung erhältlich ist und er eine lange Strecke von über 3,7 km abdeckt. Die 3D-Positionsfehler der Methoden mit stochastischem Klonen bezogen auf die Referenzlokalisierung sind in Tabelle 6.12 aufgeführt.

Die mittleren Fehler \bar{e}_{3D} sind mit 14 cm alle sehr gering und unterscheiden sich nur um ein bis zwei Millimeter. Bei den maximalen Fehlern hat das Filter mit dem Offset einen um lediglich 3 cm höheren Fehler. Dass alle Ansätze so nahe beisammen liegen, hängt mit dem GNSS zusammen, welches sehr geringe Unsicherheiten aufweist, und somit der Einfluss der anderen Messungen auf die Lokalisierungsschätzung geringer ausfällt. Aus

Tabelle 6.12: 3D-Positionsfehler in m.

	\bar{e}_{3D}	$\max(e_{3D})$
EK	0,14	1,90
PEK	0,14	1,90
IK	0,14	1,90
PIK	0,14	1,90
O	0,14	1,93

diesem Grund sind auch die Einflüsse der Integrationsmethode der relativen Messungen geringer. Abbildung 6.42 zeigt das Kartierungsergebnis mit der Filter-Methode des impliziten Klonens mit Smoothing, welches mit einer Straßenkarte hinterlegt wurde.

Da sich mit GNSS kaum Unterschiede zwischen den Methoden ergaben, wurde auch hier ein Ausfall des GNSS simuliert und die 3D-Positionsfehler gegenüber der Referenzlokalisierung evaluiert, siehe Abbildung 6.43. Der Ausfall beginnt nach 8,5 s, damit die Filter Zeit hatten einzuschwingen, und bleibt für die restliche Zeit bestehen, d. h., es stehen ab diesem Zeitpunkt nur noch die IMU-Daten und das Scan-Matching der 3D-LiDAR-Daten als relative Posenmessung für die Lokalisierungsschätzung zur Verfügung. In der Abbildung ist der Beginn des Ausfalls mit einer vertikalen gestrichelten Linie gekennzeichnet. Bei allen Methoden ist der Fehler mit der Zeit tendenziell steigend. Das partielle explizite Klonen (PEK) ist wieder nicht in die Abbildung aufgenommen, da es weniger als 7 cm Unterschied zum kompletten expliziten Klonen aufwies. Der Unterschied zwischen dem impliziten Klonen mit Smoothing (IK) und dessen partieller Methode (PIK) fällt hier mit zunehmender Zeit größer aus und auch der Unterschied zu der expliziten Methode (EK) wird im späteren Verlauf deutlicher als bei dem kurzen Ausfall in Abschnitt 6.5.2. Wie schon bei den Ergebnissen zuvor, unterscheidet sich der Verlauf bei der Offset-Methode (O) schon zu Beginn deutlichen von den anderen Methoden.



Abbildung 6.42: Ergebnis der Kartierung mit der Filter-Methode des impliziten Klonens mit Smoothing über einer Straßenkarte. Die Höhe ist farbkodiert von blau (niedrig) über grün zu rot (hoch). Die Straßenkarte im Hintergrund ist von OpenStreetMap: ©OpenStreetMap-Mitwirkende (<https://www.openstreetmap.org/copyright>).

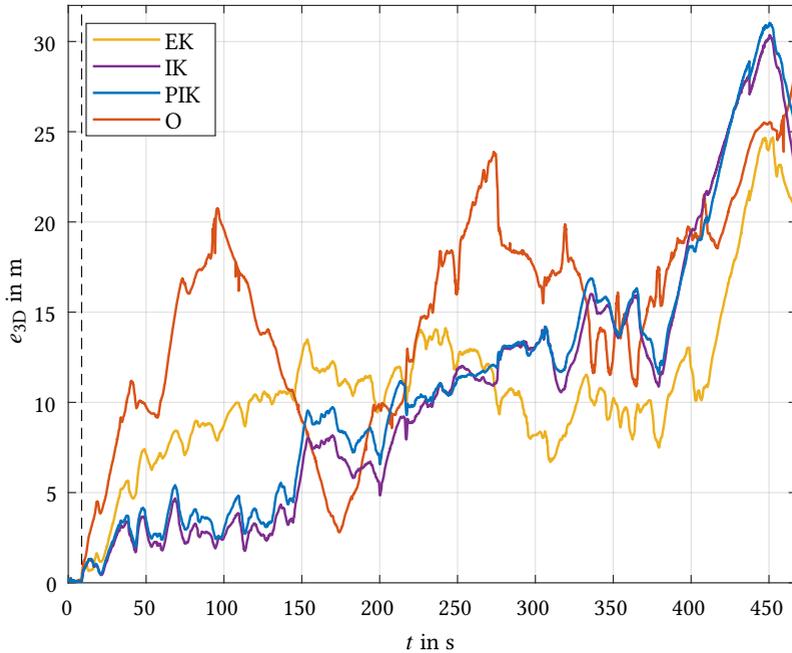


Abbildung 6.43: 3D-Positionsfehler mit GNSS-Ausfall. Der Beginn des Ausfalls ist mit einer vertikalen gestrichelten Linie gekennzeichnet.

Stehen nur relative Positionsmessungen für lange Zeiträume zur Verfügung, bietet sich zur Evaluation die Betrachtung eines relativen Fehlers bezogen auf die zurückgelegte Wegstrecke an. Dieser relative Fehler wird wie folgt definiert:

$$q_{3D,k} = e_{3D,k} / s_k, \quad (6.7)$$

wobei s_k die bis dahin zurückgelegte Strecke ist.

Abbildung 6.44 zeigt den zeitlichen Verlauf des relativen 3D-Positionsfehlers. Dieser ist zu Beginn für alle Methoden sehr hoch, da hier die zurückgelegte Wegstrecke sehr klein ist und sich damit kleine Abweichungen sehr stark auswirken. Mit der Zeit fallen die relativen 3D-Positionsfeh-

ler aller Methoden auf unter 1 % ab und weisen nur noch geringe Schwankungen auf.

Tabelle 6.13: Mittlerer relativer 3D-Positionsfehler in %.

	\bar{q}_{3D}	\bar{q}_{3D} ab 150 s
EK	1,09	0,59
PEK	1,09	0,59
IK	0,77	0,62
PIK	0,78	0,64
O	1,69	0,72

In Tabelle 6.13 sind die mittleren relativen 3D-Positionsfehler für die verschiedenen Methoden aufgelistet. Wobei dieser jeweils einmal für die gesamte Dauer und einmal für den Zeitraum ab 150 s ermittelt wurde. Letzteres wurde mit aufgenommen, da aus Abbildung 6.44 ersichtlich ist, dass die relativen 3D-Positionsfehler der verschiedenen Methoden im späteren Verlauf recht nahe beieinanderliegen und die anfänglichen großen Abweichungen keine Rolle mehr spielen. Die Methoden des impliziten Klonens mit Smoothing (IK und PIK) zeigen hier über den gesamten Zeitraum die geringsten relativen 3D-Positionsfehler im zweiten Zeitraum sind jedoch die Methoden des expliziten Klonens (EK und PEK) besser. Die Offset-Methode (O) zeigt vor allem in dem gesamten Zeitraum einen viel höheren mittleren relativen 3D-Positionsfehler, da hier der relative 3D-Positionsfehler am Anfang sehr viel länger als die anderen Methoden braucht, um unter 1 % zu fallen. In dem späteren Zeitraum liegt der mittlere relative 3D-Positionsfehler der Methode mit dem Offset näher an den anderen Methoden, welche sehr ähnliche Ergebnisse liefern.

Zur Verifikation wurden noch zwei weitere Datensätze, „Sequenz 02“ und „Sequenz 05“, hinzugezogen. Zusammen sind mit den drei Datensätzen insgesamt knapp über 10,9 km Fahrstrecke abgedeckt. In Tabelle 6.14 sind die zusammengefassten mittleren 3D-Positionsfehler über alle drei Datensätze aufgelistet und es zeigen sich die gleichen Verhältnisse wie

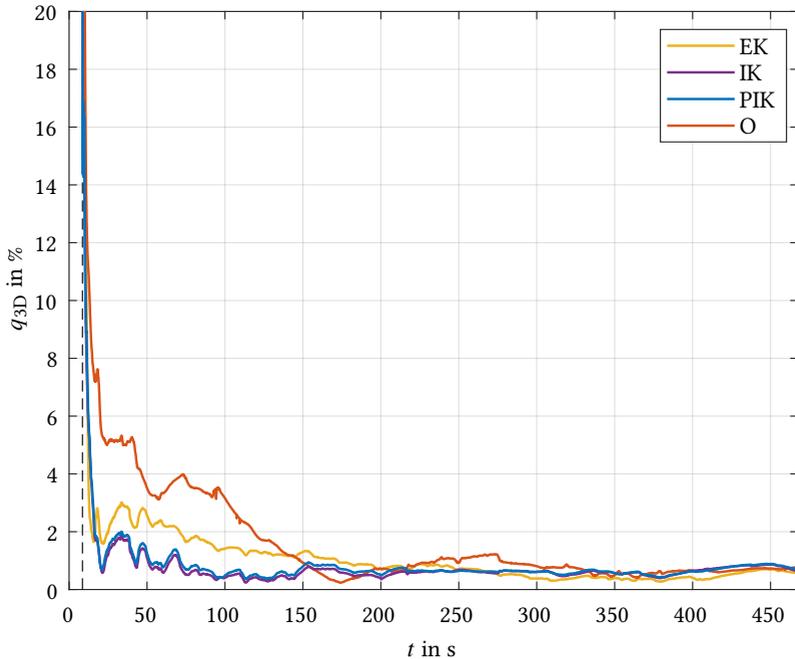


Abbildung 6.44: Relativer 3D-Positionsfehler mit GNSS-Ausfall. Der Beginn des Ausfalls ist mit einer vertikalen gestrichelten Linie gekennzeichnet.

bei der Analyse nur des Datensatzes „Sequenz 00“. So sind die impliziten Methoden (IK und PIK) über die gesamte Zeit leicht besser als die expliziten Methoden (EK und PEK) und umgekehrt für den Zeitraum ab 150 s. Die Methode mit dem Offset (O) weist wieder jeweils einen höheren mittleren relativen 3D-Positionsfehler auf.

Dynamische Objekte

Der Datensatz „Sequenz 00“ enthält auch dynamische Objekte und kann daher für die Analyse der Eignung der Existenzwahrscheinlichkeit zu deren Behandlung bei der Kartierung nach Abschnitt 5.3.2 herangezogen werden.

Tabelle 6.14: Mittlerer relativer 3D-Positionsfehler in %.

	\bar{q}_{3D}	\bar{q}_{3D} ab 150 s
EK	1,47	0,99
PEK	1,47	0,99
IK	1,39	1,10
PIK	1,41	1,12
O	1,82	1,30

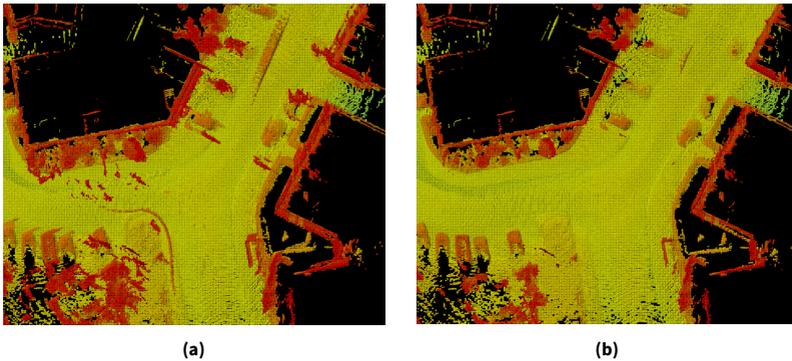
**Abbildung 6.45:** Kartierung in einer Situation mit dynamischen Objekten. Einfache Kartierung (a), mit Existenzwahrscheinlichkeiten (b). Die Höhe ist farbkodiert von blau (niedrig) über grün zu rot (hoch).

Abbildung 6.45 zeigt einen Vergleich zwischen der einfachen Kartierung und der Kartierung mit der Einbeziehung der Existenzwahrscheinlichkeit. In Abbildung 6.45a mit der einfachen Kartierung sind deutliche, langgezogene Artefakte von vorbeifahrenden Objekten zu erkennen. In Abbildung 6.45b sind diese nicht mehr vorhanden, d. h., sie konnten erfolgreich durch die Schätzung einer Existenzwahrscheinlichkeit pro Voxel eliminiert werden.

6.5.7 Zusammenfassung

Bei der Evaluation mit dem simulierten GNSS-Ausfall haben die Methoden des impliziten Klonens mit Smoothing und mit dem expliziten Klonen nahezu gleiche Ergebnisse erzielt, welche deutlich besser sind als bei der Methode mit zusätzlichem Offset. Mit dem Zurückpropagieren der Information durch das Smoothing bei dem impliziten Klonen kann in etwa die gleiche Genauigkeit wie beim expliziten Klonen erreicht werden. Während bei dem partiellen expliziten Klonen nahezu kein Unterschied zum kompletten expliziten Klonen zu beobachten war, sind die Unterschiede zwischen dem partiellen und kompletten impliziten Klonen mit Smoothing deutlicher, da die Rekonstruktion der augmentierten Systemkovarianzmatrix beim partiellen impliziten Klonen eine zusätzliche Approximation erfordert.

Beim Vergleich zu einer Referenzlokalisierung von einem Faktor-Graph-Algorithmus haben alle Filter ähnliche Ergebnisse gezeigt, bis auf die Methode mit dem zusätzlichen Offset, welche beim Gierwinkel größere Abweichungen aufwies als die anderen Methoden. Die Einbeziehung eines GNSS-Offsets für das Schließen von Schleifen hat Vorteile gezeigt, wenn der relative 3D-Positionsfehler betrachtet wird und es konnten damit qualitativ bessere Karten erzeugt werden. Die inkrementellen Faktor-Graph-Methoden haben gegenüber den Filter-Methoden in der Genauigkeit nur leichte Vorteile gezeigt, was auch an einer leicht besseren Karte zu erkennen war. Es konnte jedoch mit keiner der schritthaltenden Methoden die Genauigkeit der Referenzlokalisierung erreicht werden, weshalb für eine möglichst genaue Kartenerstellung eine abschließende Optimierung über den gesamten Pfad vorzuziehen ist. Dies ist jedoch nur möglich, wenn die Karte erst anschließend benötigt wird.

Bei den Betrachtungen zum Rechenaufwand benötigten die Filter-Methoden im Vergleich zu den inkrementellen Faktor-Graph-Methoden sehr viel weniger Rechenzeit. Alle Filter-Methoden sind mit einer Auslastung der verfügbaren Rechenzeit von ca. 1 % sehr gut für reale Anwendungen und in Kombination mit einer Parallelverarbeitung für eine konstant hohe Ausgaberate geeignet. Dabei benötigen die Filter-Methoden des

impliziten Klonens mit dem Smoothing jeweils weniger Rechenzeit als die expliziten Methoden.

Da das partielle explizite Klonen in der Genauigkeit kaum einen Unterschied zum kompletten expliziten Klonen aufweist, ist es diesem aufgrund des geringeren Rechenaufwands vorzuziehen. Sollen bestehende Filter um die Funktionalität des stochastischen Klonens erweitert werden, empfiehlt sich das implizite Klonen mit Smoothing, da die bestehenden Filterelemente, wie Zustandsvektor und Systemkovarianzmatrix, nicht angepasst werden müssen. Außerdem weist es gegenüber dem expliziten Klonen einen geringeren Rechenaufwand auf. Darüber hinaus fallen die Berechnungen für das stochastische Klonen nur an, wenn auch Messungen der relativen Sensoren verarbeitet werden. D. h., auch bei sich ändernden Sensorkonfigurationen kann das implizite Klonen mit Smoothing flexibler eingesetzt werden. Sollen weitere relative Sensoren integriert werden, weisen die impliziten Methoden eine bessere Skalierbarkeit hinsichtlich des Rechenaufwands auf.

Mit der Anwendung der Zeitstempelfilterung wurde eine deutliche Verringerung der Schwankungen der Zeitstempeldifferenzen aufeinanderfolgender Messdaten erzielt und bei der resultierenden Kartierung konnte eine verbesserte Genauigkeit beobachtet werden.

Zur Bestätigung der Resultate konnten bei der Evaluation anhand von alternativen Datensätzen mit den Filter-Methoden des impliziten Klonens wieder vergleichbare Ergebnisse zu den Methoden des expliziten Klonens erzielt werden.

Kapitel 7

Zusammenfassung und Ausblick

Nach einer Zusammenfassung der vorgestellten Arbeiten soll ein Ausblick mögliche folgende und darauf aufbauende Weiterentwicklungen beleuchten.

7.1 Zusammenfassung

Das Ziel der vorliegenden Arbeit war es, Methoden für die Kombination und Integration mehrerer Sensoren für die Lokalisierung und Kartenerstellung in heterogenen Umgebungen zu entwickeln.

Es wurde ein erweitertes Landmarkenmodell mit zusätzlichen ortsunabhängigen und visuellen Merkmalen für die robustere Wiedererkennung entwickelt und in einen RBPF-SLAM-Algorithmus integriert. Mit der verbesserten Datenzuordnung konnte gezeigt werden, dass weniger Partikel für das erfolgreiche Schließen der Schleifen benötigt werden und somit das Erstellen einer konsistenten Karte vereinfacht wird. Darüber hinaus

konnte gezeigt werden, dass mit dem erweiterten Landmarkenmodell auch die Präzision der Karte erhöht wird.

Durch die konsequente Auslegung der Erweiterungen des RBPF-SLAM-Algorithmus in probabilistisch konsistenter Weise war es möglich, die Kombination einer Landmarkenkarte mit einer 2D-Rasterkarte für die Kartierung von Mischgebieten in einem hybriden Ansatz zu realisieren. Hierdurch konnten genauere Karten im Vergleich zu den Einzelansätzen erzeugt werden.

Für die Kombination von SLAM mit Multi-Sensor-Fusion wurde ebenfalls die modulare Erweiterbarkeit des RBPF-SLAM-Algorithmus genutzt, um einen zusätzlichen absoluten Positionssensor für die Begrenzung der globalen Unsicherheit zu integrieren. Durch die Fusion mit dem absoluten Sensor konnte die Robustheit hinsichtlich des Schließens großer Schleifen effektiv gesteigert werden. So konnte gezeigt werden, dass bei großen Schleifen weniger Partikel zum Erstellen genauer Karten notwendig sind bzw. bei gleicher Partikelanzahl die Zielverteilung besser approximiert wird und damit auch die benötigte Rechenzeit verringert wird.

Für die Integration eines zusätzlichen relativen Sensors in ein Lokalisierungsfilter mittels stochastischem Klonen wurde eine neuartige Methode mit implizitem Klonen in Kombination mit Smoothing vorgestellt und dem expliziten Klonen gegenübergestellt. Die Methoden des stochastischen Klonens wurden miteinander verglichen und es konnten bei der Evaluation mit einem simulierten GNSS-Ausfall vergleichbare Ergebnisse erzielt werden. Beim Vergleich der Methoden zu einer Referenzlokalisierung von einem Faktor-Graph-Algorithmus konnten sie ebenfalls ähnliche Ergebnisse erzielen. Mit dem für das Schließen von Schleifen eingeführten GNSS-Offset konnten ein geringerer relativer 3D-Positionsfehler erzielt und qualitativ bessere Karten erzeugt werden. Die ebenfalls zum Vergleich herangezogenen inkrementellen Faktor-Graph-Methoden zeigten gegenüber den Filter-Methoden in der Genauigkeit nur leichte Vorteile. Bei der Evaluation der Rechenzeit hingegen zeigten sich die Filter-Methoden im Vergleich zu den inkrementellen Faktor-Graph-Methoden weit überlegen

und für reale Echtzeit-Anwendungen sehr gut geeignet. In Verbindung mit der Parallelverarbeitung können mit den Filtermethoden trotz der Integration des rechenaufwändigen Scan-Matchings konstant hohe Ausgaberraten erreicht werden.

Mit der Anwendung der Zeitstempelfilterung konnte eine deutliche Verringerung der Schwankungen der Zeitstempeldifferenzen aufeinanderfolgender Messdaten erzielt werden und auch bei der resultierenden Kartierung konnte eine verbesserte Genauigkeit beobachtet werden.

Ein weiterer Aspekt der Arbeit war eine möglichst praxisnahe Umsetzung der entwickelten Methoden und deren Einsatz in realen Anwendungsgebieten. So konnten mit der Implementierung der vorgestellten Algorithmen sowie der Umsetzung der Zeitstempelfilterung und der parallelen Verarbeitung umfassende Verbesserungen bei der Lokalisierung und Kartierung für mobile Robotersysteme geschaffen werden. Diese reichen von der Akquise der Sensordaten über die Fusion mehrerer relativer und absoluter Sensoren bis hin zur Integration in SLAM-Algorithmen. Ein Großteil der vorgestellten Algorithmen ist auf verschiedenen Forschungsplattformen mit unterschiedlichen Sensorausstattungen erfolgreich im Einsatz. Dazu gehören verschiedene Landfahrzeuge und schwere Arbeitsmaschinen sowie Wasserfahrzeuge [Hei19; Emt17; Kle20].

7.2 Ausblick

Das Scan-Matching von Punktwolken mit einer großen Anzahl von Punkten ist sehr rechenaufwändig und konnte auf dem zur Evaluation verwendeten System nicht mit der vollen Rate des 3D-LiDARs ausgeführt werden. Hier besteht das Potential für Verbesserungen in Form alternativer Algorithmen, welche das Scan-Matching effizienter ausführen können, oder die Portierung des GICP-Algorithmus auf massiv parallele Plattformen wie beispielsweise Grafikprozessoren (englisch: *general-purpose computation on graphics processing units* – GPGPU).

Als Alternative zu der Verwendung von einem 3D-LiDAR mit Scan-Matching könnte auch eine kamera-basierte visuelle Odometrie eingesetzt werden, wobei bei monokularer visueller Odometrie zusätzlich die Skalierung geschätzt werden muss, da diese mit nur einer Kamera nicht beobachtbar ist.

Da die Genauigkeit der Höhenmessung des GNSS im Vergleich zu den horizontalen Messungen sehr viel schlechter ist, wäre die Fusion mit einem weiteren, die Höhe messenden Sensor, wie zum Beispiel einem barometrischen Drucksensor, eine zu untersuchende Option [Zal14].

Die inkrementellen Faktor-Graph-Methoden haben gegenüber den Filtern mit stochastischem Klonen für die Lokalisierung nur eine kleine Verbesserung bei der Genauigkeit gezeigt und dabei wesentlich mehr Rechenzeit benötigt. Sie haben jedoch den Vorteil, dass bei der Optimierung auch alle zurückliegenden Posen verbessert werden, was i. A. im späteren Verlauf in einer präziseren Karte resultiert. Daher könnte ein hybrides Verfahren mit einem SC-EKF für die Echtzeit-Lokalisierung und lokale Kartierung in Kombination mit einem inkrementellen Faktor-Graph-Algorithmus für die Kartierung einer globalen Karte die Vorteile beider Methoden vereinen.

Literaturverzeichnis

- [Agh13] F. Aghili und A. Salerno. „Driftless 3-D Attitude Determination and Positioning of Mobile Robots By Integration of IMU With Two RTK GPSs“. In: *IEEE/ASME Transactions on Mechatronics* 18.1 (2013).
- [Agr06] M. Agrawal und K. Konolige. „Real-time Localization in Outdoor Environments using Stereo Vision and Inexpensive GPS“. In: *18th International Conference on Pattern Recognition (ICPR'06)*. Bd. 3. 2006.
- [And79] B. D. O. Anderson und J. B. Moore. *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.
- [Aul08] J. Aulinas, Y. Petillot, J. Salvi und X. Lladó. „The SLAM Problem: A Survey“. In: *Proceedings of the 2008 Conference on Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2008.
- [Bad19] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. F. R. Jesus, R. F. Berriel, T. M. Paixão, F. W. Mutz, T. Oliveira-Santos und A. F. de Souza. „Self-Driving Cars: A Survey“. In: *Computing Research Repository (CoRR)* abs/1901.04407 (2019).

- [Bai02] T. Bailey. „Mobile Robot Localisation and Mapping in Extensive Outdoor Environments“. Diss. University of Sydney, 2002.
- [Bha09] A. Bhattacharya, P. Kar und M. Pal. „On Low Distortion Embeddings of Statistical Distance Measures into Low Dimensional Spaces“. In: *CoRR* abs/0909.3169 (2009). arXiv: 0909.3169.
- [Ble08] G. Bleser und D. Stricker. „Using the Marginalised Particle Filter for Real-time Visual-inertial Sensor Fusion“. In: *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. 2008.
- [Bos03] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten und S. Teller. „An Atlas Framework for Scalable Mapping“. In: *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*. Bd. 2. 2003.
- [Bos04] M. Bosse, P. Newman, J. Leonard und S. Teller. „SLAM in Large-scale Cyclic Environments Using the Atlas Framework“. In: *International Journal of Robotics Research* (2004).
- [Bra01] M. Braasch. „Performance Comparison of Multipath Mitigating Receiver Architectures“. In: *Aerospace Conference, 2001, IEEE Proceedings*. Bd. 3. 2001.
- [Bra07] D. M. Bradley, R. Unnikrishnan und J. Bagnell. „Vegetation Detection for Driving in Complex Environments“. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007.
- [Bra17] M. S. Braasch. „Multipath“. In: *Springer Handbook of Global Navigation Satellite Systems*. Hrsg. von P. J. Teunissen und O. Montenbruck. Cham: Springer International Publishing, 2017.

- [Bre65] J. E. Bresenham. „Algorithm for Computer Control of a Digital Plotter“. In: *IBM Systems Journal* 4.1 (1965).
- [Bur09] A. Burns und A. Wellings. *Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX*. 4th. USA: Addison-Wesley Educational Publishers Inc, 2009.
- [Car14] L. Carlone, Z. Kira, C. Beall, V. Indelman und F. Dellaert. „Eliminating Conditionally Independent Sets in Factor Graphs: A Unifying Perspective Based on Smart Factors“. In: *Int. Conf. on Robotics and Automation (ICRA)*. 2014.
- [Che03] Z. Chen. „Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond“. In: *Statistics* 182.1 (2003).
- [Chi11] A. Chilian, H. Hirschmüller und M. Görner. „Multisensor Data Fusion for Robust Pose Estimation of a Six-legged Walking Robot“. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Sep. 2011.
- [Com00] D. Comaniciu, V. Ramesh und P. Meer. „Real-time Tracking of Non-rigid Objects Using Mean Shift“. In: *Proceedings of the 2000 IEEE Conference Computer Vision and Pattern Recognition*. Bd. 2. 2000.
- [Cov91] T. Cover und J. Thomas. *Elements of Information Theory*. New York: Wiley, 1991.
- [Cox91] I. J. Cox. „Blanche—An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle“. In: *IEEE Transactions on Robotics and Automation* 7.2 (1991).
- [CS] *Cyril Stachniss – Robotics Datasets*. URL: <http://www2.informatik.uni-freiburg.de/~stachnis/datasets.html> (besucht am 15. Mai 2019).

- [Cub17] G. D. Cubber, D. Doroftei, K. Rudin, K. Berns, A. Matos, D. Serrano, J. Sanchez, S. Govindaraj, J. Bedkowski, R. Roda, E. Silva und S. Ourevitch. „Introduction to the Use of Robotic Tools for Search and Rescue“. In: *Search and Rescue Robotics*. Rijeka: IntechOpen, 2017. Kap. 1.
- [Cur15] F. Curatella, P. Vinetti, G. Rizzo, T. Vladimirova, L. D. Vendic-tis, T. Emter, J. Peterreit, C. W. Frey, D. Usher, I. Stanciugelu, J. Schaefer, E. den Breejen, L. Gisslén und D. Letalick. „Toward a Multifaceted Platform for Humanitarian Demining“. In: *13th IARP Workshop on Humanitarian Demining and Risky Intervention*. 12th International Symposium „MINE ACTION 2015“. 2015.
- [DAL15] L. D’Alfonso, W. Lucia, P. Muraca und P. Pugliese. „Mobile Robot Localization via EKF and UKF: A Comparison Based on Real Data“. In: *Robotics and Autonomous Systems* 74 (2015).
- [Del12] F. Dellaert. *Factor Graphs and GTSAM: A Hands-on Introduction*. Techn. Ber. Center for Robotics und Intelligent Machines, Georgia Institute of Technology, 2012. URL: <http://borg.cc.gatech.edu/sites/edu.borg/files/downloads/gtsam.pdf>.
- [Del99] F. Dellaert, D. Fox, W. Burgard und S. Thrun. „Monte Carlo Localization for Mobile Robots“. In: *Proceedings of the 1999 IEEE International Conference on Robotics and Automation (ICRA)*. Bd. 2. 1999.
- [Dis00] G. Dissanayake, H. Durrant-Whyte und T. Bailey. „A Computationally Efficient Solution to the Simultaneous Localisation and Map Building (SLAM) Problem“. In: *Robotics and Automation, 2000. Proceedings. ICRA ’00. IEEE International Conference on*. Bd. 2. 2000.

- [Dis01] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte und M. Csorba. „A Solution to the Simultaneous Localization and Map Building (SLAM) Problem“. In: *IEEE Transactions on Robotics and Automation* 17 (3 2001).
- [Dou00a] A. Doucet, N. de Freitas, K. Murphy und S. Russel. „Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks“. In: *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence* (2000).
- [Dou00b] A. Doucet, S. Godsill und C. Andrieu. „On Sequential Monte Carlo Sampling Methods for Bayesian Filtering“. In: *Statistics and Computing* 10 (3 2000).
- [Dur06a] H. Durrant-Whyte und T. Bailey. „Simultaneous Localization and Mapping: Part I“. In: *IEEE Robotics & Automation Magazine* 13 (2 2006).
- [Dur06b] H. Durrant-Whyte und T. Bailey. „Simultaneous Localization and Mapping: Part II“. In: *IEEE Robotics & Automation Magazine* 13 (3 2006).
- [Dur88] H. F. Durrant-Whyte. „Sensor Models and Multisensor Integration“. In: *The International Journal of Robotics Research* 7.6 (1988).
- [Eigen] *Eigen*. URL: <http://eigen.tuxfamily.org/> (besucht am 4. Jan. 2020).
- [Elf87] A. Elfes. „Sonar-based Real-world Mapping and Navigation“. In: *IEEE Journal on Robotics and Automation* 3.3 (1987).
- [Eli03] A. Eliazar und R. Parr. „DP-SLAM: Fast, Robust Simultaneous Localization and Mapping without Predetermined Landmarks“. In: *Proceedings of the 2003 International Joint Conference on Artificial Intelligence (IJCAI)*. Bd. 3. 2003.

- [Eli06] P. Elinas, R. Sim und J. Little. „ σ SLAM: Stereo Vision SLAM Using the Rao-Blackwellised Particle Filter and a Novel Mixture Proposal Distribution“. In: *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*. 2006.
- [Emt07] T. Emter, E. Monari, C. Frey, T. Müller, H.-B. Kuntze, A. Laubenheimer und M. Müller. „AMROS – an Autonomous Mobile Robotic System for Multisensor Surveillance of Real Estates“. In: *Proceedings of the Future Security: 2nd Security Research Conference*. Hrsg. von J. Beyerer. Fraunhofer Verbund für Verteidigungs- und Sicherheitsforschung, 2007.
- [Emt08] T. Emter, C. Frey und H.-B. Kuntze. „Multisensorielle Überwachung von Liegenschaften durch mobile Roboter – Multi-Sensor Surveillance of Real Estates Based on Mobile Robots“. In: *Robotik 2008: Leistungsstand – Anwendungen – Visionen – Trends*. VDI/VDE-GMA, 2008.
- [Emt10a] T. Emter. *Probabilistic Localization and Mapping for Mobile Robots*. Techn. Ber. Fraunhofer IOSB und Institute for Anthropomatics, Vision and Fusion Laboratory, 2010.
- [Emt10b] T. Emter, A. Saltoğlu und J. Petereit. „Multi-Sensor Fusion for Localization of a Mobile Robot in Outdoor Environments.“ In: *Proc. VDE-Verlag: ISR/Robotik 2010: Visions are Reality*. (2010).
- [Emt10c] T. Emter und T. Ulrich. „Visuelle Information zur robusten Zuordnung von Landmarken für die Navigation mobiler Roboter“. In: *Forum Bildverarbeitung*. Hrsg. von F. Puente León und M. Heizmann. 2010.
- [Emt11] T. Emter. *Probabilistic Localization and Mapping for Mobile Robots*. Techn. Ber. Fraunhofer IOSB und Institute for Anthropomatics, Vision and Fusion Laboratory, 2011.

- [Emt12a] T. Emter. *Integrated Multi-Sensor Fusion and SLAM for Mobile Robots*. Techn. Ber. Fraunhofer IOSB und Institute for Anthropomatics, Vision and Fusion Laboratory, 2012.
- [Emt12b] T. Emter und T. Ulrich. „Fusion of Geometrical and Visual Information for Localization and Mapping in Outdoor Environments“. In: *Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS)*. 2012.
- [Emt14] T. Emter und J. Petereit. „Integrated Multi-Sensor Fusion for Mapping and Localization in Outdoor Environments for Mobile Robots“. In: *Proceedings SPIE 9121, Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2014*. 2014.
- [Emt17] T. Emter, C. Frese, A. Zube und J. Petereit. „Algorithmen-Toolbox für autonome mobile Robotersysteme“. In: *ATZoff-highway (2017)* 10 (2017).
- [Emt18a] T. Emter und J. Petereit. „3D SLAM With Scan Matching and Factor Graph Optimization“. In: *ISR 2018; 50th International Symposium on Robotics*. 2018.
- [Emt18b] T. Emter und J. Petereit. „Stochastic Cloning and Smoothing for Fusion of Multiple Relative and Absolute Measurements for Localization and Mapping“. In: *15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. 2018.
- [Emt19a] T. Emter und J. Petereit. „Simultaneous Localization and Mapping for Exploration with Stochastic Cloning EKF“. In: *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. 2019.
- [Emt19b] T. Emter, A. Schirg, P. Woock und J. Petereit. „Stochastic Cloning for Robust Fusion of Multiple Relative and Absolute Measurements“. In: *IEEE Intelligent Vehicles Symposium (IV)*. 2019.

- [Eng14] J. Engel, T. Schöps und D. Cremers. „LSD-SLAM: Large-Scale Direct Monocular SLAM“. In: *European Conference on Computer Vision (ECCV)*. 2014.
- [Eus05] R. Eustice, M. Walter und J. Leonard. „Sparse Extended Information Filters: Insights into Sparsification“. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2005.
- [Fai09] N. Fairfield. „Localization, Mapping, and Planning in 3D Environments“. Diss. Carnegie Mellon University, 2009.
- [Far08] J. Farrell. *Aided Navigation: GPS with High Rate Sensors*. McGraw-Hill New York, 2008.
- [For15] C. Forster, L. Carlone, F. Dellaert und D. Scaramuzza. „IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation“. In: *Robotics: Science and Systems*. 2015.
- [Fox03] D. Fox, J. Hightower, L. Liao, D. Schulz und G. Borriello. „Bayesian Filtering for Location Estimation“. In: *IEEE Pervasive Computing* 2.3 (2003).
- [Fre10] U. Frese. „Interview: Is SLAM solved?“ In: *Künstliche Intelligenz*. Bd. 24. 2010.
- [Fre15] C. Frey, T. Emter, J. Petereit, I. Tchouchenkov, T. Müller, M. Tittel, R. Worst, K. Pfeiffer, M. Walter, J. Wöllenstein, S. Rademacher, A. Wenzel und F. Müller. „Situation Responsive Networking of Mobile Robots for Disaster Management“. In: *Proceedings of the IEEE International Symposium on Technologies for Homeland Security*. 2015.
- [Gei12] A. Geiger, P. Lenz und R. Urtasun. „Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite“. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

- [Gei13] A. Geiger, P. Lenz, C. Stiller und R. Urtasun. „Vision Meets Robotics: The KITTI Dataset“. In: *International Journal of Robotics Research (IJRR)* (2013).
- [Gil06] A. Gil, O. Reinoso, O. Mozos, C. Stachniss und W. Burgard. „Improving Data Association in Vision-based SLAM“. In: *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2006.
- [Goe99] P. Goel, S. I. Roumeliotis und G. S. Sukhatme. „Robust Localization Using Relative and Absolute Position Estimates“. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1999.
- [Gre10] M. S. Grewal M. S. und A. P. Andrews. „Applications of Kalman Filtering in Aerospace 1960 to the Present“. In: *IEEE Control Systems Magazine* (2010).
- [Gre14] M. S. Grewal und A. P. Andrews. *Kalman Filtering: Theory and Practice with MATLAB*. 4th. Wiley-IEEE Press, 2014.
- [Gri05] G. Grisetti, C. Stachniss und W. Burgard. „Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling“. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*. 2005.
- [Gri07] G. Grisetti, C. Stachniss und W. Burgard. „Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters“. In: *IEEE Transactions on Robotics* 23.1 (2007).
- [Gri10] G. Grisetti, R. Kümmerle, C. Stachniss und W. Burgard. „A Tutorial on Graph-Based SLAM“. In: *Intelligent Transportation Systems Magazine, IEEE* 2.4 (2010).
- [Gro02] B. Grocholsky. „Information-Theoretic Control of Multiple Sensor Platforms“. Diss. University of Sydney, 2002.

- [Gro08] P. Groves. *Principles of GNSS, Inertial, and Multi-sensor Integrated Navigation Systems*. GNSS technology and applications series. Artech House, 2008.
- [Grö19] F. Gröger. „Integration mehrerer relativer Sensoren mittels stochastischem Klonen für die Lokalisierung“. Masterarbeit. Karlsruher Institut für Technologie (KIT), Institut für Industrielle Informationstechnik (IIT), 2019.
- [GTSAM] *GTSAM / Georgia Tech Smoothing and Mapping library*. URL: <https://github.com/borglab/gtsam> (besucht am 19. Mai 2019).
- [Gu16] Y. Gu, J. N. Gross, M. B. Rhudy und K. Lassak. „A Fault-Tolerant Multiple Sensor Fusion Approach Applied to UAV Attitude Estimation“. In: *International Journal of Aerospace Engineering* 2016 (2016).
- [Gui01] J. E. Guivant und E. M. Nebot. „Optimization of the Simultaneous Localization and Map-building Algorithm for Real-time Implementation“. In: *IEEE Transactions on Robotics and Automation* 17.3 (2001).
- [Gui02a] J. E. Guivant, F. R. Masson und E. M. Nebot. „Simultaneous Localization and Map Building Using Natural Features and Absolute Information“. In: *Robotics and Autonomous Systems* 40.2–3 (2002).
- [Gui02b] J. E. Guivant. „Efficient Simultaneous Localization and Mapping in Large Environments“. Diss. The University of Sydney, 2002.
- [Gui11] E. Guizzo. *How Google’s Self-Driving Car Works*. 2011. URL: <http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works>.

- [Gut00] J.-S. Gutmann. „Robuste Navigation Autonomer Mobiler Systeme“. Diss. Albert-Ludwigs-Universität Freiburg, 2000.
- [Gut99] J. S. Gutmann und K. Konolige. „Incremental Mapping of Large Cyclic Environments“. In: *Proceedings of the 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*. 1999.
- [Häh03] D. Hähnel, W. Burgard, D. Fox und S. Thrun. „An Efficient FastSLAM Algorithm for Generating Maps of Large-scale Cyclic Environments from Raw Laser Range Measurements“. In: *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Bd. 1. 2003.
- [Hei19] N. F. Heide, A. Albrecht, T. Emter und J. Petereit. „Performance Optimization of Autonomous Platforms in Unstructured Outdoor Environments Using a Novel Constrained Planning Approach“. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019.
- [Hin98] J. L. Hintze und R. D. Nelson. „Violin Plots: A Box Plot-Density Trace Synergism“. In: *The American Statistician* 52.2 (1998).
- [Ho05] K. Ho und P. Newman. „Combining Visual and Spatial Appearance for Loop Closure Detection in SLAM“. In: *Proceedings of the 2005 European Conference on Mobile Robots (ECMR)*. 2005.
- [Ho64] Y. Ho und R. Lee. „A Bayesian Approach to Problems in Stochastic Estimation and Control“. In: *IEEE Transactions on Automatic Control* 9.4 (1964).
- [Hof99] B. Hoffman, E. Baumgartner, T. Huntsberger und P. Shenker. „Improved State Estimation in Challenging Terrain“. In: *Autonomous Robots* 6 (1999).

- [Ind12] V. Indelman, S. Williams, M. Kaess und F. Dellaert. „Factor Graph Based Incremental Smoothing in Inertial Navigation Systems“. In: *2012 15th International Conference on Information Fusion*. 2012.
- [Ind13] V. Indelman, S. Williams, M. Kaess und F. Dellaert. „Information Fusion in Navigation Systems via Factor Graph Based Incremental Smoothing“. In: *Robotics and Autonomous Systems* 61.8 (2013).
- [Jo18] K. Jo, C. Kim und M. Sunwoo. „Simultaneous Localization and Map Change Update for the High Definition Map-Based Autonomous Driving Car“. In: *Sensors*. 2018.
- [Jul04a] S. Julier und J. Uhlmann. „Corrections to „Unscented Filtering and Nonlinear Estimation““. In: *Proceedings of the IEEE* 92.12 (2004).
- [Jul04b] S. Julier und J. Uhlmann. „Unscented Filtering and Nonlinear Estimation“. In: *Proceedings of the IEEE* 92.3 (2004).
- [Jul95] S. J. Julier, J. K. Uhlmann und H. F. Durrant-Whyte. „A New Approach for Filtering Nonlinear Systems“. In: *Proceedings of 1995 American Control Conference - ACC'95*. Bd. 3. 1995.
- [Jul97] S. J. Julier und J. K. Uhlmann. „A New Extension of the Kalman Filter to Nonlinear Systems“. In: *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulations and Controls*. 1997.
- [Kae12] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard und F. Dellaert. „iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree“. In: *The International Journal of Robotics Research* 31.2 (2012).
- [Kal60] R. E. Kalman. „A New Approach to Linear Filtering and Prediction Problems“. In: *Transactions of the ASME—Journal of Basic Engineering* 82.Series D (1960).

- [Kat06] R. Katz, N. Melkumyan, J. Guivant, T. Bailey, J. Nieto und E. Nebot. „Integrated Sensing Framework for 3D Mapping in Outdoor Navigation“. In: *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2006.
- [Klä12] J. Kläß, J. Stückler und S. Behnke. „Efficient Mobile Robot Navigation Using 3D Surfel Grid Maps“. In: *Proceedings of ROBOTIK 2012 7th German Conference on Robotics*. VDE. 2012.
- [Kle11] M. Kleinert, C. Ascher, S. Schleith, G.F. Trommer und U. Stilla. „A New Pedestrian Navigation Algorithm Based on the Stochastic Cloning Technique for Kalman Filtering“. In: *Proceedings of the 2011 International Technical Meeting of the Institute of Navigation, ION-ITM*. 2011.
- [Kle20] D. Kleiser, A. Albrecht, T. Emter, A. Zube, J. Petereit und P. Woock. „Mapping Shallow Water Environments using a Semi-Autonomous Multi-Sensor Surface Vehicle“. In: *Global Oceans 2020: Singapore - U.S. Gulf Coast*. 2020.
- [Kno03] A. Knoll und T. Christaller. *Robotik*. Fischer kompakt. Fischer-Taschenbuch-Verlag, 2003.
- [Koh11] S. Kohlbrecher, J. Meyer, O. von Stryk und U. Klingauf. „A Flexible and Scalable SLAM System with Full 3D Motion Estimation“. In: *Proceedings of the 2011 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE. 2011.
- [Kon09] K. Konolige, M. Agrawal, M. R. Blas, R. C. Bolles, B. Gerkey, J. Solà und A. Sundaresan. „Mapping, Navigation, and Learning for Off-road Traversal“. In: *Journal of Field Robotics* 26.1 (2009). eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.20271>.

- [Kou17] J. Kouba, F. Lahaye und P. Tétreault. „Precise Point Positioning“. In: *Springer Handbook of Global Navigation Satellite Systems*. Hrsg. von P. J. Teunissen und O. Montenbruck. Cham: Springer International Publishing, 2017.
- [Kro04] K. Kroschel. *Statistische Informationstechnik*. 4. Aufl. Springer Verlag Berlin Heidelberg New York, 2004.
- [Kun14] H.-B. Kuntze, C. Frey, T. Emter, J. Peterleit, I. Tchouchenkov, T. Mueller, M. Tittel, R. Worst, K. Pfeiffer, M. Walter, S. Rademacher und F. Mueller. „Situation Responsive Networking of Mobile Robots for Disaster Management“. In: *Proceedings of ISR/Robotik 2014; 41st International Symposium on Robotics*. 2014.
- [LaV03] J. J. LaViola. „A Comparison of Unscented and Extended Kalman Filtering for Estimating Quaternion Motion“. In: *Proceedings of the 2003 American Control Conference, 2003*. Bd. 3. 2003.
- [Lef82] E. Leffens, F. Markley und M. Shuster. „Kalman Filtering for Spacecraft Attitude Estimation“. In: *Journal of Guidance, Control, and Dynamics* 5.5 (1982).
- [Leo91] J. J. Leonard und H. F. Durrant-Whyte. „Simultaneous Map Building and Localization for an Autonomous Mobile Robot“. In: *Proceedings IROS '91: IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*. Bd. 3. 1991.
- [Lu97] F. Lu und E. Milios. „Globally Consistent Range Scan Alignment for Environment Mapping“. In: *Autonomous Robots* 4.4 (1997).
- [Lyn13] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli und R. Siegwart. „A Robust and Modular Multi-sensor Fusion Approach Applied to MAV Navigation“. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013.

- [Mag07] M. Magnusson, A. Lilienthal und T. Duckett. „Scan Registration for Autonomous Mining Vehicles Using 3D-NDT“. In: *Journal of Field Robotics* 24.10 (2007).
- [Mag09] M. Magnusson. „The Three-Dimensional Normal-Distributions Transform – an Efficient Representation for Registration, Surface Analysis, and Loop Detection“. Diss. Örebro Universitet, 2009.
- [Mak02] A. A. Makarenko, S. B. Williams, F. Bourgault und H. F. Durrant-Whyte. „An Experiment in Integrated Exploration“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Bd. 1. 2002.
- [McG85] L. A. McGee und S. F. Schmidt. *Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry*. Techn. Ber. NASA, 1985. URL: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19860003843.pdf>.
- [Mer04] R. van der Merwe und E. A. Wan. „Sigma-point Kalman Filters for Probabilistic Inference in Dynamic State-space Models“. In: *Workshop on Advances in Machine Learning*. Montreal, 2004.
- [Moh01] A. P. A. Mohinder S. Grewal Lawrence R. Weill. *Global Positioning Systems, Inertial Navigation, and Integration*. New York: Wiley, 2001.
- [Mon02] M. Montemerlo, S. Thrun, D. Koller und B. Wegbreit. „Fast-SLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem“. In: *AAAI-02 Proceedings* (2002).
- [Mon03a] M. Montemerlo und S. Thrun. „Simultaneous Localization and Mapping with Unknown Data Association Using Fast-SLAM“. In: *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA)*. Bd. 2. 2003.

- [Mon03b] M. Montemerlo, S. Thrun, D. Roller und B. Wegbreit. „Fast-SLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping That Provably Converges“. In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence*. IJCAI'03. Acapulco, Mexico: Morgan Kaufmann Publishers Inc., 2003.
- [Mon07] M. Montemerlo und S. Thrun. *FastSLAM – A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*. Bd. 27. STAR – Springer Tracts in Advanced Robotics. Berlin Heidelberg New York: Springer-Verlag Berlin Heidelberg, 2007.
- [Moo11] F. Moosmann und C. Stiller. „Velodyne SLAM“. In: *Proceedings of the IEEE Intelligent Vehicles Symposium*. 2011.
- [Mor08] Y. Morales, E. Takeuchi und T. Tsubouchi. „Vehicle Localization in Outdoor Woodland Environments with Sensor Fault Detection“. In: *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA)*. 2008.
- [Mou06] A. Mourikis und S. Roumeliotis. „On the Treatment of Relative-pose Measurements for Mobile Robot Localization“. In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. 2006.
- [Mou07] A.I. Mourikis, S. Roumeliotis und J.W. Burdick. „SC-KF Mobile Robot Localization: A Stochastic Cloning Kalman Filter for Processing Relative-state Measurements“. In: *IEEE Transactions on Robotics* 23.4 (2007).
- [Mur15] R. Mur-Artal, J. M. M. Montiel und J. D. Tardós. „ORB-SLAM: A Versatile and Accurate Monocular SLAM System“. In: *IEEE Transactions on Robotics* 31.5 (2015).

- [Mur99] K. P. Murphy. „Bayesian Map Learning in Dynamic Environments“. In: *Proceedings of the 12th International Conference on Neural Information Processing Systems*. NIPS99. Denver, CO: MIT Press, 1999.
- [Nei01] J. Neira und J. D. Tardos. „Data Association in Stochastic Mapping Using the Joint Compatibility Test“. In: *IEEE Transactions on Robotics and Automation* 17.6 (2001).
- [Ngu10] D. V. Nguyen, L. Kuhnert, J. Schlemper und K. Kuhnert. „Terrain Classification Based on Structure for Autonomous Navigation in Complex Environments“. In: *International Conference on Communications and Electronics 2010*. 2010.
- [Nie03] J. Nieto, J. Guivant, E. Nebot und S. Thrun. „Real Time Data Association for FastSLAM“. In: *Proceedings of the 2003 IEEE Conference on Robotics and Automation, Taipeh*. 2003.
- [Nüc07] A. Nüchter, K. Lingemann, J. Hertzberg und H. Surmann. „6D SLAM–3D Mapping Outdoor Environments“. In: *Journal of Field Robotics* 24.8-9 (2007).
- [Nüc09] A. Nüchter. *3D Robotic Mapping – The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom*. Bd. 52. STAR – Springer Tracts in Advanced Robotics. Berlin Heidelberg New York: Springer-Verlag Berlin Heidelberg, 2009.
- [Pau18] J. Pauls, T. Strauss, C. Hasberg, M. Lauer und C. Stiller. „Can We Trust Our Maps? An Evaluation of Road Changes and a Dataset for Map Validation“. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018.
- [Pér02] P. Pérez, C. Hue, J. Vermaak und M. Gangnet. „Color-Based Probabilistic Tracking“. In: *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*. London, UK: Springer-Verlag, 2002.

- [Pet13] J. Petereit, T. Emter und C. W. Frey. „Mobile Robot Motion Planning in Multi-resolution Lattices with Hybrid Dimensionality“. In: *Proceedings of the IFAC Intelligent Autonomous Vehicles Symposium*. 2013.
- [Pet14] J. Petereit, T. Emter und C. W. Frey. „Combined Trajectory Generation and Path Planning for Mobile Robots Using Lattices with Hybrid Dimensionality“. In: *Robot Intelligence Technology and Applications 2. Results from the 2nd International Conference on Robot Intelligence Technology and Applications*. Hrsg. von J.-H. Kim, E. T. Matson, H. Myung, P. Xu und F. Karray. Advances in Intelligent Systems and Computing 274. Springer International Publishing Switzerland, 2014.
- [Pet19] J. Petereit, J. Beyerer, T. Asfour, S. Gentes, B. Hein, U. D. Hanebeck, F. Kirchner, R. Dillmann, H. H. Götting, M. Weiser, M. Gustmann und T. Eglhoffstein. „ROBDEKON: Robotic Systems for Decontamination in Hazardous Environments“. In: *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. 2019.
- [Pic16] R. Piché. „Automatic Numerical Differentiation by Maximum Likelihood Estimation of State-space Model“. In: 2016.
- [Pog18] F. Poggenhans, N. O. Salscheider und C. Stiller. „Precise Localization in High-Definition Road Maps for Urban Regions“. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018.
- [Pom13] F. Pomerleau, F. Colas, R. Siegwart und S. Magnenat. „Comparing ICP Variants on Real-world Data Sets Open-source Library and Experimental Protocol“. In: *Autonomous Robots* 34.3 (2013).
- [Por14] D. Portugal, L. Marques und M. Armada. „Deploying Field Robots For Humanitarian Demining: Challenges, Requirements And Research Trends“. In: *Mobile Service Robotics*.

2014. eprint: https://www.worldscientific.com/doi/pdf/10.1142/9789814623353_0075.
- [Qui09] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler und A. Y. Ng. „ROS: An Open-source Robot Operating System“. In: *IEEE International Conference on Robotics and Automation: Workshop on Open Source Software*. 2009.
- [Ram07] F. Ramos, J. Nieto und H. Durrant-Whyte. „Recognising and Modelling Landmarks to Close Loops in Outdoor SLAM“. In: *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*. 2007.
- [Rei79] D. Reid. „An Algorithm for Tracking Multiple Targets“. In: *IEEE Transactions on Automatic Control* 24 (6 1979).
- [Rig07] G. G. Rigatos. „Extended Kalman and Particle Filtering for Sensor Fusion in Mobile Robot Localization“. In: *IEEE International Conference on Physics and Control*. Potsdam, 2007.
- [ROS] *ROS.org | Powering the world's robots*. URL: <http://www.ros.org> (besucht am 19. Mai 2019).
- [Rou01] S. I. Roumeliotis. *A Kalman Filter for Processing 3-D Relative Pose Measurements*. Techn. Ber. Robotics Laboratory, California Institute of Technology, 2001. URL: https://www-users.cs.umn.edu/~stergios/tech_reports/relative_3d_kf.pdf.
- [Rou02] S. I. Roumeliotis und J. W. Burdick. „Stochastic Cloning: A Generalized Framework for Processing Relative State Measurements“. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 2002.
- [Rou99] S. I. Roumeliotis, G. S. Sukhatme und G. A. Bekey. „Circumventing Dynamic Modeling: Evaluation of the Error-state Kalman filter Applied to Mobile Robot Localization“. In: *Pro-*

- ceedings of the IEEE International Conference on Robotics and Automation*. 1999.
- [Rub88] D. Rubin. „Using the SIR Algorithm to Simulate Posterior Distributions“. In: *Bayesian Statistics 3* (1988). Hrsg. von M. Bernardo, K. DeGroot, D. Lindley und A. Smith.
- [Rus01] S. Rusinkiewicz und M. Levoy. „Efficient Variants of the ICP Algorithm“. In: *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*. 2001.
- [Rus09] S. Russell und P. Norvig. *Artificial Intelligence: A Modern Approach*. 3rd. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009.
- [Rus11] R. B. Rusu und S. Cousins. „3D is here: Point Cloud Library (PCL)“. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China, 2011.
- [Saa13] J. Saarinen, H. Andreasson, T. Stoyanov, J. Ala-Luhtala und A. Lilienthal. „Normal Distributions Transform Occupancy Maps: Application to large-scale online 3D mapping“. In: *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*. 2013.
- [Sär13] S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.
- [Sch13] M. Scherer. „Visualisierung von 3D-NDT-Karten und Entwicklung von Methoden zur probabilistischen Integration unbelegter Volumenelemente in den Kartierungsprozess“. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), Institut für Anthropomatik und Robotik (IAR), Lehrstuhl für Interaktive Echtzeitsysteme (IES), 2013.
- [Sch14] M. Schadler, J. Stückler und S. Behnke. „Rough Terrain 3D Mapping and Navigation Using a Continuously Rotating 2D Laser Scanner“. In: *KI - Künstliche Intelligenz* 28.2 (2014).

- [Sch19] A. Schirg. „Erweiterung EKF-basiertes Lokalisierungsfilter durch Einsatz von Stochastic Cloning“. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), Institut für Anthropomatik und Robotik (IAR), Lehrstuhl für Interaktive Echtzeitsysteme (IES), 2019.
- [Seg09] A. Segal, D. Haehnel und S. Thrun. „Generalized-ICP.“ In: *Robotics: Science and Systems*. Bd. 2. 2009.
- [Sha18a] A. U. Shamsudin, K. Ohno, R. Hamada, S. Kojima, T. Westfechtel, T. Suzuki, Y. Okada, S. Tadokoro, J. Fujita und H. Amano. „Consistent Map Building in Petrochemical Complexes for Firefighter Robots Using SLAM Based on GPS and LIDAR“. In: *ROBOMECH Journal* 5.1 (2018).
- [Sha18b] T. Shan und B. Englot. „LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.
- [Sic08] B. Siciliano und O. Khatib, Hrsg. *Springer Handbook of Robotics*. Springer, 2008.
- [Sie04] R. Siegwart und I. Nourbakhsh, Hrsg. *Introduction to Autonomous Mobile Robots*. MIT Press, 2004.
- [Sim05] R. Sim, P. Elinas und M. Griffin. „Vision-based SLAM Using the Rao-blackwellised Particle Filter“. In: *In IJCAI Workshop on Reasoning with Uncertainty in Robotics*. 2005.
- [Sim06] D. Simon. *Optimal State Estimation: Kalman, H_∞ , and Nonlinear Approaches*. New Jersey: John Wiley & Sons, 2006.
- [Smi78] A. R. Smith. „Color Gamut Transform Pairs“. In: *SIGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1978.

- [Sta08] C. Stachniss, M. Bennewitz, G. Grisetti, S. Behnke und W. Burgard. „How to Learn Accurate Grid Maps with a Humanoid“. In: *2008 IEEE International Conference on Robotics and Automation*. 2008.
- [Ste08] B. Steder, G. Grisetti, C. Stachniss und W. Burgard. „Visual SLAM for Flying Vehicles“. In: *IEEE Transactions on Robotics* 24.5 (2008).
- [Ste12] A. Stelzer, H. Hirschmüller und M. Görner. „Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain“. In: *The International Journal of Robotics Research* 31.4 (2012).
- [Sto13] T. Stoyanov, M. Magnusson und A. J. Lilienthal. „Comparative Evaluation of the Consistency of Three-dimensional Spatial Representations used in Autonomous Robot Navigation“. In: *Journal of Field Robotics* 30.2 (2013).
- [Sur03] H. Surmann, A. Nüchter und J. Hertzberg. „An Autonomous Mobile Robot with a 3D Laser Range Finder for 3D Exploration and Digitalization of Indoor Environments“. In: *Robotics and Autonomous Systems* 45.3 (2003).
- [Tak06] E. Takeuchi und T. Tsubouchi. „A 3-D Scan Matching using Improved 3-D Normal Distributions Transform for Mobile Robotic Mapping“. In: *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2006.
- [Thr04] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani und H. Durrant-Whyte. „Simultaneous Localization and Mapping with Sparse Extended Information Filters“. In: *The International Journal of Robotics Research* 23.7-8 (2004).
- [Thr05] S. Thrun, W. Burgard und D. Fox. *Probabilistic Robotics*. Cambridge, Massachusetts: The MIT Press, 2005.

- [Thr06a] S. Thrun und M. Montemerlo. „The Graph SLAM Algorithm with Applications to Large-scale Mapping of Urban Structures“. In: *The International Journal of Robotics Research* 25.5-6 (2006).
- [Thr06b] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian und P. Mahoney. „Stanley: The Robot that Won the DARPA Grand Challenge“. In: *Journal of Field Robotics* 23.9 (2006).
- [Tri06] R. Triebel, P. Pfaff und W. Burgard. „Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing“. In: *Proceedings of the 2006 International Conference on Intelligent Robots and Systems*. 2006.
- [Tsa17] N. G. Tsagarakis, D. G. Caldwell, F. Negrello, W. Choi, L. Baccelliere, V. Loc, J. Noorden, L. Muratore, A. Margan, A. Cardellino, L. Natale, E. Mingo Hoffman, H. Dallali, N. Kashiri, J. Malzahn, J. Lee, P. Kryczka, D. Kanoulas, M. Garabini, M. Catalano, M. Ferrati, V. Varricchio, L. Pallottino, C. Pavan, A. Bicchi, A. Settini, A. Rocchi und A. Ajoudani. „WALK-MAN: A High-Performance Humanoid Platform for Realistic Environments“. In: *Journal of Field Robotics* 34.7 (2017).
- [Ulr10] T. Ulrich. „Robuste Extraktion und Wiedererkennung von Landmarken basierend auf Fusion geometrischer und visueller Informationen für die Navigation mobiler Roboter.“ Diplomarbeit. Karlsruher Institut für Technologie (KIT), Institut für Photogrammetrie und Fernerkundung (IPF), 2010.

- [Urm08] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y.-W. Seo, S. Singh, J. Snider, A. Stentz, W. “Red” Whittaker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, J. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms und D. Ferguson. „Autonomous Driving in Urban Environments: Boss and the Urban Challenge“. In: *Journal of Field Robotics* 25.8 (2008).
- [Wah14] Wahyono, L. Kurnianggoro, J. Hariyono und K. Jo. „Traffic Sign Recognition System for Autonomous Vehicle Using Cascade SVM Classifier“. In: *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*. 2014.
- [Wel95] G. Welch und G. Bishop. *An Introduction to the Kalman Filter*. Techn. Ber. Chapel Hill, NC, USA, 1995.
- [Wel96] G. F. Welch. „SCAAT: Incremental Tracking with Incomplete Information“. Diss. The University of North Carolina at Chapel Hill, 1996.
- [Wel97] G. Welch und G. Bishop. „SCAAT: Incremental Tracking with Incomplete Information“. In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '97. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997.
- [Wen11] J. Wendel. *Integrierte Navigationssysteme: Sensordatenfusion, GPS und Inertiale Navigation*. 2. Aufl. Oldenbourg Wissenschaftsverlag GmbH, 2011.
- [Wil12] V. V. Williams. „Multiplying Matrices Faster Than Coppersmith-winograd“. In: *Proceedings of the Forty-fourth Annual ACM*

- Symposium on Theory of Computing*. STOC '12. New York, New York, USA: ACM, 2012.
- [Wil14] S. Williams, V. Indelman, M. Kaess, R. Roberts, J. J. Leonard und F. Dellaert. „Concurrent Filtering and Smoothing: A Parallel Architecture for Real-time Navigation and Full Smoothing“. In: *The International Journal of Robotics Research* 33.12 (2014).
- [Win09] H. Winner, S. Hakuli und G. Wolf. *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort : mit 550 Abbildungen und 45 Tabellen*. ATZ-MTZ-Fachbuch. Vieweg + Teubner, 2009.
- [Wur10] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss und W. Burgard. „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*. Bd. 2. 2010.
- [Yan18] Z. Yang, J. Li und H. Li. „Real-time Pedestrian and Vehicle Detection for Autonomous Driving“. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. 2018.
- [Ye17] H. Ye und M. Liu. „LiDAR and Inertial Fusion for Pose Estimation by Non-linear Optimization“. In: *CoRR* abs/1710.07104 (2017). arXiv: 1710.07104.
- [Zal14] V. Zaliva und F. Franchetti. „Barometric and GPS Altitude Sensor Fusion“. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014.
- [Zha14a] H. Zhang, J. Zheng, G. Dorr, H. Zhou und Y. Ge. „Testing of GPS Accuracy for Precision Forestry Applications“. In: *Ara-bian Journal for Science and Engineering* 39.1 (2014).

- [Zha14b] J. Zhang und S. Singh. „LOAM: Lidar Odometry and Mapping in Real-time“. In: *Robotics: Science and Systems Conference*. 2014.
- [Zie14] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, E. Kaus, R. G. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M.ENZweiler, C. Knöppel, J. Hipp, M. Hauéis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein und E. Zeeb. „Making Bertha Drive – An Autonomous Journey on a Historic Route“. In: *IEEE Intelligent Transportation Systems Magazine* 6.2 (Summer 2014).

Liste der Publikationen

- [Kle20] D. Kleiser, A. Albrecht, T. Emter, A. Zube, J. Petereit und P. Woock. „Mapping Shallow Water Environments using a Semi-Autonomous Multi-Sensor Surface Vehicle“. In: *Global Oceans 2020: Singapore - U.S. Gulf Coast*. 2020.
- [Emt19a] T. Emter und J. Petereit. „Simultaneous Localization and Mapping for Exploration with Stochastic Cloning EKF“. In: *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. 2019.
- [Emt19b] T. Emter, A. Schirg, P. Woock und J. Petereit. „Stochastic Cloning for Robust Fusion of Multiple Relative and Absolute Measurements“. In: *IEEE Intelligent Vehicles Symposium (IV)*. 2019.
- [Hei19] N.F. Heide, A. Albrecht, T. Emter und J. Petereit. „Performance Optimization of Autonomous Platforms in Unstructured Outdoor Environments Using a Novel Constrained Planning Approach“. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019.
- [Emt18a] T. Emter und J. Petereit. „3D SLAM With Scan Matching and Factor Graph Optimization“. In: *ISR 2018; 50th International Symposium on Robotics*. 2018.

- [Emt18b] T. Emter und J. Petereit. „Stochastic Cloning and Smoothing for Fusion of Multiple Relative and Absolute Measurements for Localization and Mapping“. In: *15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. 2018.
- [Emt17] T. Emter, C. Frese, A. Zube und J. Petereit. „Algorithmen-Toolbox für autonome mobile Robotersysteme“. In: *ATZOff-highway (2017)* 10 (2017).
- [Cur15] F. Curatella, P. Vinetti, G. Rizzo, T. Vladimirova, L. D. Vendicis, T. Emter, J. Petereit, C. W. Frey, D. Usher, I. Stanciugelu, J. Schaefer, E. den Breejen, L. Gisslén und D. Letalick. „Toward a Multifaceted Platform for Humanitarian Demining“. In: *13th IARP Workshop on Humanitarian Demining and Risky Intervention*. 12th International Symposium „MINE ACTION 2015“. 2015.
- [Fre15] C. Frey, T. Emter, J. Petereit, I. Tchouchenkov, T. Müller, M. Tittel, R. Worst, K. Pfeiffer, M. Walter, J. Wöllenstein, S. Rademacher, A. Wenzel und F. Müller. „Situation Responsive Networking of Mobile Robots for Disaster Management“. In: *Proceedings of the IEEE International Symposium on Technologies for Homeland Security*. 2015.
- [Emt14] T. Emter und J. Petereit. „Integrated Multi-Sensor Fusion for Mapping and Localization in Outdoor Environments for Mobile Robots“. In: *Proceedings SPIE 9121, Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2014*. 2014.
- [Kun14] H.-B. Kuntze, C. Frey, T. Emter, J. Petereit, I. Tchouchenkov, T. Mueller, M. Tittel, R. Worst, K. Pfeiffer, M. Walter, S. Rademacher und F. Mueller. „Situation Responsive Networking of Mobile Robots for Disaster Management“. In: *Proceedings*

of *ISR/Robotik 2014; 41st International Symposium on Robotics*. 2014.

- [Pet14] J. Petereit, T. Emter und C. W. Frey. „Combined Trajectory Generation and Path Planning for Mobile Robots Using Lattices with Hybrid Dimensionality“. In: *Robot Intelligence Technology and Applications 2. Results from the 2nd International Conference on Robot Intelligence Technology and Applications*. Hrsg. von J.-H. Kim, E. T. Matson, H. Myung, P. Xu und F. Karray. Advances in Intelligent Systems and Computing 274. Springer International Publishing Switzerland, 2014.
- [Pet13] J. Petereit, T. Emter und C. W. Frey. „Mobile Robot Motion Planning in Multi-resolution Lattices with Hybrid Dimensionality“. In: *Proceedings of the IFAC Intelligent Autonomous Vehicles Symposium*. 2013.
- [Emt12a] T. Emter. *Integrated Multi-Sensor Fusion and SLAM for Mobile Robots*. Techn. Ber. Fraunhofer IOSB und Institute for Anthropomatics, Vision and Fusion Laboratory, 2012.
- [Emt12b] T. Emter und T. Ulrich. „Fusion of Geometrical and Visual Information for Localization and Mapping in Outdoor Environments“. In: *Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS)*. 2012.
- [Emt11] T. Emter. *Probabilistic Localization and Mapping for Mobile Robots*. Techn. Ber. Fraunhofer IOSB und Institute for Anthropomatics, Vision and Fusion Laboratory, 2011.
- [Emt10a] T. Emter. *Probabilistic Localization and Mapping for Mobile Robots*. Techn. Ber. Fraunhofer IOSB und Institute for Anthropomatics, Vision and Fusion Laboratory, 2010.
- [Emt10b] T. Emter, A. Saltoğlu und J. Petereit. „Multi-Sensor Fusion for Localization of a Mobile Robot in Outdoor Environments.“ In: *Proc. VDE-Verlag: ISR/Robotik 2010: Visions are Reality*. (2010).

- [Emt10c] T. Emter und T. Ulrich. „Visuelle Information zur robusten Zuordnung von Landmarken für die Navigation mobiler Roboter“. In: *Forum Bildverarbeitung*. Hrsg. von F. Puente León und M. Heizmann. 2010.
- [Emt08] T. Emter, C. Frey und H.-B. Kuntze. „Multisensorielle Überwachung von Liegenschaften durch mobile Roboter – Multi-Sensor Surveillance of Real Estates Based on Mobile Robots“. In: *Robotik 2008: Leistungsstand – Anwendungen – Visionen – Trends*. VDI/VDE-GMA, 2008.
- [Emt07] T. Emter, E. Monari, C. Frey, T. Müller, H.-B. Kuntze, A. Laubenheimer und M. Müller. „AMROS – an Autonomous Mobile Robotic System for Multisensor Surveillance of Real Estates“. In: *Proceedings of the Future Security: 2nd Security Research Conference*. Hrsg. von J. Beyerer. Fraunhofer Verbund für Verteidigungs- und Sicherheitsforschung, 2007.

Liste der betreuten studentischen Abschlussarbeiten

- [Grö19] F. Gröger. „Integration mehrerer relativer Sensoren mittels stochastischem Klonen für die Lokalisierung“. Masterarbeit. Karlsruher Institut für Technologie (KIT), Institut für Industrielle Informationstechnik (IIT), 2019.
- [Sch19] A. Schirg. „Erweiterung EKF-basiertes Lokalisierungsfilter durch Einsatz von Stochastic Cloning“. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), Institut für Anthropomatik und Robotik (IAR), Lehrstuhl für Interaktive Echtzeitsysteme (IES), 2019.
- [Gri18] S. Griffiths. „Integration eines Moving-Baseline-DGPS in ein Lokalisierungssystem für mobile Roboter“. Bachelorarbeit. Hochschule Karlsruhe Technik und Wirtschaft, Fakultät für Maschinenbau und Mechatronik (MMT), 2018.
- [Hei17] N.F. Heide. „Kalibrierung mehrerer 3D-Laserscanner auf ein gemeinsames Fahrzeugkoordinatensystem“. Masterar-

- beit. Karlsruher Institut für Technologie (KIT), Institut für Industrielle Informationstechnik (IIT), 2017.
- [Sün17] C. Sünkenberg. „Echtzeitfähige Parallelisierung, GPU-Implementierung und Evaluation eines Stereokorrespondenzalgorithmus auf multispektralen Bilddaten“. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), Institut für Anthropomatik und Robotik (IAR), Lehrstuhl für Interaktive Echtzeitsysteme (IES), 2017.
- [Neh14] A. Nehaichik. „Befahrbarkeitschätzung mittels Fusion von LIDAR- und Bilddaten“. Studienarbeit. Universität Stuttgart, Institut für Photogrammetrie, 2014.
- [Bar13] S. Barth. „Asynchrones und echtzeitfähiges Verfahren zur Multisensorfusion und SLAM auf einem mobilen Roboter“. Bachelorarbeit. Hochschule Karlsruhe Technik und Wirtschaft, Fakultät Informatik und Wirtschaftsinformatik (IWI), 2013.
- [Sch13] M. Scherer. „Visualisierung von 3D-NDT-Karten und Entwicklung von Methoden zur probabilistischen Integration unbelegter Volumenelemente in den Kartierungsprozess“. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), Institut für Anthropomatik und Robotik (IAR), Lehrstuhl für Interaktive Echtzeitsysteme (IES), 2013.
- [Rie12] S. Riedmüller. „Multi-LIDAR-Hinderniserkennung für einen mobilen Roboter“. Bachelorarbeit. Fachhochschule Bingen, Fachbereich 2 Technik, Informatik und Wirtschaft, 2012.
- [Rup11a] V. Rupp. „Entwicklung eines SLAM-Algorithmus auf Basis von 3D-Laserscans“. Diplomarbeit. Karlsruher Institut für Technologie (KIT), Institut für Photonik und Quantenelektronik (IPQ), 2011.

- [Ste11a] A. Stein. „Entwicklung eines probabilistischen 3D-SLAM-Verfahrens auf Basis des Kinect-Sensors“. Diplomarbeit. Karlsruher Institut für Technologie (KIT), Institut für Photonik und Quantenelektronik (IPQ), 2011.
- [Kis10] K. Kischlat. „Schätzung des befahrbaren Nahbereichs für einen mobilen Roboter mittels einer Laufzeitkamera“. Masterarbeit. Hochschule für Technik, Wirtschaft und Gestaltung Konstanz, Fakultät Informatik, 2010.
- [Kle10] M. Klees. „Visual Servoing zur Verfolgung einer Person mit einem mobilen Roboter“. Studienarbeit. Karlsruher Institut für Technologie (KIT), Institut für Angewandte Informatik / Automatisierungstechnik (AIA), 2010.
- [Ulr10] T. Ulrich. „Robuste Extraktion und Wiedererkennung von Landmarken basierend auf Fusion geometrischer und visueller Informationen für die Navigation mobiler Roboter.“ Diplomarbeit. Karlsruher Institut für Technologie (KIT), Institut für Photogrammetrie und Fernerkundung (IPF), 2010.
- [Pet09] J. Petereit. „Entwicklung eines Fusionsalgorithmus und Performance-Evaluierung für ein integriertes Navigationssystem“. Studienarbeit. Karlsruher Institut für Technologie (KIT), Institut für Photonik und Quantenelektronik (IPQ), 2009.
- [Sal09] A. Saltoğlu. „Sensor fusion for localization of a mobile robot for outdoor applications“. Diplomarbeit. Karlsruher Institut für Technologie (KIT), Institut für Photonik und Quantenelektronik (IPQ), 2009.

Karlsruher Schriftenreihe zur Anthropomatik (ISSN 1863-6489)

- Band 1** Jürgen Geisler
Leistung des Menschen am Bildschirmarbeitsplatz.
ISBN 3-86644-070-7
- Band 2** Elisabeth Peinsipp-Byma
Leistungserhöhung durch Assistenz in interaktiven Systemen zur Szenenanalyse.
ISBN 978-3-86644-149-1
- Band 3** Jürgen Geisler, Jürgen Beyerer (Hrsg.)
Mensch-Maschine-Systeme.
ISBN 978-3-86644-457-7
- Band 4** Jürgen Beyerer, Marco Huber (Hrsg.)
Proceedings of the 2009 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-86644-469-0
- Band 5** Thomas Usländer
Service-oriented design of environmental information systems.
ISBN 978-3-86644-499-7
- Band 6** Giulio Milighetti
Multisensorielle diskret-kontinuierliche Überwachung und Regelung humanoider Roboter.
ISBN 978-3-86644-568-0
- Band 7** Jürgen Beyerer, Marco Huber (Hrsg.)
Proceedings of the 2010 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-86644-609-0
- Band 8** Eduardo Monari
Dynamische Sensorselektion zur auftragsorientierten Objektverfolgung in Kameranetzwerken.
ISBN 978-3-86644-729-5

- Band 9** Thomas Bader
Multimodale Interaktion in Multi-Display-Umgebungen.
ISBN 3-86644-760-8
- Band 10** Christian Frese
Planung kooperativer Fahrmanöver für kognitive Automobile.
ISBN 978-3-86644-798-1
- Band 11** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2011 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-86644-855-1
- Band 12** Miriam Schleipen
Adaptivität und Interoperabilität von Manufacturing Execution Systemen (MES).
ISBN 978-3-86644-955-8
- Band 13** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2012 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-86644-988-6
- Band 14** Hauke-Hendrik Vagts
Privatheit und Datenschutz in der intelligenten Überwachung: Ein datenschutzgewährendes System, entworfen nach dem „Privacy by Design“ Prinzip.
ISBN 978-3-7315-0041-4
- Band 15** Christian Kühnert
Data-driven Methods for Fault Localization in Process Technology.
ISBN 978-3-7315-0098-8
- Band 16** Alexander Bauer
Probabilistische Szenenmodelle für die Luftbildauswertung.
ISBN 978-3-7315-0167-1
- Band 17** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2013 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0212-8

- Band 18** Michael Teutsch
Moving Object Detection and Segmentation for Remote Aerial Video Surveillance.
ISBN 978-3-7315-0320-0
- Band 19** Marco Huber
Nonlinear Gaussian Filtering: Theory, Algorithms, and Applications.
ISBN 978-3-7315-0338-5
- Band 20** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2014 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0401-6
- Band 21** Todor Dimitrov
Permanente Optimierung dynamischer Probleme der Fertigungssteuerung unter Einbeziehung von Benutzerinteraktionen.
ISBN 978-3-7315-0426-9
- Band 22** Benjamin Kühn
Interessengetriebene audiovisuelle Szenenexploration.
ISBN 978-3-7315-0457-3
- Band 23** Yvonne Fischer
Wissensbasierte probabilistische Modellierung für die Situationsanalyse am Beispiel der maritimen Überwachung.
ISBN 978-3-7315-0460-3
- Band 24** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2015 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0519-8
- Band 25** Pascal Birnstill
Privacy-Respecting Smart Video Surveillance Based on Usage Control Enforcement.
ISBN 978-3-7315-0538-9
- Band 26** Philipp Woock
Umgebungskartenschätzung aus Sidescan-Sonardaten für ein autonomes Unterwasserfahrzeug.
ISBN 978-3-7315-0541-9

- Band 27** Janko Petereit
Adaptive State × Time Lattices: A Contribution to Mobile Robot Motion Planning in Unstructured Dynamic Environments.
ISBN 978-3-7315-0580-8
- Band 28** Erik Ludwig Krempel
Steigerung der Akzeptanz von intelligenter Videoüberwachung in öffentlichen Räumen.
ISBN 978-3-7315-0598-3
- Band 29** Jürgen Moßgraber
Ein Rahmenwerk für die Architektur von Frühwarnsystemen.
ISBN 978-3-7315-0638-6
- Band 30** Andrey Belkin
World Modeling for Intelligent Autonomous Systems.
ISBN 978-3-7315-0641-6
- Band 31** Chettapong Janya-Anurak
Framework for Analysis and Identification of Nonlinear Distributed Parameter Systems using Bayesian Uncertainty Quantification based on Generalized Polynomial Chaos.
ISBN 978-3-7315-0642-3
- Band 32** David Münch
Begriffliche Situationsanalyse aus Videodaten bei unvollständiger und fehlerhafter Information.
ISBN 978-3-7315-0644-7
- Band 33** Jürgen Beyerer, Alexey Pak (Eds.)
Proceedings of the 2016 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0678-2
- Band 34** Jürgen Beyerer, Alexey Pak and Miro Taphanel (Eds.)
Proceedings of the 2017 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0779-6
- Band 35** Michael Grinberg
Feature-Based Probabilistic Data Association for Video-Based Multi-Object Tracking.
ISBN 978-3-7315-0781-9

- Band 36** Christian Herrmann
Video-to-Video Face Recognition for Low-Quality Surveillance Data.
ISBN 978-3-7315-0799-4
- Band 37** Chengchao Qu
Facial Texture Super-Resolution by Fitting 3D Face Models.
ISBN 978-3-7315-0828-1
- Band 38** Miriam Ruf
Geometrie und Topologie von Trajektorienoptimierung für vollautomatisches Fahren.
ISBN 978-3-7315-0832-8
- Band 39** Angelika Zube
Bewegungsregelung mobiler Manipulatoren für die Mensch-Roboter-Interaktion mittels kartesischer modellprädiktiver Regelung.
ISBN 978-3-7315-0855-7
- Band 40** Jürgen Beyerer and Miro Taphanel (Eds.)
Proceedings of the 2018 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0936-3
- Band 41** Marco Thomas Gewohn
Ein methodischer Beitrag zur hybriden Regelung der Produktionsqualität in der Fahrzeugmontage.
ISBN 978-3-7315-0893-9
- Band 42** Tianyi Guan
Predictive energy-efficient motion trajectory optimization of electric vehicles.
ISBN 978-3-7315-0978-3
- Band 43** Jürgen Metzler
Robuste Detektion, Verfolgung und Wiedererkennung von Personen in Videodaten mit niedriger Auflösung.
ISBN 978-3-7315-0968-4
- Band 44** Sebastian Bullinger
Image-Based 3D Reconstruction of Dynamic Objects Using Instance-Aware Multibody Structure from Motion.
ISBN 978-3-7315-1012-3

- Band 45** Jürgen Beyerer, Tim Zander (Eds.)
**Proceedings of the 2019 Joint Workshop of
Fraunhofer IOSB and Institute for Anthropomatics,
Vision and Fusion Laboratory.**
ISBN 978-3-7315-1028-4
- Band 46** Stefan Becker
Dynamic Switching State Systems for Visual Tracking.
ISBN 978-3-7315-1038-3
- Band 47** Jennifer Sander
**Ansätze zur lokalen Bayes'schen Fusion von
Informationsbeiträgen heterogener Quellen.**
ISBN 978-3-7315-1062-8
- Band 48** Philipp Christoph Sebastian Bier
**Umsetzung des datenschutzrechtlichen Auskunftsanspruchs
auf Grundlage von Usage-Control und Data-Provenance-
Technologien.**
ISBN 978-3-7315-1082-6
- Band 49** Thomas Emter
**Integrierte Multi-Sensor-Fusion für die simultane
Lokalisierung und Kartenerstellung für mobile
Robotersysteme.**
ISBN 978-3-7315-1074-1

Lehrstuhl für Interaktive Echtzeitsysteme
Karlsruher Institut für Technologie

Fraunhofer-Institut für Optronik, Systemtechnik
und Bildauswertung IOSB Karlsruhe

Für die autonome Navigation mobiler Robotersysteme sind gewisse Fähigkeiten notwendig. Die simultane Lokalisierung und Kartenerstellung (engl. simultaneous localization and mapping – SLAM) gilt dabei als eine besondere Herausforderung, da die aus den fehlerbehafteten Sensordaten resultierenden Unsicherheiten in der Lokalisierung und der erzeugten Karte voneinander abhängig sind.

In der vorliegenden Arbeit werden probabilistische Methoden für die Kombination mehrerer Sensoren mittels Multi-Sensor-Fusion für die robuste und präzise Lokalisierung und Kartenerstellung in heterogenen Außenumgebungen vorgestellt. Es werden sowohl Aspekte der robusteren Wiedererkennung von Landmarken als auch die Integration zusätzlicher absoluter und relativer Sensoren mittels erweiterter Filterverfahren beleuchtet. Die entwickelten Verfahren erlauben dabei eine Verbesserung der erzielbaren Präzision und Zuverlässigkeit für einen möglichst robusten Einsatz in urbanen als auch in unstrukturierten Bereichen.

ISSN 1863-6489
ISBN 978-3-7315-1074-1

