

Datengetriebene Methoden zur Bestimmung von Position und Orientierung in funk- und trägheitsbasierter Koppelnavigation

**Data-driven methods for determining position and
orientation in radio- and inertial-based dead reckoning
systems**

**Der Technischen Fakultät
der Friedrich-Alexander-Universität
Erlangen-Nürnberg
zur
Erlangung des Doktorgrades Dr.-Ing.**

**vorgelegt von
Tobias Feigl, M.Sc.
aus Nürnberg**

**Als Dissertation genehmigt
von der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg.**

Tag der mündlichen Prüfung: 29.09.2021

**Vorsitzender des
Promotionsorgans:** Prof. Dr.-Ing. habil. Andreas Paul Fröba

Gutachter: Prof. Dr. Michael Philippsen
PD Dr.-Ing. habil. Thomas Wittenberg

*Für meinen **Sohn**, der mir gezeigt hat, was bedingungslose Liebe ist,
für meinen vorzeitig verstorbenen **Vater**, dem ich die Motivation für
diese Arbeit verdanke und für meine **Mutter**, der ich mein Leben
verdanke.*

Danksagung

„Abschied nehmen ist immer schwierig“. Die letzten Jahre waren für mich eine wilde Reise durch eine dunkle Zeit. Diese Arbeit gab mir Stabilität und Richtung, um diese Zeit hinter mir zu lassen und stärker und klüger zu werden. Nachdem ich gleichzeitig meine eigene junge Familie und meinen Vater verlor, begann ich als naiver neugieriger Student mit dem Selbststudium verschiedener Forschungsbereiche, das in dieser Arbeit mündet. Der folgende Dank geht daher an alle Menschen, die ich in den letzten Jahren getroffen habe und die mir für eine kürzere oder eine längere Zeit aus der Dunkelheit geholfen haben: „Danke, es ist schön zu fühlen, dass ich nie allein bin!“ und „Danke für dein Lächeln, deine Geduld mit mir und deinen Rat.“ Es gibt kein Wort, dass meine Dankbarkeit ausdrücken kann, und es ist unmöglich, allen zu danken, die auf die eine oder andere Weise dazu beigetragen haben, mich zum Abschluss dieser Arbeit zu begleiten.

Diese Arbeit war nur dank der großen Unterstützung vieler Menschen möglich, sowohl in meinem beruflichen als auch in meinem privaten Umfeld. Ich danke jedem Einzelnen von euch für eure Unterstützung im Laufe der letzten Jahre. Meine wissenschaftliche Reise begann mit meinem Vorgesetzten Christopher Mutschler, der mit mir das Thema meiner Masterarbeit ausarbeitete und damit den Grundstein für diese Dissertation legte. Lieber Chris: „Vielen Dank für die Gelegenheit, an all den spannenden Projekten der letzten drei Jahre zu arbeiten, für unsere Diskussionen zu jeder Tages- und Nachtzeit und für all deine Hilfe, deinen väterlichen Rat und deine Freundschaft.“

In diesem Zusammenhang möchte ich natürlich meinem Doktorvater Prof. Dr. Michael Philippse für seine großartige Unterstützung in den letzten Jahren danken. Michael hat diese Arbeit und mich betreut und maßgeblich zur Relevanz des Inhalts dieser Arbeit und ihrer Schärfe bei der Vorbereitung beigetragen. Er brachte mir wissenschaftliches Schreiben und Präsentationsfähigkeiten bei und strukturierte meine Gedanken und mein Verhalten. Ohne sein Fachwissen, seinen väterlichen Rat und seine ermutigenden Worte wäre ich nie in der Lage gewesen, diese Arbeit abzuschließen.

In diesem Zusammenhang möchte ich meinen Gutachtern Prof. Dr. Michael Philippse und PD Dr.-Ing. habil. Thomas Wittenberg im Vor- aus für ihre Bemühungen bei der Begutachtung dieser Arbeit danken und Sie bitten, mir etwaige Unannehmlichkeiten zu verzeihen.

Diese Arbeit wurde in Zusammenarbeit zwischen dem Fraunhofer-Institut für Integrierte Schaltungen (IIS) und dem Lehrstuhl für Programmiersysteme (Informatik 2) der Friedrich-Alexander-Universität (FAU) Erlangen-Nürnberg erstellt. Ich möchte mich daher bei den Leistungsträgern beider Institute bedanken, die alle technischen und wirtschaftlichen Herausforderungen gelöst haben, damit ich diese Dissertation abschließen kann. Ich möchte allen Kollegen des IIS danken, die mich begleitet haben. Ich habe meine Zeit mit euch wirklich genossen. Der Zugang zum Forschungslabor und die einzigartige Messtechnik des IIS, der Pool von Testpersonen und das Expertenwissen waren wichtige Puzzleteile meiner Arbeit. Ich möchte natürlich auch allen Kollegen der FAU danken. Ich werde die wissenschaftlichen Diskussionen, privaten Gespräche, den Kuchen am Donnerstag und die spaßige Umgebung vermissen. Die soziale und berufliche Atmosphäre in beiden Instituten hat mich immer stabilisiert und motiviert. Ich habe jede Minute des Forschungscampus in Nürnberg und Erlangen genossen und war stolz darauf, Kleidung mit den IIS und FAU Logos zu tragen. Insgesamt haben die Fraunhofer Gesellschaft und die FAU eine freundliche und lebendige Atmosphäre geschaffen, in der es eine Freude war, zu arbeiten.

Viele Menschen haben mir geholfen, die in dieser Arbeit vorgestellten Forschungsergebnisse und diese Dissertation zu vervollständigen. Es war mir eine Ehre (in alphabetischer Reihenfolge) mit Ernst Eberlein, Dr.-Ing. Thorsten Edelhäuser, Prof. Dr. Björn Eskofier, Dr.-Ing. Stefan Gradl, Eliise Kõre, Sebastian Kram, Prof. Dr. Marc Erich Latoschik, Gruner Lisa, Christoffer Löffler, Dr.-Ing. Christopher Mutschler, Thorsten Nowak, Prof. Dr. Michael Philippse, Andreas Porada, Anes Redzepagic, Dr. Daniel Roth, Jochen Seitz, Ramiz Hassan Siddiqui, Steiner Steve, Prof. Dr.-Ing. Jörn Thielecke, Markus Wirth und Philipp Wolter erfolgreich gemeinsame Forschungsarbeiten zu veröffentlichen. Ein besonderer Dank geht an Ernst und die beiden Thorstens, die mich die komplexe Welt der Funklokalisierung lehrten, und an Daniel, der mich persönlich und wissenschaftlich immer aktiv beriet, ob bei einem wissenschaftlichen Ausflug, bei einem spontanen Bier am Bahnhof oder gemeinsamer wissenschaftlicher Ausarbeitung bis spät in die Nacht und Sebastian und Christoffer, die mir auch immer mit Rat und Tat zur Seite standen. Euer vielfältiges Wissen hat dazu beigetragen, meine Ausbildung zu erweitern, und euer Feedback hat dazu beigetragen, die Richtung meiner Forschung zu bestimmen. Insbesondere die akzeptierten (gemeinsamen) Papiere [131, 133, 128, 129, 127, 124, 125, 126, 122, 134] boten herausragende Lösungen für die datengetriebene Fußgängerkoppelna-

vigation in VR- und Sportanwendungen. Ich möchte mich auch bei den vielen Menschen hinter dem Vorhang bedanken, die mit ihrer Hilfe und Unterstützung bei der Entwicklung von Softwarekomponenten, bei der Durchführung von Studien oder Diskussionen halfen (in alphabetischer Reihenfolge): Andreas Eidloth, Robert Frank, Norbert Franke, Georgios Kontes, Arne Niitsoo, Tim Nisslbeck, Stephan Otto, Maximilian Stahlke, Nicolas Witt und allen Studienteilnehmern.

An dieser Stelle möchte ich mich auch bei allen Studierenden bedanken, die ihre Qualifizierungsarbeit unter meiner Aufsicht abgeschlossen haben und deren Beiträge die Entwicklung dieser Dissertation unterstützten (in alphabetischer Reihenfolge): Thomas Altstidl [16], Jan Niklas Bauer [31], Peter Bauer [32], Frank Brendel [53], Christian Daxer [96], Felix Gruber [157], Lisa Gruner [159], Oskar Herrmann [183], Christian Jakob [207], Stephanie Mehltretter [296], Felix Ott [322], Lea Otte [324], Andreas Porada [341, 342] und Lukas Schmidt [391].

Ich möchte auch die Gelegenheit nutzen, meiner Mutter Regina für ihre ständige Unterstützung, Hilfe, Freundlichkeit und Ermutigung zu danken, obwohl sie selbst Unterstützung brauchte. Abschließend möchte ich meinen Lebensgefährten und Freunden danken. Ohne euch wäre ich längst „*aus dem Fenster gesprungen*“. Vielen Dank für die Feierlichkeiten, die meinen Horizont erweiterten, die erholsamen Tage am Strand und die vielen sportlichen Aktivitäten und Wettbewerbe. Besonderer Dank geht an Melanie, Marlen, Jeanette, Max und Daniel für ihre Liebe und Unterstützung. Ohne all diese Unterstützung wäre diese Arbeit nicht das gewesen, was sie war und jetzt ist. Vielen Dank, dass ihr für mich da wart, als es dringend notwendig war.

Die in dieser Arbeit vorgestellten Arbeiten wurden vom Bayerischen Ministerium für Wirtschaft, Infrastruktur, Verkehr und Technologie und dem Europäischen Fonds für regionale Entwicklung sowie vom interdisziplinären Zentrum für eingebettete Systeme unterstützt.

Tobias Feigl

Originalitätserklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Gerhardshofen, den 12. Oktober 2021

Tobias Feigl

Kurzzusammenfassung

Standortbezogene Unterhaltung ist mittlerweile zu einem Grundbedürfnis geworden. Die erforderliche Genauigkeit und Zuverlässigkeit von Lokalisierungssystemen wächst nicht nur für intelligente Systeme wie selbstfahrende Fahrzeuge, Lieferdrohnen und mobile Geräte, sondern auch für alltägliche Fußgänger. Aufgrund der allgegenwärtigen Sensoren wie Kameras, GPS und Trägheitssensoren werden mit aufwendig handgefertigten Modellen und Algorithmen eine Vielzahl von Lokalisierungssystemen entwickelt. Um eine Einschränkung der freien Sicht und unterschiedliche Lichtverhältnisse von Kamerasystemen zu vermeiden werden typischerweise Funk- und Trägheitssensoren zur Lokalisierung verwendet. Unter idealen Laborbedingungen können diese Sensoren und Modelle, Positionen und Orientierungen langfristig genau abschätzen. In realen Umgebungen wirken sich jedoch viele Probleme wie ungenaue Systemmodellierung, unvollständige Sensormessungen, Rauschen und komplexe Umgebungsdimensionen auf die Genauigkeit und Zuverlässigkeit aus. Individuell betrachtet haben Funk- und Trägheitssensoren Schwierigkeiten: Funk lokalisiert aufgrund mehrerer Pfade durch statische oder dynamische Objekte entlang der Ausbreitungspfade zwischen Sender und Empfänger sehr ungenau. Im Gegensatz dazu akkumulieren Trägheitssensoren im Laufe der Zeit Entfernungs- und Orientierungsfehler und können keinen absoluten Bezug zur Weltkarte herstellen. Verfahren des Stands der Technik ergänzen beide Sensoren, um komplementäre Effekte zu verwenden, können jedoch die Schwierigkeiten nicht beheben. Darüber hinaus können sie mit einfachen Bewegungsmodellen wie konstanter Beschleunigung oder Geschwindigkeit keine stark nichtlinearen menschlichen Bewegungen beschreiben.

Das Hauptziel dieser Arbeit ist es daher, die Auswirkungen datengetriebener Methoden und verschiedener Sensordatenströme von lose platzierten Sensoren auf die Genauigkeit der Schätzung menschlicher Posen in hochdynamischen Situationen zu untersuchen. Die absolute Genauigkeit der erhaltenen Ergebnisse wird mit Filtermethoden nach dem Stand der Technik verglichen. Um die Probleme von Menschen entworfenen Lokalisierungsmodellen zu lösen, werden in dieser Arbeit maschinelle und tiefe Lernmethoden verwendet. Es werden Lernmethoden zur Positions-, Geschwindigkeits- und Orientierungsschätzung sowie zur Rekonstruktion der Trajektorie unter Verwendung mul-

timodaler Messungen von Funk- und Trägheitssensoren vorgestellt, um eine genaue und robuste Lokalisierung zu erreichen. Die Auswirkungen datengetriebener Verfahren entlang einer typischen Verarbeitungskette für die Lokalisierung mit Funk- und Trägheitssensoren werden untersucht. Die Verarbeitungskette ist lose gekoppelt in atomare Komponenten unterteilt, sodass jedes datengetriebene Verfahren problemlos ausgetauscht werden kann. Sequenzbasierte Lernmethoden werden entlang der Verarbeitungskette verwendet, um absolute Positionen aus Ankunftszeitstempeln von Funksignalen mit Mehrwegeausbreitung zu schätzen, ungerichtete Geschwindigkeitsvektoren von Trägheitssensoren zu schätzen, Bewegungsmuster zu klassifizieren, die die Ausrichtung der Trajektorie kalibrieren und um schließlich die einzelnen Komponenten zu einer Trajektorie zu fusionieren. Die vorgeschlagenen Methoden lernen, mit unterschiedlichem Bewegungsverhalten umzugehen und ermöglichen eine robuste und präzise Lokalisierung. Im Rahmen von Großstudien werden Mess- und Referenzdaten mit verschiedenen Bewegungsformen bei unterschiedlichen Geschwindigkeiten erfasst. Umfangreiche Experimente zeigen die Wirksamkeit und das Potenzial der vorgeschlagenen Methoden. Die datengetriebene, modulare Verarbeitungskette liefert genauere und robustere Schätzungen als bekannte Verfahren, auch bei dynamischen Bewegungen mit verrauschten Trägheitssensoren und Funkumgebungen mit Mehrwegeausbreitung.

Abstract

Location-based entertainment has now become a basic need. The required accuracy and reliability of localization systems is growing not only for intelligent systems such as self-driving vehicles, delivery drones, and mobile devices, but also for everyday pedestrians. Due to the ubiquitous sensors such as cameras, GPS, and inertial sensors, a large number of localization systems are developed with elaborately hand-crafted models and algorithms. To avoid problems with the occlusion and different light conditions of camera systems, radio and inertial sensors are typically used for localization. Under ideal laboratory conditions, these sensors and models can accurately estimate positions and orientations over the long term. However, in real-world environments, many problems such as inaccurate system modeling, incomplete sensor measurements, noise, and complex environmental dynamics affect accuracy and reliability. Viewed individually, radio and inertial sensors have difficulties: radio localises very imprecisely due to multipaths through static or dynamic objects along the propagation paths between transmitter and receiver. In contrast, inertial sensors accumulate distance and orientation errors over time and cannot establish an absolute reference to the world coordinate system. State-of-the-art methods complement both sensors to use complementary effects, but fail to remedy the difficulties. In addition, they cannot describe highly non-linear human movements using simple motion models such as constant acceleration or velocity.

The main goal of this thesis is therefore to investigate the effects of data-driven methods and different sensor data streams from loosely placed sensors on the accuracy of the estimation of human poses in highly dynamic situations. The absolute accuracy of the results obtained is compared with state-of-the-art filtering methods. To solve the problems of human-designed localization models, machine and deep learning methods are used in this thesis. Thus, methods that learn to estimate positions, velocities, and orientations as well as to reconstruct trajectories using multimodal measurements from radio and inertial sensors are presented to achieve an accurate and robust localization. The effects of data-driven processes along a typical processing chain for localization with radio and inertial sensors are examined. The processing chain is loosely coupled and divided into atomic components so that any data-driven process can easily be replaced. Sequence-based

learning methods are used along the processing chain to estimate absolute positions from time of arrival values of radio signals with multipath propagation, to estimate undirected velocity vectors from inertial sensors, to classify movement patterns that calibrate the alignment of the trajectory, and finally to merge the individual components into one trajectory. The proposed methods learn to deal with different movement behavior and enable a robust and precise localization. As part of large-scale studies, measurement and reference data with various forms of movement at different velocities are recorded. Extensive experiments show the effectiveness and potential of the proposed methods. The data-driven, modular processing chain provides more accurate and more robust estimates than state-of-the-art methods, even for dynamic movements, with noisy inertia sensors, and radio environments with multipath propagation.

Abbildungsverzeichnis

1.1	Verarbeitungskette der vorgeschlagenen Methode und Aufbau der Dissertation.	23
2.1	Verarbeitungskette der visuellen Trägheitsdometrie.	32
2.2	Verarbeitungskette der Fingerabdruck-basierten Lokalisierung.	35
2.3	Fusion von Funk- und Trägheitssensoren.	41
2.4	Positionsbestimmung mit Trilateration und Ankunftszeiten.	46
2.5	Positionsgenauigkeit von modell- und datengetriebenen Funklokalisierungssystemen.	49
2.6	Enge und lose Fusion von Sensoren.	55
2.7	Verarbeitungskette eines Kalman-Filters.	62
3.1	Kamerabasiertes Lokalisierungssystems.	76
3.2	Kartesisches- und polares Koordinatensystem.	77
3.3	Integrationsfehler der Positionsrechnung.	79
3.4	Integrationsfehler der Orientierungsrechnung.	80
3.5	Beispielhafte Darstellung der Koppelnavigation.	81
3.6	Antennenplatzierung eines Funklokalisierungssystems. .	83
3.7	Mehrwege-Funkausbreitungsszenarien.	84
3.8	Verarbeitungskette der Funklokalisierung.	86
3.9	Kanalimpulsantworten und Korrelatorausgabe.	88
3.10	Signalverarbeitungskette einer Zeitreihenanalyse.	96
3.11	Gleitendes Mittelwert Filter.	98
3.12	Exponentialfilter.	99
3.13	Butterworth-Tiefpassfilter.	100
3.14	$tanh(x)$ -Aktivierungsfunktion und Gradient.	101
3.15	Detaillierte Verarbeitungskette eines Kalman-Filters. .	103
3.16	Neuronales Netz.	107
3.17	Neuron eines neuronalen Netzes.	108
3.18	Vorwärtsgekoppeltes neuronales Netz.	109
3.19	Eine zweidimensionale Faltung.	112
3.20	Eine zweidimensionale Standardfaltung.	114
3.21	Maximale- und Durchschnitts-Vereinigung.	115
3.22	Rekurrentes neuronales Netz.	126

Abbildungsverzeichnis

3.23	Abgerolltes rekurrentes neuronales Netz.	127
3.24	Gradientenfluss durch ein rekurrentes neuronales Netz. .	128
3.25	Elman-Zelle.	131
3.26	Lang-Kurzzeitgedächtniszelle.	132
3.27	Gatter der Lang-Kurzzeitgedächtniszelle.	133
3.28	Vergessensgatter der Lang-Kurzzeitgedächtniszelle. . . .	134
3.29	Eingangs- und Aktualisierungsgatter der Lang-Kurzzeitgedächtniszelle.	136
3.30	Ausgabegatter der Lang-Kurzzeitgedächtniszelle.	137
3.31	Gegatterte rekurrente Zelle.	138
3.32	Rücksetzungsgatter einer gegatterten rekurrenten Zelle. . . .	139
3.33	Verborgenes Zustandsgatter einer gegatterten rekurrenten Zelle.	140
3.34	Aktualisierungsgatter einer gegatterten rekurrenten Zelle.	141
3.35	Gestapelte RNN-Architektur.	142
3.36	Dichte RNN-Architektur.	144
3.37	Bidirektionale RNN-Architektur.	144
4.1	Einordnung der Positionsschätzung.	147
4.2	Verschiedene Kalman-Filttereinstellungen.	153
4.3	LSTM Zelle.	161
4.4	Viele-zu-Viele LSTM-Architektur.	163
4.5	Datenvorbereitungsprozess der Positionsrechnung. . . .	165
4.6	Vollständig verbundene Schicht der Positionsschätzung. .	166
4.7	Funksignaldrift in einer Mehrwegeumgebung.	167
4.8	Signalverarbeitungskette eines Funklokalisierungssystems.	170
4.9	Plattformen zur Datenakquise.	172
4.10	Datensätze zur Positionsschätzung.	173
4.11	Reale Umgebungen der Positionsschätzung.	176
4.12	Ergebnisse der Positionsschätzung mit synthetischen Datensätzen.	179
4.13	Ergebnisse der Positionsschätzung mit zufälligen Bewegungsformen.	181
4.14	Ergebnisse der Positionsschätzung mit Mäander-Bewegungsformen.	183
4.15	Ergebnisse der Positionsschätzung mit Zickzack-Bewegungsformen.	185
4.16	Ergebnisse der Positionsschätzung mit Spaziergänger- und Gitter-Bewegungsformen.	186

4.17	Ergebnisse der Positionsschätzung mit Mehrwegeausbreitung	188
4.18	Anwendung der datengetriebenen Positionsschätzung unter Mehrwegeausbreitung	191
5.1	Einordnung der Geschwindigkeitsschätzung	193
5.2	PDR Verarbeitungskette	196
5.3	SINS Verarbeitungskette	198
5.4	Verarbeitungskette der Geschwindigkeitsschätzung	213
5.5	Vorverarbeitungskette der Geschwindigkeitsschätzung	214
5.6	<i>SMV(acc)</i> Ausschnitt einer Gehaktivität	215
5.7	Architektur der ressourcenschonenden Geschwindigkeitsschätzung	223
5.8	Architektur von Resteinheiten	229
5.9	Architektur der hochgenauen Geschwindigkeitsschätzung .	230
5.10	Zeitlicher Merkmalsverfolger der Geschwindigkeitsschätzung	232
5.11	Hardwarekonfiguration der Geschwindigkeitsschätzung .	236
5.12	Referenztrajektorien und -geschwindigkeiten eines Probanden	240
5.13	Datenmenge der Geschwindigkeitsschätzung	241
5.14	<i>acc</i> -Signale und Referenzgeschwindigkeit der Geschwindigkeitsschätzung	242
5.15	CDFs der Geschwindigkeitsschätzung auf den Testdatensätze V1-V3	259
5.16	PDFs der Geschwindigkeitsschätzung auf den Testdatensätze V1-V3	262
5.17	DETs der Geschwindigkeitsschätzung auf den Testdatensätze V1-V3	264
5.18	CDF, PDF, DET der Generalisierbarkeitsauswertungen der Geschwindigkeitsschätzung	275
6.1	Einordnung der Orientierungsschätzung	287
6.2	Auswirkung fehlerhafter Orientierungsschätzung	290
6.3	Sensor drift und ihre Auswirkung auf die Orientierungsschätzung	293
6.4	Hardwareaufbau der Orientierungsschätzung	297
6.5	Rotationstisch zur Analyse von Orientierungsfehlern	298
6.6	VR-Szene der Studie der Orientierungsfehler	300
6.7	Verarbeitungskette der Orientierungsschätzung	307

Abbildungsverzeichnis

6.8	<i>acc^{raw}</i> einer Gehbewegung zur Orientierungsschätzung.	308
6.9	Sensorausrichtung der Orientierungsschätzung.	309
6.10	Gefilterte Sensordatenströme der Orientierungsschätzung.	311
6.11	VR-Szenarios zur Datenakquise der Orientierungsschätzung.	319
6.12	Ich-Ansicht der VR-Darstellung zur Datenakquise der Orientierungsschätzung.	320
6.13	VR-Szenario des Bewertungsexperiments der Orientierungsschätzung.	329
6.14	Kalibrierung des Gier-Winkels in einer typischen VR-Anwendung.	332
7.1	Einordnung der Posenschätzung.	337
7.2	Verarbeitungskette der Posenschätzung.	352
7.3	Architektur der Posenschätzung.	355
7.4	Eingabeformat der Posenschätzung.	356
7.5	Referenztrajektorien der Posenschätzung.	357
7.6	Rekonstruierte (Gehen) Trajektorien der modellgetriebenen Posenschätzung.	366
7.7	Rekonstruierte (Joggen) Trajektorien der modellgetriebenen Posenschätzung.	368
7.8	Rekonstruierte (Rennen) Trajektorien der modellgetriebenen Posenschätzung.	370
7.9	Rekonstruierte (Zufällig) Trajektorien der modellgetriebenen Posenschätzung.	372
7.10	Ergebnisse (variierende Eingabedatenströme) der datengetriebenen Posenschätzung.	379
7.11	Ergebnisse (variierende Prognosehorizont) der datengetriebenen Posenschätzung.	381
7.12	Ergebnisse (variierende Sequenzlänge) der datengetriebenen Posenschätzung.	384
7.13	Ergebnisse (dynamische Bewegung) der datengetriebenen Posenschätzung.	387
7.14	Rekonstruierte Trajektorien (dynamische Bewegung) der datengetriebenen Posenschätzung.	389
7.15	Generalisierbarkeit der datengetriebenen Posenschätzung.	391
8.1	Anwendung der datengetriebenen Zeitreihenanalyse zur Gangrekonstruktion in VR.	440

8.2	Anwendung der datengetriebenen Zeitreihenanalyse zur Klassifikation von Simulatorkrankheit in VR.	441
8.3	Anwendung der datengetriebenen Posenschätzung in einem VR Museum.	443
8.4	Zukünftige Verarbeitungskette der Funklokalisierung. . .	448

Tabellenverzeichnis

0.1	Deutsche Begriffe, deren Übersetzung ins Englische und (optional) die verwendete Abkürzung.	xxix
2.1	Ergebnisse verwandter Arbeiten zur Fusion von IMUs und verschiedenen Sensortypen.	37
2.2	Ergebnisse verwandter Arbeiten zur Fusion von IMUs und Funksystemen.	52
2.3	Ergebnisse verwandter Arbeiten zur Fusion von IMUs und Funksignalen mit speziellen Methoden.	59
2.4	Ergebnisse verwandter Arbeiten zur Fusion von IMUs und Funksystemen mit Kalman-Filter Methoden. . . .	67
2.5	Ergebnisse verwandter Arbeiten zur Fusion von IMUs und Funksystemen mit Partikel-Filter Methoden. . . .	69
2.6	Ergebnisse verwandter Arbeiten zur Fusion von IMUs und Funksystemen mit datengetriebenen Methoden. . .	73
3.1	Allgemeine Aktivierungsfunktionen.	112
4.1	Datensätze zur Positionsschätzung.	174
4.2	Ergebnisse der Rastersuche der Positionsschätzung. . .	177
4.3	Ergebnisse der Positionsschätzung in der realen Umge- bung.	185
5.1	Statistik des Datensatzes der Geschwindigkeitsschätzung.	255
5.2	Genauigkeit der Geschwindigkeitsschätzung der Daten- sätze V ₁ - V ₃	260
5.3	Ergebnisse der Geschwindigkeitsschätzung auf den Da- ten der ausgelassenen Probanden.	271
5.4	Rechenaufwand der Geschwindigkeitsschätzung. . . .	272
5.5	Ergebnisse der Generalisierbarkeit der Geschwindig- keitsschätzung.	276
6.1	Ergebnisse der Analyse der Orientierungsfehler. . . .	301
6.2	Klassifikationsergebnisse der Orientierungsschätzung. .	322
6.3	Erfolgsraten der Orientierungsschätzung.	325
7.1	Statistik des Datensatzes zur Posenschätzung.	363

Tabellenverzeichnis

7.2	Ergebnisse der modellgetriebenen PDR Posenschätzung.	365
7.3	Ergebnisse (variierende Eingabedatenströme) der daten- getriebenen Posenschätzung.	377
7.4	Ergebnisse (variierende Prognosehorizont) der datenge- triebenen Posenschätzung.	380
7.5	Ergebnisse (dynamische Bewegung) der datengetriebe- nen Posenschätzung.	388

Abkürzungsverzeichnis

Symbole		<i>g</i>	Gravitationskonstante (in Deutschland: $g=9.81 \text{ m/s}^2$)
# Anzahl		GB	Gigabyte
A		Geschw.	Geschwindigkeit
Abb. Abbildung		ggf.	gegebenenfalls
<i>acc.</i> Beschleunigungsdatenstrom		GHz	Gigahertz
<i>Acc.</i> Beschleunigungssensor		<i>gyr.</i>	Drehratendatenstrom
Alg. Algorithmus		<i>Gyr.</i>	Gyroskop oder Drehratensensor
B		H	
bspw. beispielsweise		<i>h</i>	Stunde
bzgl. in Bezug auf, bezüglich		<i>h</i>	Höhe (engl. height)
bzw. beziehungsweise		Hz	Hertz, Einheit der Frequenz, die die Anzahl der sich wiederholenden Vorgänge pro Sekunde in einem periodischen Signal angibt.
C		I	
cm Zentimeter		i.d.R.	in der Regel
Conf. Conference (dt. Konferenz)		i.e.	lat. id est (dt. d.h.)
Corr Correlation (dt. Korrelation)		Intl.	International
CPU Central Processing Unit (dt. zentrale Recheneinheit)		K	
D		kalib.	Kalibriert/e
d.h. das heißt		km	Kilometer
dB Dezibel		km/s	Kilometer pro Sekunde
dBm Leistungspegel ausgedrückt in Dezibel auf ein Kilowatt		L	
dps degree per second (dt. Grad pro Sekunde)		L.I.N.K.	Lokalisierung, Identifikation, Navigation und Kommunikation
dt. Deutsch		M	
E		m	Meter
e.g. lat. exempli gratia (dt. z.B.)		m.a.W.	mit anderen Worten
engl. Englisch		m/km	Meter pro Kilometer
et al. lat. et alii (dt. und andere, u.a.)		m/s	Meter pro Sekunde, Einheit der Geschwindigkeit
F		m/s ²	Meter pro Sekunde im Quadrat, Einheit der Beschleunigung
FLOP Floating Point Operation Per Second (dt. Gleitkommaoperationen pro Sekunde)		<i>mag.</i>	Magnetfelddatenstrom
frame Einzelbild in Animationen und Computerspielen		<i>Mag.</i>	Magnetfeldsensor
		<i>max</i>	obere festgelegte Intervallgrenze
		Meas.	Messung
		MHz	Megahertz
		<i>min</i>	Minimum, minimal oder untere festgelegte Intervallgrenze
G		min	Minute
		mm	Millimeter

Abkürzungsverzeichnis

ms	Millisekunden	T	
N		TB	Terrabyte (10^{12})
		TFLOP	Terra FLOPS, 10^{12} Gleitkommaoperationen pro Sekunde
ns	Nanosekunden	Trans.	Transaction
		Tx	Transmitter (dt. Sender)
P		U	
PFLOP	Peta FLOPS, 10^{15} Gleitkommaoperationen pro Sekunde	u.a.	und andere/n oder unter anderen/m
Proc.	Proceeding/s	μs	Mikrosekunden
ps	Pikosekunden	V	
R		v.A.	vor allem
Ref.	Referenz	Valid.	Validation
resp.	respektive		
Rx	Receiver (dt. Empfänger)	W	
S		w w.r.t.	Weite, bspw. eines Signalfensters with respect to (dt. bzgl.)
s	Sekunden	Z	
S.	Seite		
Schätz.	Schätzung		
Symp.	Symposium	z.B.	zum Beispiel

Akronymverzeichnis

Symbol			
1D	Eindimensional	CIR	Channel Impulse Response (dt. Kanalimpulsantwort)
2D	Zweidimensional	CKF	Kubatur Kalman-Filter
3D	Dreidimensional	CNN	Faltendes neuronales Netz
5G	Fünfte Generation des Mobilfunks	CP	Channel Parameter (dt. Funkkanalparameter)
6D	Sechsdimensional	CPU	Zentrale Rechen" und Steuereinheit eines Computers
A		CR	Channel Response
		CSI	Channel State Information (dt. Kanalzustandsinformation)
ANN	Artificial Neural Network (dt. künstliches neuronales Netz)	D	
AoA	Angle of Arrival (dt. Ankunftsinkel)	DAE	Denoising Autoencoder (dt. entrauschender Autoenkdierer)
API	Application Programming Interface (dt. Programmierschnittstelle)	DEPM	Distance Error Per Meter (dt. Entfernungsfehler pro Meter)
AR	Augmented Reality (dt. augmentierte Realität)	DET	Distance Error Total (dt. Entfernungsfelhersumme)
ARCore	Augmented Reality-Programmierschnittstelle für Android-Geräte	DL	Deep Learning (dt. tiefes Lernen)
ARKit	Augmented Reality-Programmierschnittstelle für Apple-Geräte	DNN	Deep Neural Network (tiefes neuronales Netz)
AWGN	Additive White Gaussian Noise (dt. additives weißes Gaußsches Rauschen)	DoF	Degree of Freedom (Anzahl der Freiheitsgrade)
B		DR	Dead Reckoning (dt. Koppelnavigation)
BC	Bancroft	DS	Delay Spread (dt. Verzögerungsspreizung)
BeiDou	Chinesisches Satellitennavigationsystem (dt. Großer Bär)	DT	Decision Tree (dt. Entscheidungsbaum)
BERT	Bidirectional Encoder Representations from Transformers	E	
BLE	Bluetooth Low Energy	EKF	Extended Kalman-Filter (dt. erweitertes Kalman-Filter)
BLSTM	Bidirectional LSTM	ESKF	Error-State Kalman-Filter (dt. Fehlerzustands-Kalman-Filter)
BN	Batch Normalization (dt. Stapelnormalisierung)	ESPRIT	Estimation of Signal Parameters via Rotational Invariance Technique (dt. Schätzung von Signalparametern mittels Rotationsinvarianztechnik)
BP	Backpropagation	F	
BPTT	Backpropagation Through Time	F₁	F-Maß=1.0 bestätigt perfekte Präzisions" und Rückruf-Werte einer Klassifikation
BRNN	Bidirectional Recurrent Neural Network	FDPoA	First Direct Path of Arrival (dt. erster direkter Ankunftsypfad)
C		FF	Feedforward (dt. Vorwärtskopplung)
C/RNN	Convolutional / Recurrent Neural Network		
CART	Classification and Regression Trees		
CDF	Cumulative Distribution Function		
CE	Channel Estimation (dt. Kanalschätzung)		
CEP	Circular Error Probability		

Akronymverzeichnis

FFNN	Feedforward Neural Network (dt. Vorwärtsgekoppeltes neuronales Netz)	INS	Inertial Navigation System (dt. Trägheitsnavigationssystem)
FFT	Fast Fourier Transformation (dt. schnelle Fourier-Transformation)	IQR	Interquartilsabstand
FIR	Finite Impulse Response (dt. Filter mit einer endlichen Impulsantwort)	IRNN	Identitäts rekurrentes neuronales Netz
FPGA	Field Programmable Gate Array (dt. programmierbare Logikgatter-Anordnung)	ISAM	Incremental Smoothing and Mapping (dt. inkrementelle Glättung und Zuordnung)
FRFNN	Feedforward Recurrent Feedforward Neural Network	J	
G		JVM	Java Virtual Machine
Galileo	Galileo Satellite Navigation System	K	
GD	Gradient Descent (dt. Gradientenabstieg)	k-NN	K-Nearest-Neighbor (dt. Nächste-Nachbarn-Klassifikation)
GLONASS	Globales Navigationssatellitensystem der Russischen Föderation	KF	Kalman-Filter
GNSS	Globales Navigationssatellitensystem	KF (Avg.)	Optimiertes KF mit starrem Bewegungsmodell
GP	Gaußprozess	kF	Ricean k-Faktor
GPS	Global Positioning System (dt. Globales Positionsbestimmungssystem)	KKF	Komplementäres Kalman-Filter
GPU	Graphics Processing Unit (dt. Grafikprozessor)	KNN	Künstliches neuronales Netz
GRU	Gated Recurrent Unit (dt. gegitterte rekurrente Zelle)	L	
GTM	Ground Truth Moment (dt. Referenzmoment)	LiDAR	Light detection and ranging (dt. optische Abstands- und Geschwindigkeitsmessung)
Gyr.	Gyroskop	LKF	Lineares Kalman-Filter
H		LM	Levenberg Marquardt
HF	Hochfrequenz (engl. radio frequency, RF)	LoS	Line of Sight (dt. Sichtverbindung)
HMD	Head-Mounted-Display (dt. Kopfmontierter Bildschirm)	LP	Low-pass Filter (dt. Tiefpassfilter)
HMM	Hidden Markov Model	LR	Learning Rate (dt. Lernrate)
HP	Hochpassfilter	LSP	Large Scale Parameter
HW	Hardware	LSTM	Long Short-Term Memory (dt. Lang-Kurzzeitgedächtnis)
I		LTE	Long-Term Evolution
IFP	Inflection Point Method (dt. Wendepunktmethode)	M	
iGPS	indoor GPS (dt. Globales Positionsbestimmungssystem für Innenräume)	M₅₂	Matern52 Kernel eines GP
IIR	Infinite Impulse Response (dt. Filter mit einer unendlichen Impulsantwort)	MAE	Mean Absolute Error (dt. Mittlerer absoluter Fehler)
IIS	Institut für Integrierte Schaltungen	MAG	Magnetometer
IMU	Inertial Measurement Unit (dt. Inertiale Messeinheit)	MCC	Maximum Correntropy Criterion (dt. Maximales Korrentropiekriterium)
		MCKF	Maximum Correntropy Kalman-Filter (dt. Maximales Korrentropie-Kalman-Filter)
		MEMS	Micro-Electro-Mechanical Systems (dt. Mikrosystem)
		ML	Machine Learning (dt. Maschinelles Lernen)
		ML-GP	Maschinelles Lernen mit Gaußprozessen

MLP	Multi Layer Perceptrons (dt. mehrschichtige Perzeptronen)	RAM	Random-Access Memory (dt. Speicher mit wahlfreiem/direktem Zugriff)
MPC	Multipath Components (dt. Mehrwegkomponenten)	RAN	Radio Access Network (dt. Funkzugangsnetz)
MSCKF	Multi-State Constraint Kalman-Filter (dt. Multi Zustand eingeschränktes Kalman-Filter)	RBF	Radial Basis Function
MSE	Mean Squared Error (dt. Mittlere quadratische Abweichung)	RedFIR	Innovative funkbasierte Ortstechnologie des Fraunhofer IIS
MTP	Motion To Photon (dt. durchschnittliche Bewegung zu Photon Latenz)	ReLU	Rectified Linear Unit (dt. Rectifier)
MUSIC	Multiple Signal Classification (dt. Mehrfachsignalklassifikation)	ResNet	Residual neural Network (dt. Residuale neuronales Netz)
N		RF	Radio Frequency (dt. Hochfrequenz)
NaN	Not-a-Number	RFID	Radio-Frequency Identification (dt. Identifizierung mit Hilfe elektromagnetischer Wellen)
NDK	Native Development Kit für Android	RGB	RGB-Farbraum der Grundfarben Rot, Grün und Blau
NLoS	Non-Line-Of-Sight (dt. Nicht-Sichtverbindung)	RGB-D	RGB-Farbraum der Grundfarben Rot, Grün und Blau mit Tiefeninformationen (engl. Depth)
NN	Neuronales Netz	RIEKF	Right Invariant Extended Kalman-Filter (dt. Rechtsinvariante erweiterte Kalman-Filter)
NP	No-Pose	RiMAX	Richter MAXimum likelihood parameter estimation framework (dt. Richter Schätzung der maximalen Wahrscheinlichkeitsparameter)
NR	New Radio	RMS-DS	Quadratisches Mittel der Verzögerungsspreizung
NTP	Network Time Protocol	RMSE	Root-Mean-Square Error (dt. Mittleres Abweichungsquadrat)
O		RNN	Recurrent Neural Network (dt. rekurrentes neuronales Netz)
OFDM	Orthogonal Frequency Division Multiplexing	RoNIN	Robust Neural Inertial Navigation
OLoS	Obstructed-Line-Of-Sight (dt. Gestörte-Sichtverbindung)	RQ	Rotational Quadratic
ORB	Oriented FAST and rotated BRIEF	RQ	Research Question
OvA	One-vs-All	RSS	Received Signal Strength (dt. Empfangsfeldstärke)
P		RSSI	Received Signal Strength Indication (dt. Indikator für die Empfangsfeldstärke)
PCA	Principal Component Analysis (dt. Hauptkomponentenanalyse)	RTLS	Real-Time Locating System (dt. System zur Echtzeit-Lokalisierung)
PDF	Probability Density Function (dt. Wahrscheinlichkeitsdichtefunktion)	RTT	Round Trip Time (dt. Paketumlaufzeit)
PDR	Pedestrian Dead Reckoning (dt. Fußgängerkoppelnavigation)	Rx	Receiver (dt. Empfänger)
PEAK	Verfahren zur Spitzenerkennung in Kanalimpulsantworten	S	
PF	Partikel-Filter	SAGE	Space Alternating General Expectation
Q		SD	Standard Deviation
QR	Quick Response	SEP	Spherical Error Probable
QuaDRiGa	QUAsi Deterministic Radio channel GenerAtor	SG	Savitzky-Golay Filter
R		SGD	Stochastischer Gradientenabstieg
radio	Funksystem, Funksignaldatenstrom	SH-KF	Sage Husk Kalman-Filter
		SINC	Sinus Cardinalis

Akronymverzeichnis

SINS	Strapdown Inertial Navigation System (dt. fahrzeugfestes Trägheitsnavigationssystem oder inertiales Navigationssystem)	UDP	User Datagram Protocol
SLAM	Simultaneous Localization and Mapping (dt. Simultane Positionsbestimmung und Kartierung)	UFIR	Unbiased Finite Impulse Response (dt. unverzerrtes Filter mit endlicher Impulsantwort)
SMA	Simple Moving Average (dt. gleitender Mittelwert)	UKF	Unscented Kalman-Filter (dt. Sigma-Punkt Kalman-Filter)
SMV	Signalgrößenvektor	USRKF	Unscented Square-Root Kalman-Filter (dt. Sigma-Punkt Quadratwurzel-Kalman-Filter)
SNR	Signal-to-Noise Ratio (dt. Signal-Rausch-Verhältnis)	UTC	Coordinated Universal Time (dt. koordinierte Weltzeit)
STFT	Short-Term Fourier-Transformation	UTDoA	Uplink Time Difference of Arrival
SVD	Singular Value Decomposition (dt. Singularwertzerlegung)	UWB	Ultra-wideband (dt. Ultrabreitband)
SVM	Support Vector Machine (dt. Stützvektormaschine)	V	
SVR	Support Vector Regression (dt. Stützvektor-Regression)	Vanilla	Standard RNN mit Elman-Zelle
T		V_i	Datensatz Variante i , mit $i \in [1..3]$
TanH	Tangens Hyperbolicus	VEP	Vertical Error Probable
TCN	Temporal Convolutional Networks (dt. zeitliches faltendes neuronales Netz)	VI-SLAM	Visual Inertial Simultaneous Localization and Mapping
TDNN	Time Delay Neural Networks (dt. zeitverzögertes neuronales Netz)	VINS	Visual Inertial Navigation System
TDoA	Time Difference of Arrival (dt. Laufzeitunterschied eines Ankunftszeitstempels)	VIO	Visual and Inertial Odometry
TFT	Temporal Feature Tracker (dt. zeitlicher Merkmalsverfolger)	VR	Virtuelle Realität
ToA	Time of Arrival (dt. Ankunftszeit)	W	
ToF	Time of Flight (dt. Signallaufzeit)	Wi-Fi	Wireless Fidelity (dt. kabellose Wiedergabekette)
Tx	Transmitter (dt. Sender)	WLAN	Wireless Local Area Network (dt. lokales Funknetz gemäß IEEE-802.11)
U		WSS	Wide Sense Stationary
		Z	
		ZARU	Zero Angular Rate Update
		ZUPT	Zero Velocity UPdaTe

Symbolverzeichnis

Symbole

.	Skalarprodukt, Multiplikation	b_{acc}	Fehler des Beschleunigungsmessers; zeitkorrelierte Größen, z. B. die deterministischen Rauschkomponenten eines Sensors
*	Faltungsoperator	b_{gyr}	Fehler des Gyroskops; zeitkorrelierte Größen, z. B. die deterministischen Rauschkomponenten eines Sensors
\odot	Elementweises Hadamard Produkt, z. B. in GRU, LSTM oder RNN Zellen	\mathcal{B}	Stapel (engl. batch)
\times	Elementweise Hadamard Multiplikation einer GRU, LSTM oder RNN Zelle	B_w	Bandbreite eines Funksignals
\times	Kreuzprodukt, Multiplikation oder Darstellung von Dimensionen	$B \cdot u$	Deterministische Anteil, ausgedrückt durch die wirkende Störung u und deren Dynamik B
+	Addition einer GRU, LSTM oder RNN Zelle		
A			
α	Komplexer Kanalkoeffizient der LSP Familie zur Beschreibung komplexer statistischer Kanalmodelle	c	Absolutes Korrelationssignal
α_i	Komplexes Gewicht eines Dirac Impulses, beschreibt den Einfluss von Pfadverlust und Materialwechselwirkungen für einen bestimmten Signalfeld	c	Kurzfristiges Rauschen der Geschwindigkeitsgeschätzung
a	Ausgabe der Aktivierungsfunktion eines Neuron	$c_{v,d}$	Kovarianz des Rauschen der Übergangsfunktion eines KF
a	Aktivität eines Modells zur Schrittlangenschätzung	$c_{v,m}$	Kovarianz des Messrauschens eines KF
a	Filterkoeffizienten eine Butterworth Filters in der Rückkopplung	C	Zellzustand, Kontextvektor oder Gedächtniszelle des LSTM
a	Funkkanalkoeffizienten	C_k	Faltungskoeffizienten eines SG Filter
A	Amplitude der (Ko)Sinus Funktionen	$corr_{yz}$	Korrelation zwischen der Y und Z Achse der Orientierungsschätzung
acc	Beschleunigungsmessung oder Beschleunigungsdatenstrom		
acc_n	Beschleunigungen	D	
acc_{grav}	Schwerkraft	d	Anzahl der Fenster der Trainingsdaten
acc_{grav}^{IIR}	Schwerkraft mit IIR gefiltert	d	Bündel aus ToF Werten (Wert pro Rx) der Funklokalisierung
acc_{in}^{IIR}	Lineare Beschleunigung	d	Distanz
acc_{in}^{IIR}	Lineare Beschleunigung mit IIR gefiltert	d_n	Versatz zwischen aktueller und letzter Position
acc_{SG}	Beschleunigung mit SG gefiltert	d_{ref}	Referenzdistanz
acc_{raw}	Ungefilterte Beschleunigung (enthält lineare Beschleunigung und Schwerkraft)	d'	Geschätzte Entfernung mit Zeitspanne
Acc	gleichbedeutend mit acc_{raw}	dt	D
	Beschleunigungsmesser(-sensor)		Nicht nullzentrierte Amplitude der (Ko)Sinus Funktionen
B			
b	Filterkoeffizienten eine Butterworth Filters in der Vorwärtsskopplung	D	Verwerfungsschicht (engl. dropout layer)
b	Körperrahmenorientierung	D_i	Datensatz Variante i
b	Stapelgröße des ML und DL	D_{rd}	Synthetischer Datensatz zur Funkposition Auswertung
b	Verzerrung (engl. bias)	dp	Verwerfungsrate
E			
		e_{acc}	Fehlerterm und Sensorrauschen, stochastische Größen ohne zeitliche Korrelation, z. B. die thermomechanischen Rauschkomponenten des Beschleunigungssensors

Symbolverzeichnis

e_{gyr}	Fehlerterm und Sensorrauschen, stochastische Größen ohne zeitliche Korrelation, z. B. die thermomechanischen Rauschkomponenten des Gyroskops	Gyr	Drehratenmesser (Sensor)
e_v	Fehler der Geschwindigkeit	H	
E	Zu minimierender Fehler der Verlustfunktion eines ML oder DL Modells	h	Verbogener Zustand eines NN oder einer LSTM Zelle
F		h	Kanalimpulsantwort (engl. CIR) der Funklokalisierung
f	Lösungsfunktion eines Modells	h	Körperhöhe der modellbasierten Schrittängenschätzung
f_i	Vergessensgatter einer LSTM Zelle im Zeitschritt i auch $f_t \dots$ im Zeitschritt t	$h(t)$	Kanal der Funklokalisierung
f_k	Filterkern einer Faltungsschicht	h_{Bw}	Bandbreitenbegrenzte CIR
f_q	Schrittängefrequenz eines modellbasierten Schrittängenschätzers	h_t	Verbogener Zustand im Zeitschritt t
f_s	Abtastrate eines Signals	\tilde{h}_t	Verbogenes Zustandsgatter einer GRU Zelle
$f_s(x)$	Räumliche Merkmale, auch nur f_s	H	Beobachtungsmatrix eines KF
\hat{f}_s	Interne Repräsentation der räumlichen Merkmale in einem Modell	H	Verbogene Schicht
f_{Smin}	Minimale Abtastfrequenz nach Nyquist-Shannon	H	Vereinfachte Darstellung einer RNN oder LSTM oder GRU Zelle
$f_t(x)$	Entstehung und Entwicklung der räumlichen Merkmale $f_s(x)$	H_b	Verbogene rückwärts LSTM Schicht
\hat{f}_t	Interne Repräsentation der zeitlichen Merkmale in einem Modell	H_f	Verbogene vorwärts LSTM Schicht
f_v	Filtervektor einer Faltungsschicht	I	
f_w	Merkmale eines Datensatzes für ML	i	Zeitpunkt oder Zeitschritt, mit δi der Länge eines Zeitschrittes
F	KF Übergangsfunktion beschreibe die Bewegungsdynamik (Übergänge konsekutiver Zustände), wie sich der Zustand während eines einzelnen Zeitschritts voraussichtlich ändern wird	i_i	Eingangsgatter einer LSTM Zelle
F	Anzahl der Filterkerne einer Faltungsschicht	I	Identitätsmatrix
F_s	Dimension eines Filterkerns $d \times N_w$	I_g	Gini-Diversity-Index
F_s	Fenster eines Datensatzes mit Dimensionen $d \times N_w$	J	
G		j_t	Aktualisierungsgatter einer LSTM Zelle
		$J(\theta)$	Zielfunktion des Gradientenabstiegs
γ	Impuls eines stochastischen Gradientenabstiegs	K	
g	Aktivierungsfunktion eines Neuron	k	Anzahl Filterkerne einer Faltungsschicht
g	Gravitationskonstante eines Beschleunigungssignals	k	Distanzparameter des k-NN Verfahrens
g	Gruppe aus Bündeln von ToF Werten d und Referenzposition p der Funklokalisierung	K	Kalibrierungsparameter einer modellbasierten Schrittängenschätzung
g^n	Gravitationskraftvektor eines Beschleunigungssignals	K_s^a	Kernelfunktion eines ML Verfahrens
g_t	Gradient g im Zeitschrift t	K	Kalman-Filter Gewinn
G	Stapel mit Gruppen g der Funklokalisierung	K_g	Genereller Kalibrierungsparameter einer modellbasierten Schrittängenschätzung
gyr	Drehratenmessung oder Drehratendatenstrom	kF	Kalibrierungsparameter einer modellbasierten Schrittängenschätzung für eine bestimmte Person s und eine bestimmte Aktivität a
gyr_{SG}	Drehrate mit SG gefiltert	L	Rician k-Faktor der Funklokalisierung
gyr^{raw}	Rohe ungefilterte Drehrate	λ	Zerfallsrate der Lernrate η
		l	Cluster der Verzögerungsspreizung der Funklokalisierung
		l	Schicht eines NN
		$l(t)$	Übergangsfunktion eines KF

L	Modell zur Schrittängenschätzung	p	Position
L	Upsampling" oder Interpolationsfaktor der Datenvorverarbeitung	p_0	Ursprungposition der Posenschätzung
L	Verlustfunktion mit Fehler E eines NN	p_{abs}	Absolute Position der Orientierungsschätzung
le	Linke Flanke der Wendepunktmethode der Funklokalisierung	p_{abs}^{ref}	Absolute Referenzposition der Orientierungsschätzung
M		p_{est}	Geschätzte Position oder Pose der Posenschätzung
m	Aktualisierungsvektor des stochastischen Gradientenabstiegs	p_{radio}	Funkposition
m	Eingabemesswerte einer Eingabesequenz eines ML und DL Verfahrens	p_{ref}	Referenzposition oder Referenzpose der Posenschätzung
\vec{m}	Bewegungsrichtung der Orientierungsschätzung	p_τ	Spitzenschwellenwert der Funklokalisierung
M	Modellgetriebenes oder datengetriebenes Modell, bzw. Optimierer, bspw. KF oder LSTM	p_t	Zweidimensionaler Positionsvektor der x und y Dimensionen zum Zeitpunkt t
M_{high}	Filterordnung für Hochpassfilter in der Rückkopplung	$p(t)$	Position der Geschwindigkeitsschätzung im Zeitschritt t
M_{low}	Filterordnung für Tiefpassfilter in der Rückkopplung	\hat{p}	Geschätzte Position
N		P	Dimension der Verschiebung (Pooling)
η	Korrekturwert	P	Kovarianz des KF
η	Lernrate, bzw. GD Schrittgröße	P	Leistung [dB], bspw. eines MPC Clusters der Funklokalisierung
n	Anzahl der Kanäle und Schritte s einer Faltungsschicht	P	Polynomordnung des SG Filters
n	Fensterlänge des SG Filters	P_{LoS}	Potenz des direkten Pfads
n	Globaler Navigationsrahmen	P_{NLoS}	Summe der Potenzen aller Mehrwegkomponenten
n	Grad Anzahl eines Polynoms des SMA- und Exponential-Filters	pca	Größter Hauptkomponenten der Hauptkomponentenanalyse
N_{high}	Filterordnung für Hochpassfilter in der Vorrwärtskopplung	Q	
N_{low}	Filterordnung für Tiefpassfilter in der Vorrwärtskopplung	q_n	Parameter des Exponentialfilters
N_s	Summe einzelner Signalfäde der Funklokalisierung	q_b^n	Einheitsquaternion
N_w	Fenstergröße oder Größe des Schiebefensters einer Eingabesequenz	Q	KF Prozessrauschen
O		R	
o_i	Ausgabegatter einer LSTM Zelle im Zeitschritt i	r	Anzahl Neuronen pro Schicht
θ_{rel}	Relative Orientierung der Orientierungsschätzung	r	Radius, bzw. Distanz, zwischen 2 Positionen bei der Umwandlung von kartesischen und polaren Koordinaten
P		r	Langfristiges Rauschen, zeitliches Verhalten der Fehler, der Geschwindigkeitsschätzung
ϕ	Ankunftsinkel der Funklokalisierung	\vec{r}	Reale Kopf" / Blickrichtung der Orientierungsschätzung
Φ	Phase der Funklokalisierung	r_t	Rücksetzgatter einer LSTM Zelle
ϕ	Rollwinkel der Orientierungsschätzung	R	Ausgabe bzw. resultierender Stapel mit Ergebnissen r eines LSTM
ψ	Gier Winkel	R	KF Messrauschen
ψ'	Drift der Kopforientierung bzw. Versatz zwischen den realen \vec{r} und der virtuellen \vec{r} Kopfrichtung	$radio$	Rotationsmatrix
			Funksystem das Funksignale, nämlich CIR und ToA Datenströme, und Funkpositionen p_{radio} liefert
		ref	Referenzpose der Posenschätzung oder Referenzposition der Geschwindigkeitsschätzung
		$ReLU$	Rectified Linear Unit
		S	

Symbolverzeichnis

σ	Aktivierungsfunktion eines Neurons	T	Visuelles Ziel, um einen gewünschten
σ	Standardabweichung		Blickwinkel ω in VR einzuhalten
σ^2	Varianz	T	Kürzeste Abtastperiode, bzw. inverse Abtastfrequenz, nach Shannon-Nyquist der Funklokalisierung mit
s	Person der modellbasierten Schrittängenschätzung		$T = 1/f_{S\min}$
s	Segment oder Segmentlänge eines Datensatzes oder Sequenzlänge oder Teil einer Trajektorie t	T_i	Signalausbreitungszeit der Funklokalisierung
s_b	Stapelgröße	\tanh	Tangens hyperbolicus Aktivierungsfunktion
$s(t)$	Gesendetes Signal, bspw. Pseudozufallssequenz, der Funklokalisierung	ToA_{rel}	Relative Ankunftszeit der Funklokalisierung
S	Softmax Funktion eines NN		
S	Schrittänge der modellbasierten Schrittängenschätzung	U	
$S(t)$	Analoges Signal der Signalverarbeitung		
SD	Standard Deviation (dt. Standardabweichung)	μ	Mittelwert
		u	Messinterferenz der Signalverarbeitung
		U	Gewichtsmatrix eines Neurons
T			
δ	Relative ToA der Funklokalisierung	V	
τ	Verzögerungen, bzw. relative ToA, die entlang einer Verzögerungsleitung mit L MPCs einen CIR, h , beschreiben	v	Geschwindigkeit
τ_l	Verzögerung, bzw. relativer ToA	v_i	Geschwindigkeit im Zeitschritt i
		v_k	Messrauschen eines Kalman-Filters
		v_{ref}	Referenzgeschwindigkeit
		$v(t)$	Geschwindigkeit im Zeitschritt t
		v'	Geschwindigkeitsgeschätzung und Eingangsgeschwindigkeit für einen LKF Optimierer
O		v''	Geschätzte und durch LKF optimierte Geschwindigkeit
θ	Orientierung	\vec{v}	Startorientierung der Orientierungsschätzung
T		\vec{v}	Virtuelle Blickrichtung der Orientierungsschätzung
θ	Gier-Winkel der Geschwindigkeitsschätzung	$\vec{v} + \psi'$	Endorientierung der Orientierungsschätzung
θ	Nickwinkel der Orientierungsschätzung		
θ	Parametersatz mit Gewichten w und Bias b der Rückpropagierung		
θ_{ori}	Gier-Winkel der Posenschätzung	W	
θ_{ref}	Referenzorientierung der Posenschätzung	ω	Rotationsgeschwindigkeit
θ_{radio}	Gier-Winkel des Funksystems aus zwei konsekutiven Positionen	ω	Versatz zwischen der realen und der virtuellen Bewegungsrichtung
t	Zeitschritt für Modell-interne Betrachtungen	ω_{imm}	Kleiner und immersiver Teil der Drift ψ'
t	Trajektorie der Positionsschätzung	ω_g	Butterworth-Tiefpassfilters zweiter Ordnung
t	Zeitpunkt oder Zeitschritt, mit δt der Länge eines Zeitschrittes	$\omega(t)$	Winkelrate bzw. Winkelgeschwindigkeit im Zeitschritt t
t_{live}	Durchschnittliche Zeit der Vorhersage der Geschwindigkeit für ein einzelnes Fenster-/Merkmalsbündel oder Inferenzzeiten	$\omega=n^\circ$	Momentaufnahme der Bewegung, wenn der Versatz ω um n° von der realen Bewegungsrichtung \vec{m} abweicht
t_{train}	Gesamte Trainingszeit	w	Dynamisches Messrauschen nicht deterministisches, dynamisches Rauschen als stochastische Größe (zeitlich unkorreliertes dynamisches Rauschen) folgt dabei einer mehrdimensionalen Normalverteilung mit Erwartungswertvektor 0 und Kovarianzmatrix Q
T	Übertragungszeit der Signalverarbeitung		
T	Zeitintervalle der Datenvorverarbeitung		

w	Fenster eines Datensatz Segmente	X_{int}	Eingabedaten einer Schicht, bzw. interne Repräsentation der Eingabedaten
w	Gewicht eines Neuron		
w	Wellenlänge der Positionsschätzung	X_n	Merkmalsvektor der Länge n
w_v	Fenstervektor eines ML oder DL Verfahrens		
W	Gewichtsmatrix mit Gewichten w eines Neurons		
W^T	Transponierte Gewichtsmatrix in der BPTT	y	Normalisierte Aktivierung des Stapels der Stapelnormalisierung
X		y	y -Dimension der Position
x	X-Koordinaten im euklidischen Koordinatensystem	y	Aktuelle Vorhersage, Schätzung
x	Aktivierung zu jedem Datenpunkt x_i der Stapelnormalisierung	y	Ausgabe $y=f(x)$ bzw. $y=f(x \cdot w + b)$, eines NN und Ausgabe je Schicht
x	x-Dimension der Position	y_i	Gefilterte Beschleunigung im Zeitschritt i
x_0	Startzustand des KF	y_k	Gefilterter Wert einer Zeitreihe
x_i	Beschleunigungsmessungen im Zeitschritt i der Geschwindigkeitsschätzung	$y(t)$	Empfangenes Signal der Funklokalisierung
x_i	Kanaldimension des BLSTM	\hat{y}	Referenz und Trainingsziel und Ausgabe oder erwartete Vorhersage
x_k	Messpunkt einer Zeitreihe im Zeitschritt k	\hat{y}_n	Aktuelle Geschwindigkeitsschätzung als Endergebnis für v'
x_t	Zustand eines KFs im Zeitschritt t	Y	Ausgabedaten, Schätzung und Ausgabeschicht
\hat{x}_i	Gradient in einer Resteinheit	\hat{Y}	Ziel, bzw. Referenz eines Schätzers
x, X	Eingangsdaten, eines NN und Eingangsdaten je Schicht	\tilde{Y}, \hat{Y}	Ausgabe, bzw. Ergebnis eines Schätzers
X	Zustand eines KF definiert durch μ Mittelwert und Σ Varianz	Y^n	Geschwindigkeitsschätzung
X_i	Eingangsdaten und Eingabeschicht im Zeitschritt i		
\hat{X}_i	Interne Zwischenwerte der Eingangsdaten in einer Resteinheit	z	Logit-Funktion eines NN, die unnormalierte Ausgabewerte liefert
\hat{X}'_i	Interne Zwischenwerte der Eingangsdaten in einer Resteinheit, die an eine andere Schicht ausgegeben werden	z_t	Aktualisierungsgatter einer GRU Zelle
		z_t	KF Beobachtung im Zeitschritt t
		Z	Messprozess, bzw. die Beobachtungsgleichung mit Beobachtung z

Glossar

- ANN** Ein künstliches neuronales Netz (engl. Artificial Neural Network, ANN) basiert auf einer Menge verbundener künstlicher Neuronen. Neuronen werden zu Schichten zusammengefasst. Die Daten bzw. Signale fließen von der ersten Schicht (der Eingabeschicht) zur letzten Schicht (der Ausgabeschicht), möglicherweise nachdem sie die Schichten mehrmals (rekursiv) durchlaufen haben. Jede Verbindung kann ein Signal an andere Neuronen übertragen. Ein künstliches Neuron, das ein Signal empfängt, verarbeitet das Signal und kann wiederum verbundene Neuronen kontaktieren. Die Ausgabe jedes Neurons wird durch eine nichtlineare Funktion berechnet. Gewichte passen sich mit fortschreitendem Lernen an die Eingaben an und erhöhen oder verringern die Stärke des Signals. Unterschiedliche Schichten können die Eingaben unterschiedlich transformieren. [56]
- BLSTM** Bidirektionale Lang-Kurzzeitgedächtnis Netze (engl. Bidirectional Long-Short-Term Memory, BLSTM) verbinden zwei verborgene Schichten entgegengesetzter Flussrichtung der Eingangsdaten mit demselben Ausgang. Dadurch kann die Ausgabeschicht gleichzeitig Informationen aus vergangenen (rückwärts) und zukünftigen (vorwärts) Zuständen erhalten, um die Menge der dem Netz zur Verfügung stehenden Eingabeinformationen zu erhöhen. Im Gegensatz zu anderen Architekturen müssen die Eingabedaten des BLSTM nicht festgelegt werden und zukünftige Eingabeinformationen sind vom aktuellen Zustand aus erreichbar. BLSTM verbessern die Genauigkeit der Bewegungserkennung durch Kenntnis der Bewegungsform vor und nach der aktuellen Bewegung. [392]
- CIR** Um den Ausbreitungsweg zwischen jedem Sender“ und jedem Empfängerelement in einem funkbasierten Lokalisierungssystem zu charakterisieren, wird ein (Breitband)Signal ausgesendet. Die kontinuierliche periodische Sequenz des Senders ist mit der ursprünglichen Sequenz korreliert und kommt verändert durch die Ausbreitungsumgebung und Rauschen beim Empfänger an. Diese impulsartige Korrelationsfunktion wird als Kanalimpulsantwort (engl. Channel Impulse Response, CIR) bezeichnet. CIRs werden genutzt, um die Kanalumgebung zu charakterisieren, Ankunftszeitpunkte zu bestimmen, Mehrwege und deren Effekte auf die Signalausbreitung zu beschreiben und die Lokalisierungsleistung zu verbessern. [477]
- CNN** Faltendes neuronales Netz (engl. Convolutional Neural Network, CNN) bestehen aus einer oder mehreren Faltungsschichten, gefolgt von einer Gruppierungsschicht. Die Aktivität jedes Neurons wird über eine Faltung berechnet. Dabei wird schrittweise eine (kleine) Faltungsmatrix, der Filterkernel, über die Eingabe, eine Matrix z. B. die Pixel eines Bildes bewegt und das innere Produkt des Filterkernels mit dem aktuell unterliegenden Bildausschnitt berechnet. Dementsprechend reagieren benachbarte Neuronen einer Faltungsschicht auf sich überlappende Bereiche, bspw. ähnliche Frequenzen in Funk“ oder Trägheitssignalen. [356]
- DL** Tiefes Lernen (engl. Deep Learning, DL) ist eine Klasse von Algorithmen für maschinelles Lernen, die mehrere (mindestens zwei verborgene) Schichten verwenden, um schrittweise übergeordnete Merkmale aus den Eingabedaten zu extrahieren. Bspw. identifizieren bei der Bildverarbeitung mit Faltungsschichten niedrigere Schichten Kanten, während höhere Schichten Konzepte wie Buchstaben oder Gesichter identifizieren. [56]
- DR** Koppelnavigation (engl. Dead Reckoning, DR) beschreibt ein Verfahren zur Berechnung der aktuellen Position eines sich bewegenden Objekts unter Verwendung einer zuvor bestimmten Position und des anschließenden Einbeziehens von Schätzungen der Geschwindigkeit und der Ausrichtung der Bewegung über die verstrichene Zeit. Koppelnavigation unterliegt kumulativen Fehlern. Daher wird DR bislang nur von Trägheitsnavigationssystemen verwendet, die sehr genaue Informationen liefern. [58]

Glossar

- Drift** Im Rahmen der Dissertation ist Drift folgendermaßen definiert: Drift ist eine unerwünschte, unbekannte, langsame, langfristige, oft zufällige Änderung einer Eigenschaft in einer Zeitreihe oder in einem System, die auch ungeordnet variieren kann. Im Gegensatz zu zeitkonstanten systematischen Abweichungen, wie z. B. einem Trend, kann eine Drift nur, wenn überhaupt, durch wiederholte Messungen anhand von (langen) Zeitreihen festgestellt werden. [50]
- FDPoA** Der erste direkte Ankunfts pfad (engl. First Direct Path of Arrival, FDPoA) eines empfangenen Funksignals bestimmt die fehlerfreie Ankunftszeit des Signals und liefert genaue Positionen. Allerdings wird der Ankunfts pfad von OLoS und NLoS umgeleitet, sodass der tatsächliche direkte Pfad indirekt und geschwächt oder gar nicht beim Empfänger ankommt und die Bestimmung der Ankunftszeit und somit der Position verfälscht oder unmöglich ist. [9]
- FFNN** Ein vorwärts gekoppeltes neuronales Netz (engl. Feedforward-Neural Network, FFNN) ist der älteste und einfachste Typ eines KNN. In FFNN fließen die Informationen nur vorwärts, d. h. von der Eingabeschicht über die (optionalen) verborgenen Schichten zu den Ausgabeschichten ohne Schleifen. Daher unterscheiden sie sich von rekurrenten neuronale Netzen. [56]
- FRFNN** Diese Architektur wurde in dieser Arbeit erforscht und kombiniert vorwärts gekoppelte (FF) mit rekurrenten (R) und vorwärts gekoppelten (FF) Schichten in einem neuronalen Netz (NN). Experimente zeigen, dass FRFNN die Posen und Trajektorien aus „Funk“ und Trägheitssignalen genauer und zuverlässiger rekonstruiert als herkömmliche PDR-Methoden.
- GRU** Eine gegatterte rekurrente Zelle (Gated Recurrent Unit, GRU) ist ein Gatter-Mechanismus in RNNs, der den Informationsfluss in einem NN beeinflusst. Die GRU hat weniger Gatter und weniger Parameter als die LSTM-Zelle, die ebenfalls anders strukturiert sind als die Gatter der LSTM-Zelle. Experimente zeigen, dass die Posenschätzung von GRU etwas ungenauer und unzuverlässiger ist als die von LSTM. [76]
- Hybrid** Diese Architektur wurde im Rahmen dieser Dissertation erforscht und kombiniert Resteinheiten (engl. residual units, vgl. mit ResNet) mit BLSTM-Schichten, um hochgenaue Geschwindigkeiten zu schätzen. Hybrid optimiert diese Schätzungen weiter mit einem nachgeschalteten linearen Kalman-Filter. Experimente zeigen, dass Hybrid Geschwindigkeiten aus dem Signalgrößenvektor der Rohbeschleunigung eines Trägheitssensors genauer und zuverlässiger rekonstruiert als herkömmliche Verfahren.
- iGPS** iGPS ist ein skalierbares Referenzlokalisierungssystem in Innen“ und Außenbereichen. Im Gegensatz zu GPS werden GPS-Satelliten durch Infrarot-iGPS-Lasersender ersetzt, die ein Messfeld von der Größe eines Raumes ermöglichen. iGPS bietet eine Genauigkeit, die signifikant höher ($200 \mu\text{m}$) ist als GPS-Systeme. Z. B. können Fußgänger mit iGPS-Empfängern ausgestattet werden, die von den Sendern verfolgt werden. Daher können Positionen genau gemessen oder dynamisch verfolgt werden, um genaue Referenzen bereitzustellen.
- IMU** Eine Trägheitsmesseinheit (engl. Inertial Measurement Unit, IMU) ist ein elektronisches Gerät, das die Beschleunigung, Drehrate oder Winkelgeschwindigkeit und (optional) die Ausrichtung eines Körpers mithilfe einer Kombination aus Beschleunigungsmessern, Gyroskopen und (optional) Magnetometern misst und meldet. IMUs werden typischerweise zum Manövrieren von Flugzeugen verwendet. Jüngste Entwicklungen der Mikrosysteme ermöglichen die Herstellung von IMU-fähigen GPS-Geräten. Mit einer IMU kann ein GPS-Empfänger Positionen bereitstellen, wenn bspw. in Gebäuden keine GPS-Signale verfügbar sind. [435]
- INS** Ein Trägheitsnavigationssystem (engl. Inertial Navigation System, INS) ist ein Navigationsgerät, das einen Computer, Bewegungssensoren (Beschleunigungsmesser) und Rotationssensoren (Gyroskope) verwendet, um kontinuierlich die Position, die Ausrichtung und die Geschwindigkeit (Richtung und Geschwindigkeit der Bewegung) eines sich bewegenden Objekts ohne externe Referenzen zu berechnen. Oft werden die Trägheitssensoren durch Magnetsensoren (Magnetometer) und Odometriemessgeräte ergänzt. INS werden in mobilen Robotern, Schiffen und Flugzeugen eingesetzt. [435]

- KF** Das Kalman-Filter (KF) wird typischerweise verwendet, um eine Reihe von Messungen über die Zeit zu beobachten. Obwohl die Messungen statistisches Rauschen und andere Ungenauigkeiten enthalten, schätzt das KF unbekannte Variablen genauer als Methoden, die nur auf einer einzigen Messung basieren, indem es für jedes Zeitintervall eine gemeinsame Wahrscheinlichkeitsverteilung über die aktuelle und die vorangegangene Variablen schätzt. [222]
- KNN** Künstliches neuronales Netz, KNN, siehe Artificial Neural Network (ANN). [56]
- LoS** Die Ausbreitung entlang der direkten Sichtverbindung (engl. Line of Sight, LoS) ist ein Merkmal der Ausbreitung elektromagnetischer Strahlung, d. h. Signale bewegen sich mit Lichtgeschwindigkeit auf einem direkten Weg in einer geraden Linie von einem Sender zu einem Empfänger. In LoS Ausbreitungsbedingungen kann der erste und direkte Pfad als korrekter Ankunftszeitpunkt zur fehlerfreien Positionierung verwendet werden. Die Signale können allerdings von Hindernissen gebeugt, gebrochen, reflektiert oder absorbiert werden, siehe OLoS und NLoS, das zu fehlerhaften Ankunftszeiten und Positionen führt. [177]
- LSTM** Das Langzeit-Kurzzeitgedächtnis (engl. Long-Short-Term Memory, LSTM) ist eine rekurrente neuronale Netzarchitektur (RNN). Im Gegensatz zu FFNN verfügt LSTM über Rückkopplungsverbindungen. Es kann einzelne Datenpunkte (z. B. Bilder) sowie ganze Datensequenzen (z. B. Videos) verarbeiten. Eine LSTM-Zelle besteht aus einem Gedächtnis, einem Eingangsgatter, einem Ausgangsgatter und einem Vergessengatter. Das Gedächtnis speichert Werte über ein beliebiges Zeitintervall und die drei Gatter regulieren den Informationsfluss in die Zelle hinein und aus dieser heraus. LSTM wurde entwickelt, um das Problem des verschwindenden Gradienten zu lösen, das beim Training traditioneller RNNs auftritt. LSTM klassifiziert, verarbeitet und prognostiziert basierend auf Zeitreihendaten, trotz Verzögerungen unbekannter Dauer zwischen wichtigen Ereignissen. Die Unempfindlichkeit gegenüber Datenausfällen ist ein Vorteil von LSTM gegenüber RNNs und Hidden-Markov-Modellen. [191]
- ML** Maschinelles Lernen (engl. Machine Learning, ML) bietet Algorithmen, die sich durch Erfahrung und Verwendung von Daten automatisch verbessern. ML Algorithmen erstellen, bzw. lernen, ein Modell basierend auf Beispieldaten, die sogenannten Trainingsdaten, um Vorhersagen oder Entscheidungen zu treffen, ohne explizit dafür programmiert zu sein. ML Algorithmen werden in Anwendungen verwendet, in denen es schwierig oder unmöglich ist, herkömmliche Algorithmen zur Lösung von Problemen zu entwickeln. [56]
- ML-GP** ML-GP ist ein Algorithmus des ML, der mit einem Gaußschen Prozess (engl. Gaussian Process, GP) und einer Kernelfunktion, d.h. ein Maß für die Ähnlichkeit zwischen Punkten, Werte für unbekannte Datenpunkte aus Trainingsdaten vorhersagt. Die Vorhersage ist nicht nur eine Schätzung für diese Punkte, sondern enthält auch Unsicherheitsinformationen. [354]
- MLP** Das Mehrschichtige Perzeptron (engl. MultiLayer Perceptron, MLP) ist eine Klasse von FFNN. Ein MLP besteht aus mindestens drei Schichten: einer Eingabeschicht, einer verborgenen Schicht und einer Ausgabeschicht. Mit Ausnahme der Eingabeschicht enthält jede Schicht mindestens ein Neuron, das eine nichtlineare Aktivierungsfunktion verwendet. Mehrere Schichten und nichtlineare Aktivierungsfunktionen ermöglichen die Unterscheidung von Daten, die nicht linear getrennt werden können. Die Parameter werden durch Fehlerrückführung (engl. Backpropagation) überwacht trainiert. [56]
- MPC** Mehrwege der Funksignalausbreitung bewirken, dass Funksignale auf zwei oder mehr Wegen die Empfangsantenne erreichen, z. B. LoS, OLoS und NLoS. Ursachen für Mehrwege sind z. B. Berge, Gebäude oder Hindernisse in Innenräumen, die das Signal beugen, brechen, reflektieren oder absorbieren. Die Mehrwegeausbreitung verursacht konstruktive und destruktive Störungen und Phasenverschiebungen im Signal. Störungen schwächen das Funksignal oder verschieben seine Phase, sodass die korrekte Ankunftszeit nicht bestimmt werden kann. Wenn eine Mehrwegekomponente (engl. MultiPath Component, MPC), bspw. eine Sichtlinienkomponente, dominiert, wird sie typischerweise als Rician-Fading bezeichnet. [9, 177]

Glossar

- NLoS** Die Funkübertragung ohne Sichtverbindung, sogenannte Nichtsichtverbindung (engl. Non Line of Sight, NLoS) erfolgt außerhalb der direkten Sichtverbindung (LoS) zwischen Sender und Empfänger, z. B. durch Bodenreflexionen oder Behinderung durch ein physisches Objekt zwischen Sender und Empfänger. Zu den Hindernissen, die häufig zur Ausbreitung von NLoS führen, gehören Gebäude, Berge und Objekte in Innenräumen. Einige dieser Hindernisse reflektieren, während andere die Signale einfach absorbieren oder verstümmeln und die effektive Empfangsleistung verringern. So (zer)stören sie die Funkübertragung und reduzieren die Genauigkeit der Lokalisierung. [177]
- OLoS** Im Gegensatz zur NLoS-Funkausbreitung kann die Sichtverbindung zwischen Sender und Empfänger nur teilweise gestört sein, sogenannte gestörte Sichtverbindung (engl. Obstructed Line of Sight, OLoS). OLoS beeinflusst sowohl die Empfangsleistung als auch die klare und korrekte Identifizierung und Interpretation der MPCs, bspw. der Ankunftszeit, und reduziert die Genauigkeit der Lokalisierung. [177]
- PDR** Die Koppelnavigation für Fußgänger (engl. Pedestrian Dead Reckoning, PDR) funktioniert genau wie DR. Die Sensoren sind jedoch an Fußgängern angebracht und unterliegen den dynamischen, nichtlinearen Auswirkungen menschlicher Bewegungen. [58, 435]
- RNN** Ein rekurrentes neuronales Netz (engl. Recurrent Neural Network, RNN) ist eine Klasse von KNN. Im Gegensatz zu KNN, bilden die Verbindungen zwischen Neuronen einen gerichteten Graphen entlang einer Zeitsequenz, um dynamisches Verhalten über die Zeit abzubilden. RNNs nutzen ihren internen Zustand (Speicher), um Sequenzen von Eingaben variabler Länge zu verarbeiten. Dies ermöglicht die Vorhersage sehr individueller Bewegungsformen und Bewegungsparameter und der Posen" und Trajektorienrekonstruktion. [109]
- SINS** Das fahrzeugfeste Trägheitsnavigationssystem (engl. Strapdown Inertial Navigation System, SINS) funktioniert genauso wie das INS, allerdings werden die Sensoren direkt und fest am Fahrzeug befestigt. Das reduziert den Kalibrierungsaufwand und erhöht die Genauigkeit, indem einige der beweglichen Teile eliminiert bzw. deterministisch werden. [435]
- SMV** Der Signalbetragvektor (engl. Signal Magnitude Vector, SMV) deutet die Bewegungsintensität und Beschleunigungsspitzenwellen an. SMV wird aus dem Beschleunigungssignal geschätzt und ermöglicht die Zerlegung der Beschleunigung in Gravitationskraft und lineare Beschleunigung, ohne die absolute Ausrichtung des Beschleunigungsmessers zu kennen.
- SVM** Die Stützvektor-Maschine (engl. Support-Vector Machine, SVM) ist eine der robustesten überwachten Klassifikations", Regressions" und Vorhersagemethoden. Zur Klassifikation, wird bspw. für eine Reihe von binären kategorischen Trainingsbeispielen, ein Modell erlernt, das der einen oder anderen Kategorie neue Beispiele zuweist und dabei die Distanz zwischen den beiden Kategorien maximiert. Zusätzlich zur linearen Klassifizierung können SVMs mithilfe des Kernel-Tricks effizient eine nichtlineare Klassifizierung durchführen. [87]
- TDoA** Der Zeitunterschied der Ankunft (engl. Time Difference of Arrival, TDoA) ist der Unterschied zwischen den Ankunftszeiten (ToA) mehrerer Empfänger. TDoA-Techniken wie die Multilateration verwenden die gemessene Zeitdifferenz zwischen ToAs zur Positionsbestimmung. [298]
- ToA** Die Ankunftszeit (engl. Time of Arrival, ToA) ist der absolute Zeitpunkt, zu dem ein von einem Sender ausgehendes Funksignal einen Empfänger erreicht. Die seit der Übertragung verstrichene Zeitspanne (engl. Time of Transmit, ToT) ist die ToF. Der Zeitunterschied der Ankunft TDoA ist der Unterschied zwischen den Ankunftszeiten ToAs verschiedener Empfänger des selben Funksignals. Viele Funkortungssysteme verwenden ToA-Messungen, um eine Positionierung über eine Multilateration durchzuführen. Die Entfernung kann direkt aus der ToA berechnet werden, wenn sich Signale mit einer bekannten Geschwindigkeit fortbewegen. ToAs von zwei Empfängern grenzen eine Position auf einen Positionsreich ein und ToAs eines dritten Empfängers lösen die genaue Position in einem Punkt auf. [477]

ToF Die Flugzeit (engl. Time of Flight, ToF) ist die seit der Übertragung eines Funksignals von einem Sender verstrichene Zeitspanne. [298]

UWB Ultra-Breitband (engl. Ultra-WideBand, UWB) ist eine Funktechnologie, die ein sehr niedriges Energieniveau für kurzreichweite (> 100 m) Kommunikation mit hoher Bandbreite (> 500 MHz) über einen großen Teil des Funkspektrums verwendet. Seit 2019 unterstützen viele Mobiltelefone, bspw. Apple iPhone 11, die UWB-Funktechnologie. UWB ermöglicht sowohl die Übertragung einer großen Menge an Signalenergie als auch eine hochgenaue Positionierung der Sender“ und Empfängereinheiten. [298]

VR Die virtuelle Realität (engl. Virtual Reality, VR) ist eine simulierte Erfahrung, die der realen Welt ähnlich oder völlig anders sein kann. VR-Anwendungen umfassen bspw. Unterhaltung und Bildung. Gegenwärtig werden typischerweise am Kopf getragene VR-Systeme, die aus am Kopf montierten Bildschirmen vor den Augen und anderen Sensoren bestehen, verwendet. Diese erzeugen realistische Bilder, Töne und andere Empfindungen, die die physische Präsenz eines Benutzers in einer virtuellen Umgebung simulieren. Ein VR-Nutzer kann sich umschauen, sich bewegen und mit virtuellen Objekten in der künstlichen Welt interagieren. [416]

Wörterbuch

In der Tabelle 0.1 sind international anerkannte Begriffe der englischen Sprache, ihre Übersetzung ins Deutsche und die in der Wissenschaft anerkannte Abkürzung aufgeführt.

Tabelle 0.1: Deutsche Begriffe, deren Übersetzung ins Englische und (optional) die verwendete Abkürzung.

Deutsch	Englisch (Amerikanisch)	Abk.
[o-9] 10-fache Kreuzvalidierung	10-fold cross-validation	CV
A		
Abflachungsschicht	Flatten layer	
Abgegriffene Verzögerungsleitung	Tapped delay line	
Abschattung	Shadowing	
Abstrakte	Samplingrate	
Abstratenerhöhung	Upsampling	
Aktivierungsfunktion	Activation functions	
Aktualisierungsgatter	Update gate	
Anfängliche Lernrate	Initial learning rate	
Anfängliche Verzögerung	Initial delay	
Ankunftsinkel	Angle of Arrival	AoA
Ankunftszeit des Übertragungssignals	Time of Arrival	ToA
Ankunftszeitdifferenz	Time-Difference-of-Arrival	TDoA
Funksignal-Schwingung	Radio signal burst	
Anzahl der Filterkerne	Number of filter kernels	F
Ausgabe	Output	
Ausgabeschicht	Output layer	
Ausgabegatter	Output gate	
Außerhalb der Laufzeit	Offline	
Azimut	Azimuth	
B		
Beschleunigungsmesser	Accelerometer	Acc.
Beugung	Diffraction	
Bidirektionales Lang-Kurzzeitgedächtnis	Bidirectional Long-Short-Term-Memory	BLSTM
Echtzeit-Lokalisierungssystem	Real-Time Locating System	RTLS
Bündel	Bundle	
D		
Dämpfung	Attenuation	
Darstellen	Render	
Datenpunkt	Sample	
Dichte Schicht	Dense layer	
Dimension	Dimension	
Drehratensensor	Gyroscope	GYRO
E		
Eindringlich	Immersive	
Eine-gegen-Alle	One-vs-All	OvA
Eingabeschicht	Input layer	X
Eingangsgatter	Input gate	

Wörterbuch

Einheitsvarianz	Unit variance	
Emissionsverteilung	Emission distribution	
Endliche Impulsantwort	Finite Impulse Response	FIR
Engpassschicht	Bottleneck layer	
Entfernungsfehler pro Meter	Distance Error Per Meter	DEPM
Entfernungsfehlersumme	Distance Error Total	DET
Entrauschen	Denoise	
Entscheidungsbaum	Decision Tree	DT
Entscheidungsbaum Klassifikation	Decision Tree Classification	DTC
Entscheidungsbaum Regression	Decision Tree Regression	DTR
Epochen	Epoches	
Erweitertes Kalman-Filter	Extended Kalman-Filter	EKF
Explodierender Gradient	Exploding gradient	
F		
Fahrzeugfest	Strapdown	
Fahrzeugfestes Trägheitsnavigationssystem	Strapdown Inertial Navigation System ...	SINS
Falten	Convolute	
Faltendes neuronales Netz	Convolutional Neural Network	CNN
Faltungsschichten	Convolutional layer	
Fehlerrückführung	Backpropagation	BP
Fehlerrückführung durch die Zeit	Backpropagation Through Time	BPTT
Fensterbündel	Window bundle	
Filterkerngröße	Filter kernel size	
Fingerabdruck	Fingerprint	
Forschungsfrage	Research Question	RQ
Frühzeitiges Anhalten	Early stopping	
Funk	Radio	
Funkbasierte Lokalisierung	Radio localization	
Funksender	Beacon	
Funksignalstärke	Radio Signal Strength	RSS
Fußgänger Koppelnavigation	Pedestrian Dead Reckoning	PDR
G		
Gatter	Gate	
Gaußprozess	Gaussian Process	GP
Gaußprozess Klassifikation	Gaussian Process Classification	GPC
Gaußprozess Regression	Gaussian Process Regression	GPR
Geometrische Verdünnung der Präzision	Geometrical dilution of precision	
Gestörte Sichtbedingungen	Obstructed Line of Sight	OLoS
Gegarterte rekurrente Einheit	Gated Recurrent Units	GRU
Gewichtsinitialisierung	Weight initialization	
Gierwinkel	Yaw	ψ
Glättung	Smoothing	
Gleichgerichtete lineare Einheitsschichten	Rectified Linear Unit layers	ReLUs
Gleitendes Mittelwert Filter	Sliding Moving Average	SMA
Globale Maximagruppierungsschicht	Global max pooling layer	
Globales Positionsbestimmungssystem	Global Positioning System	GPS
Gradientenabstieg	Gradient Descent	GD
Gradientenbeschneidung	Gradient clipping	
Grenzfrequenz	Cut-off frequency	
Gruppierungsschicht	Pooling layer	
Guckloch	Peephole	
Gültigkeitsannahme	Assumption of validity	
H		
Hauptkomponentenanalyse	Principal Component Analysis	PCA
Heruntertaktung	Downsampling	

Hochfrequenz	Radio Frequency	RF
Hochtaktung	Upsampling	
I		
Identifizierung mit Hilfe elektromagnetischer Wellen	Radio-Frequency Identification	RFID
Identitäts rekurrentes neuronales Netz	Identity Recurrent Neural Network	IRNN
Indikator für Empfangsfeldstärke	Received Signal Strength Indicator	RSSI
Inertielle Messeinheit	Inertial Measurement Unit	IMU
Inertiales Navigationssystem	Inertial Navigation System	INS
Informationsfusion	Information fusion	
Inkrementelle Glättung von Verzögerungen	Incremental Smoothing and Mapping	ISAM
Interne Aktionseffekte	Internal action effects	
Interquartilsabstand	InterQuartile Range	IQR
Inverses Porträt	Inverse portrait	
K		
k-Nächste-Nachbarn	k-Nearest-Neighbor	kNN
k-Nächste-Nachbarn Klassifikation	k-Nearest-Neighbor Classification	kNNC
k-Nächste-Nachbarn Regression	k-Nearest-Neighbor Regression	kNNR
Kalman Verstärkung	Kalman gain	
Kamera-trägheitsbasierte-Systeme	Visual Inertial Navigation System	VINS
Kanalimpulsantwort	Channel Impulse Response	CIR
Keine-Pose-Verfolgung	No-Pose-Tracking	NP
Komplexe Koeffizienten	Complex coefficients	
Konstante Geschwindigkeit	Constant velocity	
Kontextvektor	Context vector	
Koppelnavigation	Dead Reckoning	DR
Kostenfunktion	Cost function	
Kovarianz	Covariance	
Kovarianz des Zustandsübergangsrauschens	Covariance of the state transition noise	
Kubatur Kalman-Filter	Cubature Kalman-Filter	CKF
Kumulative Verteilungsfunktion	Cumulative Distribution Function	CDF
Künstliches neuronales Netz	Artificial Neural Network	KNN
Kurs	Heading	
Kursschätzung	Heading estimate	
Kurzzeit	Short-term	
L		
Lang-Kurzzeitgedächtnis	Long-Short-Term-Memory	LSTM
Laufzeitmessung	Time of Flight	ToF
Leistung	Power	
Lernrate	Learning rate	
Lineare Bewegungsübergangsfunktion	Linear motion transition function	
Lineare Zustandsübergangsmodelle	Linear state transition models	
Löser	Solver	
M		
Magnetometer	Magnetometer	MAG
Maximagruppierungsschicht	Max pooling layer	
Mehrschichtige Perzeptrone	Multi-Layer Perceptrons	MLP
Mehrwegeausbreitung	Multipath propagation	
Mehrwegkomponenten	MultiPath Component	MPC
Merkmalzbündel	Feature bundle	
Merkmalsextraktion	Feature engineering	
Merkmalsslankarte	Feature-map	
Merkma尔斯strom	Feature stream	
Messmodell	Measurement model	
Messrauschen	Measurement noise	

Wörterbuch

Mischen	Shuffle	
Mittelwert	Mean	μ
Mittelwertfrei	Zero mean	
Mittelwertfreie Wahrscheinlichkeitsdichtefunktion	Zero-mean Probability Density Function	PDF
Mittlerer absoluter Fehler	Mean Absolute Error	MAE
Mittlerer quadratischer Fehler	Mean Square Error	MSE
Mittleres Abweichungssquadrat	Root Mean Square Error	RMSE
Mobiles intelligentes Telefon	Smartphone	
N		
Nachverarbeitung	Post-processing	
Nachverarbeitungskette	Post-processing pipeline	
Neunachsiger Freiheitsgrad	9 Degrees of Freedom	9DoF
Neuronales Netz	Neural Network	NN
Nichtsichtverbindung	Non Line of Sight	NLoS
Niedrigeres Abtasten	Downsampling	
Nullgeschwindigkeitsaktualisierung	Zero Velocity Update	ZUPT
Nullwinkelgeschwindigkeitsaktualisierung	Zero Angular Rate Update	ZARU
O		
Offenen Schleifen Integration	Open-loop-integration	
Optimierer	Optimizer	
P		
Parameter für große Flächen	Large Scale Parameters	LSP
Positionsversatz	Displacement	
Programmierschnittstelle	Application Programming Interface	API
Prozessrauschen	Process noise	
Pur	Raw	
Q		
Quadratisches Mittel	Root Mean Square	RMS
R		
Radialbasisfunktion	Radial Basis Function	RBF
Rastersuche	Grid search	
Referenz	Label	
Referenzbezeichnung	Ground truth label	
Referenzmomente	Ground Truth Moments	GTM
Reflexion	Reflection	
Regressionsschicht	Regression layer	
Rekurrentes neuronales Netz	Recurrent Neural Network	RNN
Resteinheit	Residual unit	
Restnetz	Residual Network	ResNet
Roh	Raw	
Rotationsquadrat	Rotational Quadratic	RQ
Rücksetzgitter	Reset-Gate	
Rückwärts	Backward	
Rückwärtsschicht	Backward layer	
S		
Schicht	Layer	
Schnappschuss	Snapshot	
Schnelle Fourier-Transformation	Fast Fourier Transform	FFT
Schnittpunkt	Cut-off point	
Schrittänge	Stride length	
Schwarzer Kasten	Blackbox	
Schwund	Fading	
Schwund durch Bewegung	Motion fading	

Schwund in großem Maßstab	Large-Scale-Fading	LSF
Schwund in kleinem Maßstab	Small-Scale-Fading	SSF
Schwung	Momentum	
Segmentbündel	Segmentation bundle	
Sendeeinheiten	Transmitter	
Sendeempfangsleitung	Transceiver-Pipeline	
Sensorfusion	Sensor fusion	
Sequenzlänge	Sequence length	
Sichtverbindung	Line of Sight	LoS
Sigma-Punkt Kalman-Filter	Unscented Kalman-Filter	UKF
Sigma-Punkt Quadratwurzel-Kalman-Filter	Unscented Square-Root Kalman-Filter	USRKF
Sigmoid-Gatter-Aktivierungsfunktion	Sigmoid gate activation function	
Signal-Rausch-Verhältnis	Signal to Noise Ration	SNR
Signalspitze	Peak	
Signalverzerrung	Signal drift	
Simultane Lokalisierung und Kartografie	Simultaneous Localization and Mapping	SLAM
Sprung-Verbindungen	Skip connections	
Standardabweichung	Standard deviation	
Stapel	Batch	
Stapel-Normalisierungs-Schicht	Batch Normalization layer	BN
Stapelgröße	Batch size	
Stichprobewiederholung	Resampling	
Störer	Scatterer	
Streuung	Scattering	
Stützvektor Klassifikation	Support Vector Classification	SVC
Stützvektor Regression	Support Vector Regression	SVR
Stützvektormaschine	Support Vector Machine	SVM
Systematische Verzerrung	Systematic bias	
T		
Taktverschiebung	Clock drift	
Tiefe neuronale Netze	Deep Neural Networks	DNN
Tiefe	Depth	
Trägheitsmesseinheit	Inertial Measurement Unit	IMU
Trainingsziel	Target	
Tröpfchen	Blob	
U		
Überabtastung von Minderheiten	Minority oversampling	
Überanpassung	Overshooting	
Übergeordnete Argumentation	High-level reasoning	
Überschießen	Ultra-Wideband	UWB
Ultrabreitband Funktechnologien	Redirected walking	
Umgeleitetes Gehen	Infinite Impulse Response	IIR
Unendlichen Impulsantwort	Underfitting	
Unteranpassung	Undershooting	
Unterschießen	Uplink-Time-Difference-of-Arrival	UTDoA
V		
Verarbeitung	Processing	
Verarbeitungskette	Processing pipeline	
Verborgene Schichten	Hidden layer	
Verbogener Zustandsvektor	Hidden state vector	
Verfolgungsbereich	Tracking area	
Vergessensgatter	Forget gate	
Vergleichsmaßstab	Benchmark	
Verlässlichkeit	Reliability	
Verlust	Loss	

Wörterbuch

Verlustfunktion	Loss function	
Versatz	Offset	
Verschwindende Gradienten	Vanishing gradients	
Verstärkendes Lernen	Reinforcement Learning	RL
Verwerfungsperiode	Drop period	
Verwerfungsschicht	Dropout layer	
Verwerfungsrate	Dropout rate	
Verzerrung	Bias	
Verzerrungsterm	Bias term	
Verzögerung	Delay	
Verzögerungsspreizung	Delay spread	
Virtuelle Realität	Virtual Reality	VR
Visualisierung	Rendering	
Visuelle trägeheitsbasierte Odometrie	Visual-Inertial Odometry	VIO
Visuelle trägeheitsbasierte simultane Lokalisierung und Kartografie	Visual-Inertial Simultaneous Localization And Mapping	VI-SLAM
Vollständig verbundene dichte Schicht	Fully connected dense layer	
Von Innen nach Aussen	Inside-Out	I/O
Vorverarbeitung	Pre-processing	
Vorverarbeitungskette	Pre-processing pipeline	
Vorwärts	Forward	
Vorwärtsgekoppelte neuronale Netze	Feed Forward Neural Networks	FFNN
Vorwärtsschicht	Forward layer	
W		
Wahrscheinlichkeitsdichtefunktion	Probability Density Function	PDF
Winkelspreizung	Angular spread	
Z		
Zeit für eine Rundreise	Round Trip Time	RTT
Zeitlicher Merkmalsverfolger	Temporal Feature Tracker	TFT
Zellgröße	Cell size	
Zirkuläre Fehlerwahrscheinlichkeit	Circular Error Probability	CEP
Zufällige (stochastische) Irrfahrt	Random walk	
Zufällige Verzerrung	Random bias	
Zur Laufzeit	Online	
Zustand	State	
Zustandsraumdarstellung	State-space representation	
Zustandsübergangsmodell	State transition model	
Zuverlässigkeit	Confidence	
Zwischenwerte	Intermediate values	

Inhaltsverzeichnis

Abbildungsverzeichnis	i
Tabellenverzeichnis	vii
Abkürzungsverzeichnis	xi
Akronymverzeichnis	xiii
Symbolverzeichnis	xvii
Glossar	xxiii
Wörterbuch	xxix
1 Einleitung	1
1.1 Einführung	1
1.2 Problembeschreibung	6
1.2.1 Stand der Technik	6
1.2.2 Verrauschte Trägheitsmessungen	7
1.2.3 Mehrwegeausbreitung in Funksystemen	7
1.2.4 Gier-Winkel Schätzung	8
1.2.5 Multimodale Informationsfusion	9
1.3 Beiträge und Forschungsfragen	9
1.4 Abgrenzung der veröffentlichten Beiträge	12
1.4.1 Beiträge zur Positionsschätzung	12
1.4.2 Beiträge zur Geschwindigkeitsschätzung	14
1.4.3 Beiträge zur Orientierungsschätzung	16
1.4.4 Beiträge zur Posenschätzung	18
1.4.5 Wissensvermittlung	20
1.5 Methode	22
1.5.1 Positionsschätzung	23
1.5.2 Geschwindigkeitsschätzung	24
1.5.3 Orientierungsschätzung	24
1.5.4 Posenschätzung	25
1.6 Ergebnisse	26
1.7 Aufbau	26

Inhaltsverzeichnis

2 Verwandte Arbeiten	29
2.1 Lokalisierung mit Trägheitssensoren	29
2.1.1 Magnetfeld	29
2.1.2 Akustisch	30
2.1.3 Optisch	32
2.1.4 Bluetooth und Nahfeldkommunikation	33
2.1.5 Drahtloses lokales Netzwerk	34
2.2 Funk- und trägheitsbasierte Lokalisierung	38
2.2.1 Globales Positionsbestimmungssystem	38
2.2.2 Nach- und Vorteile der Fusion	39
2.2.3 Einfluss der Systemarchitektur	44
2.2.4 Einfluss zusätzlicher Kontextinformation	45
2.2.5 Einfluss von Mehrwegeausbreitung	48
2.3 Methoden zur Kopplung von Funk- und Trägheitssensoren	51
2.3.1 Enge und lose Kopplung	51
2.3.2 Hochspezialisierte Methoden	56
2.3.3 Kalman-Filter	61
2.3.4 Partikel-Filter	66
2.3.5 Kalman- und Partikel-Filter im Vergleich	68
2.3.6 Datengetriebene Methoden	70
3 Grundlagen	75
3.1 Kamerabasierte Lokalisierung	75
3.2 Trägheitsbasierte Lokalisierung	76
3.2.1 Bewegungsgleichungen	77
3.2.2 (Fahrzeugfestes) Trägheitsnavigationssystem	79
3.2.3 (Fußgänger) Koppelnavigation	81
3.3 Funkbasierte Lokalisierung	82
3.3.1 Hochfrequenzlokalisierung	82
3.3.2 Kanalparameter zur Lokalisierung	87
3.3.3 Einschränkungen von Kanalparametern	89
3.3.4 Ankunftszeiten Schätzer	91
3.3.5 Lokalisieren mit Ankunftszeiten	94
3.4 Datenvorverarbeitung	95
3.4.1 Signalerfassung	95
3.4.2 Aggregation	96
3.4.3 Bereinigung	97
3.4.4 Synchronisation	99
3.4.5 Datenreduktion und -erweiterung	100
3.4.6 Transformation	101

3.5	Modell-getriebenen Verfahren	102
3.5.1	Kalman-Filter	102
3.5.2	Erweitertes Kalman-Filter	105
3.6	Künstliche neuronale Netze	106
3.6.1	Neuronen	107
3.6.2	Vorwärtsgekoppelte neuronale Netze	109
3.6.3	Schichten	111
3.6.4	Verlustfunktionen	116
3.6.5	Optimierungsstrategien	117
3.6.6	Datensatz	122
3.6.7	Rastersuche	123
3.7	Sequenzbasierte Lernverfahren	123
3.7.1	Herausforderungen der Zeitreihenanalyse	124
3.7.2	Rekurrente neuronale Netze	125
3.7.3	Optimierungsstrategien	127
3.7.4	Architekturen rekurrenter Zellen	129
3.7.5	Architekturen rekurrenter Netze	141
4	Positionsschätzung	147
4.1	Einführung	149
4.2	Problembeschreibung	152
4.2.1	Terminologie	152
4.2.2	Driftende ToF-Fehler	153
4.3	Verwandte Arbeiten	155
4.3.1	Kalman- und Partikel-Filter	155
4.3.2	Numerische Optimierungsverfahren	156
4.3.3	Verfahren des Maschinellen Lernens	156
4.3.4	Hybride Verfahren	157
4.4	Beiträge	158
4.5	Methode	159
4.5.1	Einführung	159
4.5.2	Architektur	163
4.5.3	Datenvorverarbeitung	164
4.6	Experiment	167
4.6.1	Simulationsumgebung	168
4.6.2	Reale Umgebung	170
4.6.3	Parametrisierung der Verfahren	175
4.7	Auswertung	178
4.7.1	Fehlermetriken	178
4.7.2	Synthetische Datensätze	179

Inhaltsverzeichnis

4.7.3	Reale Datensätze	180
4.8	Diskussion	189
4.9	Zusammenfassung	190
5	Geschwindigkeitsschätzung	193
5.1	Einführung	195
5.2	Problembeschreibung	198
5.2.1	Herausforderungen	198
5.2.2	Lösungsansatz	201
5.3	Verwandte Arbeiten	203
5.3.1	Fahrzeugfeste Trägheitsnavigationssysteme	203
5.3.2	Fußgänger Koppelnavigation	205
5.3.3	Datengetriebene Methoden	207
5.4	Beiträge	211
5.5	Methode	212
5.5.1	Datenvorverarbeitung	213
5.5.2	Ressourcenschonende Hauptverarbeitung	217
5.5.3	Hochgenaue Hauptverarbeitung	225
5.5.4	Nachbearbeitung	235
5.6	Experiment	236
5.6.1	Hard- und Softwarekomponenten	236
5.6.2	Datenerfassung	239
5.6.3	Datensätze	241
5.6.4	Parametrisierung der Geschwindigkeitsschätzer .	245
5.7	Auswertung und Diskussion	253
5.7.1	Datensatzvorbereitung	253
5.7.2	Genauigkeit	257
5.7.3	Rechenaufwand	270
5.7.4	Generalisierbarkeit	274
5.7.5	Effekte der Hyperparameter	283
5.8	Zusammenfassung	284
6	Orientierungsschätzung	287
6.1	Einführung	289
6.2	Problembeschreibung	292
6.2.1	Hintergrund	292
6.2.2	Formalisierung	293
6.2.3	Lösungsvorschlag	295
6.3	Fehleranalyse	295
6.3.1	Einführung	295

6.3.2	Messinfrastruktur	297
6.3.3	Drift-Akkumulationsstudie	298
6.4	Verwandte Arbeiten	301
6.4.1	Klassische Ansätze	302
6.4.2	Simultane Lokalisierung und Kartografie	302
6.4.3	Anwendungsspezifische Ansätze	303
6.4.4	PDR Ansätze	304
6.4.5	Ansätze des umgeleiteten Gehens	305
6.5	Beiträge	305
6.6	Methode	306
6.6.1	Signalverarbeitung	308
6.6.2	Datenvorverarbeitung	309
6.6.3	Einfluss von Bewegung auf IMU-Signale	313
6.6.4	Merkmalsauswahl	314
6.6.5	Immersive Anpassung der VR Ansicht	317
6.7	Experiment	318
6.7.1	Messinfrastruktur und -szenario	319
6.7.2	Datensatz	320
6.7.3	Konfiguration der Klassifikatoren	321
6.8	Auswertung	323
6.8.1	Auswertung der Klassifikatoren	324
6.8.2	Auswertung der Merkmale	324
6.8.3	Optimierung der Klassifikationsgenauigkeit	326
6.8.4	Auswertung der Anpassung der VR-Ansicht	327
6.8.5	Anwendbarkeit für reale Anwendungsfälle	329
6.9	Diskussion	331
6.10	Zusammenfassung	335
7	Posenschätzung	337
7.1	Einführung	340
7.2	Problembeschreibung	345
7.2.1	Datenlücken und asynchrone Datenströme	345
7.2.2	Nichtdeterministische Kalibrierung	346
7.2.3	Lösungsvorschlag	347
7.3	Verwandte Arbeiten	347
7.3.1	Rekursive probabilistische Verfahren	347
7.3.2	Datengetriebene Verfahren	349
7.4	Beiträge	350
7.5	Methoden	351
7.5.1	Datenvorverarbeitung	351

Inhaltsverzeichnis

7.5.2	Klassische modellbasierte Methode	353
7.5.3	Datengetriebene Methode	353
7.6	Experimente	356
7.6.1	Messinfrastruktur	358
7.6.2	Datensatz	358
7.6.3	Parametrisierung der Posenschätzer	359
7.7	Auswertung der modellbasierten PDR Methode	362
7.7.1	Metriken	363
7.7.2	Allgemeine Erkenntnisse	365
7.7.3	Auswirkung der Geschwindigkeit	367
7.7.4	Auswirkungen der Neukalibrierung	371
7.7.5	Zusammenfassung	371
7.8	Auswertung der datengetriebenen Methode	373
7.8.1	Vorabinformationen	374
7.8.2	Metriken	376
7.8.3	Auswirkung von Eingabevariationen	376
7.8.4	Auswirkung des Vorhersagehorizonts	379
7.8.5	Auswirkung der Sequenzlänge	383
7.8.6	Auswirkung plötzlicher Bewegungsänderungen .	386
7.8.7	Auswirkung unbekannter Trajektorienformen .	390
7.8.8	Auswirkung von Datenlücken	393
7.8.9	Zusammenfassung	395
7.9	Diskussion	396
7.10	Zusammenfassung	401
8	Zusammenfassung, Diskussion, Anwendungen und Ausblick	405
8.1	Zusammenfassung	405
8.1.1	Datengetriebene Positionsschätzung	406
8.1.2	Datengetriebene Geschwindigkeitsschätzung .	406
8.1.3	Datengetriebene Orientierungsschätzung . . .	407
8.1.4	Datengetriebene Fusion	408
8.2	Diskussion	409
8.2.1	Abhängigkeit von der Sequenzlänge	409
8.2.2	Datensatz Limitierungen	410
8.2.3	Funklokalisierung	410
8.2.4	Geschwindigkeits- und Schrittängenschätzung .	412
8.2.5	Orientierungsschätzung	414
8.2.6	Fusion multimodaler Sensordaten	417
8.3	Beiträge	419
8.3.1	Allgemeine Forschungsfragen	421

8.3.2	Forschungsfragen zur Funkpositionsschätzung	423
8.3.3	Forschungsfragen zur Geschwindigkeitsschätzung	426
8.3.4	Forschungsfragen zur Orientierungsschätzung	427
8.3.5	Forschungsfragen zur Fusion	429
8.3.6	Methodische Forschungsfragen	433
8.4	Anwendungen in der virtuellen Realität	439
8.4.1	Gangrekonstruktion	439
8.4.2	Simulatorkrankheit	440
8.4.3	Museum	442
8.5	Zukünftige Arbeiten	443
8.5.1	Unsicherheit und Erklärbarkeit	444
8.5.2	Automatisiertes maschinelles Lernen	445
8.5.3	Kontinuierliche Weiterbildung	446
8.5.4	Transfer Lernen	446
8.5.5	Lokalisierung	447
8.6	Letzte Worte	448
9	Eigene Veröffentlichungen, Patente und Qualifizierungsarbeiten	449
9.1	Konferenzbeiträge	449
9.2	Zeitschriftenartikel	449
9.3	Patente	449
9.4	Qualifizierungsarbeiten	449
Bibliographie		450

1 Einleitung

Es ist bekannt, dass die Positionsschätzung mit Verfahren zur Fußgängerkoppelnavigation (engl. Pedestrian Dead Reckoning, PDR) genaue Positionen und Orientierungen, sogenannte Posen bzw. Trajektorien einer Person schätzt, wenn Funksysteme keiner Mehrwegeausbreitung unterliegen, Trägheitssensoren an den Füßen des Benutzers angebracht sind und der Benutzer langsam geht. Ziele dieser Dissertation sind es daher, andere natürliche Positionen der Sensoren zu ermöglichen, wie der Hosentasche oder des Kopfes und darüber hinaus eine zuverlässige und genaue Posenschätzung bei hochdynamischen Bewegungen bereitzustellen, wie sie bspw. in Sportanwendungen und Anwendungen der virtuellen Realität stattfinden. Die Dissertation untersucht, ob ein Aufbrechen der Verarbeitungskette (engl. processing pipeline) durch sequenzbasierte Lernmethoden in möglichst viele verständliche, überwachbare und kontrollierbare Verarbeitungsschritte, nämlich Geschwindigkeit, Orientierung, Position und Fusion dieser Informationen, möglich ist, und genauere Posen liefert als klassische Filtermechanismen wie PDR und Kalman-Filter (KF) Verfahren.

1.1 Einführung

Lokalisierung wird typischerweise definiert als relative Änderung der Pose (Position und Orientierung) sowohl hinsichtlich der Translation als auch der Rotation zwischen aufeinanderfolgenden Sensordaten, gefolgt von der Integration dieser Posenänderungen hinsichtlich einer anfänglichen Pose in einem (absoluten) Weltkoordinatensystem, um die globale Pose abzuleiten. Die Verwendung von Sensormessungen zur Lokalisierung von Bewegungen wird seit Jahrzehnten untersucht. Bisher wurde eine große Anzahl komplizierter und mühsam parametrisierter analytischer Modelle entwickelt, die die Lokalisierung als ein Problem der Zustandsschätzung unter der Markovschen Annahme formulieren. M.a.W.: die aktuellen Systemzustände, z. B. Positionen, Geschwindigkeiten und Orientierungen, beziehen sich somit direkt auf Sensorbeobachtungen des vorherigen Systemzustands [29]. Um Bewegungsänderungen zu schätzen, integrieren fahrzeugfeste Trägheitsnavigationssysteme (engl. Strapdown Inertial Navigation System, SINS) Messungen von Trägheitssensoren direkt [387]. Im

1 Einleitung

Gegensatz dazu leiten PDR-Systeme eine Position ab, indem sie die Schrittänge und -richtung von Trägheitssensoren am Fuß abschätzen [170]. Um eine absolute Position zu schätzen, koppeln beide Verfahren ihre Ergebnisse mit exogenen Informationen, z. B. einer Kamera [259, 111] oder Funkposition [194, 195]. Andere Methoden wie visuelle trägeitsbasierte Odometrie (engl. Visual-Inertial Odometry, VIO) [140, 347] oder simultane Lokalisierung und Kartografie (engl. Simultaneous Localization and Mapping, SLAM) [95, 304] koppeln auch Trägheitssensoren und Kameras, um eine Position zu schätzen. Während diese Ansätze unter Laborbedingungen genaue Positionen liefern, wirken sich unvollständige Sensormessungen, ungenaue Systemmodellierung und komplexe Umgebungs dynamiken und Artefakte in realen Umgebungen sowohl auf die Genauigkeit als auch auf die Zuverlässigkeit der Lokalisierungssysteme aus. Die Verzerrung (engl. bias)¹, die Skalierung (engl. scaling)² und das Rauschen (engl. noise)³ kostengünstiger inertialer Messeinheiten-(engl. Inertial Measurement Unit, IMU-)Sensoren führen im Langzeitbetrieb zu großen Positionsabweichungen. Änderungen der Beleuchtung und fehlende oder beschädigte Sensordaten wirken sich auch negativ auf die Genauigkeit und Zuverlässigkeit der Systeme aus.

Im Rahmen langjähriger Forschungsaktivitäten an der fünften Generation des Mobilfunks (5G) wird immer noch untersucht, wie bestehende Funksensorik und IMU zu einer verbesserten Lokalisierungs technologie zusammengeführt werden können. Mithilfe dieser sogenannten Komplementärsensorik sollen exakte und zuverlässige Informationen über die aktuelle Pose dreidimensionale-(3D)-Position und -Orientierung im zeitlichen Verlauf eines zu verortenden Sensors bereitgestellt werden, um insbesondere in schwierigen Situationen oder ungünstigen Umgebungen zuverlässig akkurate und präzise Posen von Objekten zu schätzen. Aktuelle Methoden zur Posenschätzung basieren bspw. auf Bayes-Filtern wie z. B. KF und Partikel-Filter (PF) [60], nutzen (Kombinationen von) Funk-, z. B. Wi-Fi, GPS, und Ultrabreit-

¹Die Verzerrung (engl. bias) ist ein (konstanter) Versatz des Ausgabewerts vom Eingabewert. (Kapitel 6 erörtert Details).

²Der Skalierungsfaktor ist ein Multiplikator für ein Signal, der sich aus einem Verhältnis von Ausgang zu Eingang über den Messbereich zusammensetzt und über die Temperatur variiert. (Kapitel 5 erörtert Details).

³Das Rauschen (engl. noise) wird als zufällige Variation eines gemessenen Ausgabewerts klassifiziert. (Kapitel 5 erörtert Details).

band (engl. Ultra-wideband, UWB) und Beschleunigungs-, Gyroskop-, Magnetometer- und Barometer-Sensorik und ermöglichen die Bestimmung der Pose von z. B. Gabelstaplern oder Automobilen. Besteht direkter Sichtkontakt zwischen Sensorik, z. B. Funkempfänger, und dem zu verortenden Objekt, z. B. Funksender, lassen sich absolute Positionen mittels Ankunftszeitdifferenz (engl. Time-Difference-of-Arrival, TDoA) Verfahren zuverlässig schätzen und zusammen mit Orientierungsschätzungen der Inertialsensorik genaue (Mittlerer absoluter Fehler, $MAE \leq 0.20$ m, engl. Mean Absolute Error) Posen ermitteln. Typischerweise beschreiben Bayes-Filter einfache Bewegungsformen von Fahrzeugen, sogenannte lineare Bewegungen, physikalisch korrekt und repräsentieren den Stand der Technik. Zur absoluten Orientierungsbestimmung durch Magnetometer (*Mag.*) und relativen Positionsbestimmung durch Beschleunigungssensor und Gyroskop werden IMUs verwendet, die heute in jedem modernen Mobiltelefon verfügbar sind. Eine Fusion von funkbasierter Ortung zur absoluten Positionsbestimmung mit Inertialsensorik stabilisiert die Posenschätzung und erhöht ihre Genauigkeit.

Herausforderungen

Allerdings leidet die Genauigkeit solcher Verfahren vor allem in Innenräumen, da hier häufig abschirmende Objekte und Hindernisse, die die Sicht (teilweise) verdecken, wie z. B. Wände oder Regale, die Funksignale umlenken, streuen und dämpfen. Die Ausbreitung der Funksignale über mehrere Wege (Multipfadausbreitung) ist somit nicht eindeutig vorhersagbar. Und somit liefern laufzeitbasierte Schätzverfahren, wie TDoA, fehlerhafte Ergebnisse. Durch mathematisches Integrieren der Beschleunigung oder Drehrate kann die relative Geschwindigkeit oder Orientierung und die relative Entfernungsänderung bestimmt werden. Jedoch führen insbesondere verrauschte Beschleunigungsdaten mit jeder Integration zu einem stark akkumulierenden Fehler (doppelte Integration führt zu einem Distanzfehler von ca. 1 km/s). Die Schätzung der absoluten Orientierung um die Gier-Achse mittels Magnetometer wird durch metallische Gegenstände in der Umgebung gestört. Des Weiteren ist es schwierig, mittels Bayes-Filtern die menschliche Bewegung, die sich aus hochgradig nichtlinearen Bewegungskomponenten zusammensetzt, physikalisch korrekt zu beschreiben [340]. Aktuelle Forschungsarbeiten schätzen nichtlineare Bewegungszustände erfolgreich, ohne komplexe Bayes-Filter zu erstellen, mithilfe von neuronalen

1 Einleitung

Netzen (engl. Neural Network, NN). Sogenannte rekurrente neuronale Netze (engl. Recurrent Neural Network, RNN) bilden Zeitreihendaten, wie z. B. Funk- und Inertial-Sensorik, auf einen Zustand, wie z. B. die Pose, der der Wahrheit entspricht, korrekt und zuverlässig ab. Allerdings benötigen diese Methoden sehr viele Daten, um Bewegungen möglichst generell abzubilden. Datengetriebene Ende-zu-Ende Lernverfahren wie RNNs sind deutlich leichter zu modellieren als KF. Sie lernen durch Lang-Kurzzeitgedächtnis Zellen (engl. Long Short-Term Memory, LSTM) aus konsekutiven Bewegungsdaten sowohl kurz- als auch langfristige Abhängigkeiten zwischen den realen und gemessenen Posen-(Positions- und Orientierungs-)daten zu unterscheiden. Allerdings ist unklar, wie sich unterschiedliche Bewegungskomponenten, nämlich relative oder absolute Position, Beschleunigung, Geschwindigkeit, Distanz und Orientierung, die als Eingabe für ein RNN dienen, auf die zu lernende Pose und deren Genauigkeit auswirken. Ferner ist unbekannt, wie die relativen und absoluten Komponenten die Generalisierungsfähigkeit der RNN-Modelle beeinflussen.

Lösungsansatz

Die jüngsten Forschungsergebnisse des maschinellen und tiefen Lernens (engl. Machine Learning, ML; Deep Learning, DL) haben eine Vielzahl bemerkenswerter Erfolge in der Bild- und Sprachverarbeitung erzielt [255, 423, 176]. Im Gegensatz dazu wurde die Koppelkalibration mit Trägheits- und Funksensoren weniger gründlich untersucht. Ebenso ist es schwierig, bekannte datengetriebene Methoden, die bspw. für die Bild- oder Sprachverarbeitung entwickelt wurden, direkt auf das multimodale Lokalisierungsproblem anzuwenden. Insbesondere ist ungeklärt, welche Auswirkungen datengetriebene Methoden auf die Zuverlässigkeit und Genauigkeit von Positionen, Geschwindigkeiten, Orientierungen und die Rekonstruktion der Trajektorie über längere Zeiträume haben. Es ist unbekannt, wie datengetriebene Methoden, die in einer Domäne wie Fußgänger oder kreisende Bewegungen trainiert wurden, verallgemeinern auf eine neue Domäne wie rennende Fußgänger oder zufällige Bewegungsformen. Obwohl die klassische Fusion von unimodalen Sensorsignalen eine lange Forschungsgeschichte hat, ist unklar, wie datengetriebene Methoden eine zuverlässige und präzise Fusion von multimodalen Sensorsignalen ermöglichen können. Aufgrund dieser Unklarheiten ist

die Hauptmotivation für diese Dissertation:

Wie können sequenzbasierte Lernmethoden die Trajektorien von Fußgängern in dynamischen Anwendungen mithilfe multimodaler Sensordatenströme vorhersagen, um eine genaue und robuste Lokalisierung zu ermöglichen, ohne auf komplexe, handgefertigte Modelle zurückgreifen zu müssen?

Kernziel dieser Dissertation ist somit die Untersuchung der Auswirkungen datengetriebener Verfahren und verschiedenen Sensordatenströme von lose platzierten Sensoren auf die Genauigkeit der Schätzung der menschlichen Posen in dynamischen Situationen. Die Absolutgenauigkeit der erzielten Ergebnisse wird mit Filtermethoden des Stands der Technik und Wissenschaft verglichen. Daher werden die Auswirkungen von datengetriebenen Verfahren entlang einer typischen Verarbeitungskette zur Lokalisierung mit Funk- und Trägheitssensorik untersucht. Die Verarbeitungskette ist lose gekoppelt, sodass jedes datengetriebene Verfahren problemlos ausgetauscht werden kann. Die Positionen werden aus Ankunftszeiten eines Funksystems in Innenräumen geschätzt. Diese werden an eine parallele Orientierungsschätzung und eine nachgelagerte Posenschätzung weitergeleitet. Die Orientierungsschätzung klassifiziert zuverlässig Bewegungsmuster, um eine langzeitstabile Orientierung auf Basis des Richtungsvektors zweier Funkpositionen für die Posenschätzung bereitzustellen. Parallel wird aus rotationsinvarianten verrauschten Trägheitsmessungen genaue Geschwindigkeiten geschätzt, die ebenfalls an die Posenschätzung weitergeleitet werden. Die Posenschätzung fusioniert Positionen, Geschwindigkeiten und Orientierungen, um eine genaue Trajektorie vorherzusagen. Die in dieser Arbeit vorgeschlagene Verarbeitungskette liefert genauere und robustere Schätzungen als bekannte Verfahren des Stands der Technik, auch in dynamischen Bewegungen mit verrauschten Trägheitssensoren und Funkumgebungen mit Mehrwegeausbreitung. Im Rahmen einiger Probandenstudien, mit über 200 Menschen und einem Roboter verteilt auf 4 Studien, werden Mess- und Referenzdaten, z. B. verschiedene Bewegungspfade und Richtungsänderungen bspw. Geraden, Kreise, Ellipsen, Zufall, in einer Testhalle (45×35 m) mit unterschiedlichen Geschwindigkeiten, z. B. langsames und schnelles Laufen, Joggen und Rennen, aufgezeichnet.

1.2 Problembeschreibung

Dieser Abschnitt beschreibt die wichtigsten Herausforderungen, mit denen sich diese Dissertation befasst. Zunächst werden die Schwierigkeiten der Verarbeitungskette von rohen Messsignalen bis zur Rekonstruktion der Trajektorie diskutiert (Abschnitt 1.2.1). Anschließend werden die Probleme von verrauschten Trägheitsmessungen (Abschnitt 1.2.2) und der Mehrwegeausbreitung in Funksystemen (Abschnitt 1.2.3) in der Verarbeitungskette diskutiert. Anschließend werden die Probleme bei der Gierwinkelschätzung (Abschnitt 1.2.4) und schließlich die Schwierigkeiten beim Zusammenführen von Informationen von verschiedenen Sensoren (Abschnitt 1.2.5) erörtert.

1.2.1 Stand der Technik

Modellgetriebene Verfahren nach dem Stand der Technik, bspw. KF, können keine sensorspezifischen Fehler modellieren, bspw. Temperaturabhängigkeiten der Inertialsensorik und Mehrwegpfade eines drahtlosen Sensorsystems und liefern somit ungenaue Positionen. Ende-zu-Ende-Verfahren, die Funk- und Inertialsensorik koppeln, müssen in einem gemeinsamen Weltkoordinatensystem arbeiten, schätzen somit absolute Positionen und lassen sich daher nicht auf andere Koordinatensysteme verallgemeinern. Z. B. liefert ein datengetriebenes Verfahren, das Funk- und Inertialsensorik verwendet, um eine Person in einem Innenraum zu positionieren, in einem anderen Raum keine genaue Position. Oder eine Person, die entlang einer Einkaufsstraße läuft, bewegt sich völlig anders, wenn sie auf einem Fußballfeld rennt. In Kapitel 2 werden die Probleme herkömmlicher Verfahren ausführlich diskutiert.

Die Methoden der Dissertation adressieren diese Probleme und zeigen, wie datengetriebene Verfahren parallel und sequentiell miteinander verbunden werden, um die Pose eines Menschen genauer und robuster als die Verfahren des Stands der Technik zu schätzen und Rechenressourcen einzusparen. Dazu werden die grobe Ausgangsposition (in Kapitel 4), die feinkörnige präzise Geschwindigkeit (in Kapitel 5) und die präzise und stabile Orientierung (in Kapitel 6) zu einer Pose fusioniert (in Kapitel 7).

1.2.2 Verrauschte Trägheitsmessungen

Die Aufteilung der Gravitationskraft, die auf die drei Achsen eines Beschleunigungsmessers (engl. accelerometer) verteilt ist, kann nicht genau bestimmt werden, da die Ausrichtung des Sensors insbesondere in Anwendungen mit hohen Bewegungsdynamiken, z. B. schnelle und abrupte Richtungs- und Bewegungsänderungen, zu keinem Zeitpunkt eindeutig ist. Die Beschleunigung kann somit nicht zur Abschätzung der Orientierung um die Körperachse, sogenannter Gierwinkel verwendet werden. Da die Gravitationskraft nicht eindeutig gemessen werden kann, kann die lineare Beschleunigung nicht fehlerfrei bestimmt werden. Dieser Fehler und das thermische Rauschen werden zur Bestimmung der Geschwindigkeit und Entfernung ungefiltert in die mathematische Einzel- und Doppelintegration übertragen und liefern einen explosionsartig zunehmenden Positionsfehler mehrerer Kilometer pro Sekunde. In Kapitel 5 werden die Probleme ausführlich erläutert.

Die Methoden der Dissertation adressieren diese Probleme und zeigen, wie datengetriebene Verfahren die Geschwindigkeit aus verrauschten Trägheitsmessungen (in Kapitel 5) genauer und robuster als die Verfahren des Stands der Technik schätzen können.

1.2.3 Mehrwegeausbreitung in Funksystemen

Im Gegensatz dazu leiden Funksysteme unter Mehrwegeausbreitung, wenn Objekte zwischen einem Sender und einem Empfänger deren Sicht behindern oder sogar vollständig verdecken. In diesen Fällen kann die Ankunftszeit eines Funksignals an den Empfängern nicht mehr fehlerfrei bestimmt werden, und klassische Multilaterationsschätzer und die Methode der kleinsten Quadrate (engl. method of Least-Squares, LS) ergeben fehlerhafte Positionen. Darüber hinaus können herkömmliche Methoden des Stands der Technik die Bewegung der Objekte, die die Sichtverbindung (engl. Line of sight, LoS) stören, nicht von der eigentlichen Bewegung eines Senders getrennt werden, sodass auch hier die Ankunftszeiten verschoben werden. Aufgrund der hohen Bandbreite und Frequenznutzung ist die Sendeleistungsdichte eines UWB-Systems auf -41.3 dB m/MHz begrenzt, um Interferenzen zwischen Systemen zu vermeiden. Diese Einschränkung begrenzt die Reichweite des Systems auf 300 m in LoS und 40 m bei Nicht-Sichtverbindung (engl. Non-Line-Of-Sight, NLoS). Aus diesem Grund wird in dieser Dissertati-

on ein speziell entwickeltes System verwendet [149]. In Kapitel 3.3 und 4 werden die Probleme ausführlich erläutert.

Die Methoden der Dissertation adressieren diese Probleme und zeigen, wie datengetriebene Verfahren die Position aus fehlerbehafteten Ankunftszeiten (in Kapitel 4) genauer und robuster als die Verfahren des Stands der Technik schätzen können.

1.2.4 Gier-Winkel Schätzung

Wenn Sensoren an verschiedenen Koordinaten- und Rotationssystemen wie Körper und Kopf angebracht sind, müssen die Koordinatensysteme gemäß einem (globalen) Weltkoordinatensystem ausgerichtet werden, um eine eindeutige gemeinsame Pose aus den unterschiedlichen Sensoren zu schätzen. Da Funk- und Trägheitssensoren spezifische Fehler aufweisen, ist dies mit Verfahren des Stands der Technik unmöglich. Dennoch, während die Bestimmung der Orientierung eines Sensors am Kopf nicht fehlerfrei möglich ist, ist die Orientierungsschätzung eines Sensors in der Hosentasche trivial, da sich die Positionsrichtung, d. h. die Bewegungsrichtung des Sensors und die Ausrichtung des Sensors in der Hosentasche nicht unterscheiden. Wenn der Sensor allerdings nicht fest auf dem zu lokalisierenden Objekt platziert und befestigt ist, d. h. die Ausrichtung der beiden Koordinatensysteme abweicht, können die Ausrichtung und damit die Gravitationskraft und dadurch die lineare Beschleunigung und somit die Geschwindigkeit nicht genau bestimmt werden. Um dieses Problem zu adressieren, werden im Rahmen der Dissertation Positionen und Geschwindigkeiten mit Sensoren in der Hosentasche geschätzt, da dies dem typischen Alltagsgebrauch von Mobiltelefonen entspricht. Ferner wird die Orientierung explizit sowohl am Kopf als auch am Körper beobachtet, um die Auswirkungen unterschiedlicher Koordinaten- und Rotationssysteme auf die vorgeschlagenen Schätzverfahren zu untersuchen. In Kapitel 6 werden die Probleme ausführlich erläutert.

Die Methoden der Dissertation adressieren diese Probleme und zeigen, wie datengetriebene Verfahren die Orientierung aus verrauschten Trägheitsmessungen (in Kapitel 6) genauer und robuster als die Verfahren des Stands der Technik schätzen können.

1.2.5 Multimodale Informationsfusion

Verschiedene Umgebungseinflüsse, wie etwa Sensorinterferenzen, darunter (zeitweilige) Signalverluste, -dämpfungen, -dispersionen, -reflektionen, -drift und Sensorrauschen führen bei Koppelnavigation (engl. Dead Reckoning, DR) zu einer Orientierungs- bzw. Positionsabweichung. Gegenwärtige Fusionsmethoden leiden unter mehreren Problemen. Sensoren liefern Messwerte mit unterschiedlichen und variablen Datenraten, die von vordefinierten Filtern nicht verarbeitet werden können, da modellgetriebene Ansätze fordern, dass verschiedene Sensoren zeitsynchronisiert sind und Sensordatenströme nicht ausfallen. Moderne Funk- und Trägheitssysteme sind lose gekoppelt, die Daten werden im System übertragen, das zu nicht deterministischen Verzögerungen führt, die durch ein Modell nicht beschrieben werden. Darüber hinaus ist unklar, wie und wann individuelle Schätzfehler optimalerweise korrigiert werden. Denn ein naives modellbasiertes Einstellen bspw. der aktuellen Positionsschätzung des Trägheitssensorsystems auf die aktuelle Funkposition kann zu schwerwiegenden Sprüngen der korrigierten Position führen. Im schlimmsten Fall könnte die Schätzung des Trägheitssensorsystems genauer sein und die Funkposition würde einen Positionsfehler verursachen.

Die Methoden der Dissertation adressieren diese Probleme und zeigen, wie datengetriebene Verfahren multimodale unsynchronisierte Sensordatenströme optimal fusionieren (in Kapitel 7), um Posen und Trajektorien genauer und robuster als die Verfahren des Stands der Technik zu rekonstruieren.

1.3 Beiträge und Forschungsfragen

Die Kapitel 4 bis 7 beschreiben die Kernmethoden und Beiträge der Arbeit und definieren, analysieren und diskutieren Forschungsfragen und Beiträge im Detail. Dieser Abschnitt listet die Kernforschungsfragen der Dissertation auf, die in Kapitel 8.3 diskutiert und beantwortet werden. Die Forschungsfragen werden in Abschnitt 1.2 motiviert.

Die Methoden 4 bis 7 befassen sich mit den folgenden Problemen des Stands der Technik. SINS akkumulieren Sensorrauschen, indem sie Beschleunigungen zweimal integrieren, um eine Entfernung zu erhalten. PDR-Methoden erfordern Sensoren an den Füßen, um diese akkumulierten Fehler zu kalibrieren und zurückzusetzen, wenn die Füße

1 Einleitung

mit dem Boden in Kontakt kommen. Funklokalisierungssysteme leiden unter Mehrwegeausbreitung, die mit herkömmlichen Methoden nicht erfasst und beschrieben werden können. Eine genaue Schrittängenschätzung oder Geschwindigkeitsschätzung erfordert Sensoren an den Füßen des Benutzers. Der Benutzer darf nicht laufen, da sich der Gang ändert, kein Bodenkontakt für die Schritterkennung vorhanden ist und integrale Verfahren mit Rücksetzmechanismen nicht korrigiert werden können. Dies führt zu schwerwiegenden Fehlern bei der Geschwindigkeitsschätzung. Zusätzlich zur Schrittänge kann die Schrittrichtung langfristig nicht richtig geschätzt werden. Deshalb stellen sich die folgenden Forschungsfragen (engl. Research Questions, RQ):

RQ1: *Wie schnell und wie groß häufen sich Orientierungs- und Abstandsfehler in einem Trägheitsnavigationssystem an, wenn aktuelle MEMS-Trägheitssensoren⁴ nach dem Stand der Technik (modell- und datengetrieben) verwendet werden?*

Diese Forschungsfrage wird in Kapitel 8.3.1 ausführlich diskutiert.

RQ2: *Können datengetriebene Verfahren die Positionsschätzung eines dynamisch bewegten Funksenders aus verrauschten Ankunftszeitmessungen verbessern?*

Diese und weitere Forschungsfragen zur Funkpositionsschätzung werden in Kapitel 8.3.2 ausführlich diskutiert.

RQ3: *Können datengetriebene Verfahren die Geschwindigkeitsschätzung eines Fußgängers aus verrauschten MEMS-Trägheitsmessungen verbessern?*

Diese und weitere Forschungsfragen zur Geschwindigkeitsschätzung werden in Kapitel 8.3.3 ausführlich diskutiert.

Herkömmliche Verfahren liefern ungenaue Orientierungen, insbesondere bei sich schnell ändernden Bewegungen, magnetischen Störquellen oder lose platzierten Sensoren. Zusammen mit der falschen Schrittänge werden Trajektorien rekonstruiert, die nach kurzer Zeit von der Wahrheit abweichen.

⁴Mikrosystem, engl. Micro-Electro-Mechanical Systems, MEMS.

Heute tragen die meisten Menschen unbemerkt viele Sensoren an ihrem Körper, die die Bewegung in verschiedenen Rotations- und Koordinatensystemen aufzeichnen. Um die Informationen zusammenzuführen, müssen die Systeme kalibriert werden. Dies ist für herkömmliche Methoden in dynamischen Situationen unmöglich, da natürliche Bewegungsformen nicht deterministisch beschrieben werden können. Deshalb stellt sich die folgende Forschungsfrage:

RQ4: *Können datengetriebene Verfahren die Orientierungsschätzung eines Fußgängers aus verrauschten MEMS-Trägheitsmessungen verbessern?*

Diese und weitere Forschungsfragen zur Orientierungsschätzung werden in Kapitel 8.3.4 ausführlich diskutiert.

Die Zusammenführung mehrerer Sensordatenströme von einem und mehreren Sensoren ist ebenso schwierig. Um jeden Sensor zusammenzuführen, muss er auf komplexe, deterministische Weise beschrieben werden. Viele Sensorkonfigurationen können nur angenähert werden und entsprechen nicht der Realität. Bereits kleinste Abweichungen führen über lange Zeiträume zu schwerwiegenden Fehlern. Deshalb stellt sich die folgende Forschungsfrage:

RQ5: *Können datengetriebene Verfahren die Fusion verschiedener Sensordatenströme (MEMS-Trägheitssensor und Funksystem) verbessern?*

Diese und weitere Forschungsfragen zur Fusion werden in Kapitel 8.3.5 ausführlich diskutiert.

Ein weiterer Schwerpunkt liegt im Umgang mit Stör- und Rauschquellen. Herkömmliche Methoden können nichtdeterministisches, stochastisches Rauschen nicht beschreiben, da zufälliges Rauschen theoretisch nur angenähert werden kann. Darüber hinaus können zufällige Irrfahrten aufgrund temperaturabhängiger Schwankungen der Messungen auch bei herkömmlichen Methoden nur annähernd beschrieben werden. Herkömmliche modellgetriebene Verfahren haben Schwierigkeiten bei der Beschreibung nichtlinearer Beziehungen und datengetriebenen Verfahren fehlen Trainingsdaten. Deshalb stellt sich die folgende Forschungsfrage:

RQ6: Welche Vorteile bieten datengetriebene Verfahren zur Verarbeitung von verrauschten Zeitreihendaten?

Diese und weitere Forschungsfragen zu den vorgeschlagenen Methoden werden in Kapitel 8.3.6 ausführlich diskutiert.

1.4 Abgrenzung der veröffentlichten Beiträge

In diesem Abschnitt werden bereits veröffentlichte Forschungsarbeiten aufgeführt, die im Rahmen der Dissertation entstanden sind. Die Abschnitte 1.4.1 bis 1.4.4 grenzen veröffentlichte Forschungsergebnisse ab, die im Rahmen der Dissertation erforscht wurden und als Grundlage für diese Arbeit dienen. Abschnitt 1.4.5 grenzt veröffentlichte Forschungsarbeiten ab, die im Rahmen der Dissertation durch Forschungskooperationen, gemeinsame Veröffentlichungen und den Transfer erworbener Kenntnisse und Fähigkeiten entstanden sind.

Für jede Veröffentlichung werden die wesentlichen Kernbeiträge einzelner Autoren detailliert erörtert.⁵ Zusätzlich werden Anmerkungen zu den Beiträgen aufgeführt und die Beiträge in die inhaltliche Struktur der Dissertation eingeordnet. Bei Veröffentlichungen mit Erstautorschaft von anderen als T. Feigl (Autor der vorliegenden Dissertation) werden die genauen inhaltlichen Beiträge der einzelnen Autoren beschrieben und von denen anderer abgegrenzt.

1.4.1 Beiträge zur Positionsschätzung

[127] Tobias Feigl, Thorsten Nowak, Michael Philippse, Thorsten Edelhäuser und Christopher Mutschler. Recurrent Neural Net-

⁵**Aufschlüsselung der Kernbeiträge:** Konzeptualisierung (m.a.W. Idee und Konzept der Forschungsarbeit), Methodik (m.a.W. Entwurf und Entwicklung der Algorithmen), Software (m.a.W. Implementierung der Methoden), Validierung (m.a.W. Validierung, Evaluation und Studiendurchführung der Forschungsergebnisse), Formale Analyse (m.a.W. formale Beweisführung), Nachforschung (m.a.W. Einbettung in den Stand der Technik und abwiegen der Forschungshöhe), Ressourcen (m.a.W. Bereitstellung, Inbetriebnahme und Wartung technischer Komponenten), Datenkuration (m.a.W. sammeln, aufzeichnen und aufbereiten relevanter Daten), Verschriftlung (m.a.W. Entwurf und Ausarbeitung der Forschungsergebnisse in Form einer wissenschaftlichen Veröffentlichung), Rezension (m.a.W. inhaltliche, organisatorische und fachliche Verbesserungsvorschläge), Visualisierung (m.a.W. Tabellen, Bild- und Videomaterial), Aufsicht (m.a.W. wissenschaftliche Betreuung), Projektverwaltung (m.a.W. Projektleiter- und verantwortlicher) und Akquisition der Finanzierung.

works on Drifting Time-of-Flight Measurements. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–8. Nantes, France, 2018.

Anmerkungen: Die wissenschaftlichen Beiträge zu dieser Arbeit stammen von T. Feigl. Das Konzept und die Methoden wurden von T. Feigl unter der Aufsicht von T. Nowak (Bayes-Filter) und T. Edelhäußer (rekurrente neuronale Netze) erarbeitet. T. Nowak und T. Edelhäußer validierten das Konzept und die Methoden von T. Feigl. Die Arbeit basiert auf Daten, die im Rahmen einer Qualifizierungsarbeit von T. Feigl unter der Aufsicht von T. Edelhäußer (betreute das System zur Datenaufzeichnung) gesammelt wurden. T. Feigl säuberte und validierte die gesammelten Datensätze und analysierten formell deren Plausibilität. Die Grundstruktur der Software zur Datenerfassung, -verarbeitung und -auswertung wurde von T. Edelhäußer bereitgestellt. Die Arbeit wurde von M. Philippse und C. Mutschler beaufsichtigt und rezensiert.

Beitrag: Diese Arbeit liefert die Grundlage für die funkbasierte Lokalisierung, die im Rahmen der Dissertation für die Lokalisierung und die Erstkalibrierung und Neukalibrierung der Methode verwendet wird (siehe Kapitel 4).

- [134] Tobias Feigl, Sebastian Kram, Ernst Eberlein und Christopher Mutschler. Robust ToA-Estimation using Convolutional Neural Networks on Randomized Channel Models. *IEEE Trans. on Signal Processing (IEEE TSP)*, 68(8234):1–13, 2020 (submitted).

Anmerkungen: Die wissenschaftlichen Beiträge zu dieser Arbeit stammen von T. Feigl. Das Konzept und die Methoden wurden von T. Feigl unter der Aufsicht von E. Eberlein (Konzept und Simulation) erarbeitet. Die zugrunde liegenden Daten wurden von T. Feigl unter Anweisung von E. Eberlein (betreut den Simulator) erhoben. T. Feigl säuberte und validierte die gesammelten Datensätze und E. Eberlein und S. Kram analysierten formell deren Plausibilität. Die Grundstruktur der Software zur Datenerfassung, -verarbeitung und -auswertung wurde von T. Feigl (Methoden zur Schätzung und Auswertung der Ankunftszeit, engl. Time of Arrival, ToA) und E. Eberlein (Simulation) bereitgestellt. Die Arbeit wurde von C. Mutschler beaufsichtigt und rezensiert.

Beitrag: Diese (unveröffentlichte) Arbeit liefert Grundlagen für die Funklokalisierung (siehe Kapitel 3) und beschreibt die Gewin-

nung von ToA Werten, die in Kapitel 4 zur Positionsschätzung verwendet werden. Im Rahmen dieser Forschungsarbeit wurde auch ein Patent [132] eingereicht.

- [247] Sebastian Kram, Maximilian Stahlke, Tobias Feigl, Jochen Seitz und Jörn Thielecke. UWB Channel Impulse Responses for Positioning in Complex Environments: A Detailed Feature Analysis. *Sensors J.*, 24(2):1–23, 2019.

Anmerkungen: Die wissenschaftlichen Beiträge zu dieser Arbeit stammen hauptsächlich von S. Kram und M. Stahlke. S. Kram und M. Stahlke lieferten die Grundidee und das Konzept basierend auf einer früheren Veröffentlichung von S. Kram et al. [411]. T. Feigl übertrug Wissen aus dieser Dissertation in die bestehende Forschungsarbeit und erweiterte das Konzept von S. Kram und M. Stahlke um einen hierarchischen Klassifizierungsansatz, der maschinelles Lernen verwendet, um die Mehrwegesignalqualität zwischen Funksendern auf verschiedenen Abstraktionsebenen der Lokalisierungsgenauigkeit zu bewerten (Abschnitte 6, 6.1 und 6.2), um so robustere und genauere Ergebnisse zu liefern (Tabellen 2 und 3). S. Kram und M. Stahlke stellten die grundlegende Implementierung und Software bereit. T. Feigl erweiterte diese um eine hierarchische datengetriebene Klassifikation (siehe Grafik 12a in Abschnitt 6.1). Die Validierung, formale Analyse, Nachforschung und Visualisierung (Grafiken 12a und 13) erfolgte somit von S. Kram (Basis), M. Stahlke (Basis) und T. Feigl (Erweiterung) für den jeweiligen Teil zu quasi gleichen Anteilen. Die Arbeit wurde von J. Thielecke und J. Seitz beaufsichtigt und rezensiert.

Beitrag: Die Erkenntnisse dieser Arbeit (Qualitätsmaß für ToA Werte) könnten als Optimierungsmethode zur Positionsschätzung verwendet werden (Vorschlag in Kapitel 8.5.5), um die Genauigkeit der ToA Schätzer mit kleinsten Fehlerquadraten, bspw. Levenberg-Marquardt [6], oder datengetriebene Verfahren (siehe Kapitel 4) zu unterstützen, um genauere Positionen zu ermitteln.

1.4.2 Beiträge zur Geschwindigkeitsschätzung

- [128] Tobias Feigl, Sebastian Kram, Philipp Woller, Ramiz H. Siddiqui, Michael Philippse und Christopher Mutschler. A Bidirectional LSTM for Estimating Dynamic Human Velocities from a Single

IMU. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–8. Pisa, Italy, 2019.

Anmerkungen: Die wissenschaftlichen Beiträge zu dieser Arbeit stammen von T. Feigl. Das Konzept und die Methoden wurden von T. Feigl erarbeitet. Die Arbeit basiert auf Daten, die im Rahmen einer Forschungsstudie von T. Feigl, S. Kram, P. Woller und R. H. Siddiqui gesammelt wurden. T. Feigl stellte die Implementierung der Anwendung zur Datenaufzeichnung bereit. P. Woller reinigte die rohen Daten und kontrollierte die Datensätze. T. Feigl, S. Kram, P. Woller und R. H. Siddiqui verwalteten die Datenkuration und validierten ihre Plausibilität. Die Arbeit wurde von M. Philippsen und C. Mutschler beaufsichtigt und rezensiert.

Beitrag: Diese Arbeit dient v. A. als Grundlage für die Geschwindigkeitsschätzung in Kapitel 5 und beschreibt, wie die Kombination aus faltenden neuronalen Netzen und rekurrenten neuronalen Netzen Geschwindigkeiten schnell und genau aus rohen Beschleunigungsmessungen schätzt.

- [131] Tobias Feigl, Sebastian Kram, Philipp Woller, Ramiz H. Siddiqui, Michael Philippsen und Christopher Mutschler. RNN-aided Human Velocity Estimation from a Single IMU. *Sensors J.*, 13(4):1–31, 2020.

Anmerkungen: Die wissenschaftlichen Beiträge dieser Arbeit liegen vollständig bei T. Feigl. Die Arbeit basiert auf der Veröffentlichung von T. Feigl et al. [128], erweitert diese und liefert neue Erkenntnisse. Das Konzept und die Methoden wurden von T. Feigl und S. Kram (betreute die Entwicklung und Optimierung des KFs) erarbeitet. Die Daten wurden im Rahmen einer Forschungsstudie [128] erhoben. Die Datenkuration wurde durch T. Feigl durchgeführt. S. Kram, P. Woller und R. H. Siddiqui validierten die Ergebnisse. Die Arbeit wurde von M. Philippsen und C. Mutschler beaufsichtigt und rezensiert.

Beitrag: Diese Arbeit dient v. A. als Grundlage für die Geschwindigkeitsschätzung in Kapitel 5 und beschreibt wie die Kombination aus tiefen Resteinheiten- und bidirektionalen rekurrenten neuronalen Netzen Geschwindigkeiten hoch genau aus rohen Beschleunigungsmessungen schätzt. Der Fokus der Arbeit liegt auf dem tieferen Verständnis räumlicher und zeitlicher Merkmale in NN und deren Auswirkung auf die Schätzgenauigkeit.

1.4.3 Beiträge zur Orientierungsschätzung

- [125] Tobias Feigl, Christopher Mutschler und Michael Philippse. Human Compensation Strategies for Orientation Drifts. In *Proc. Intl. Conf. Virtual Reality and 3D User Interfaces (IEEE VR)*, Seiten 409–414. Reutlingen, Germany, 2018.

Anmerkungen: Die wissenschaftlichen Beiträge zu dieser Arbeit stammen von T. Feigl. Das Konzept und die Methoden wurden von T. Feigl erarbeitet. Die Arbeit basiert auf Daten, die von T. Feigl gesammelt wurden. T. Feigl säuberte und validierten die gesammelten Datensätze und analysierten formell deren Plausibilität. Die Grundstruktur der Software zur Datenerfassung, -verarbeitung und -auswertung wurde von T. Feigl bereitgestellt. Die Arbeit wurde von M. Philippse und C. Mutschler beaufsichtigt und rezensiert.

Beitrag: Diese Arbeit liefert Grundlagen für Bewertungstechniken und Metriken, die im Rahmen der Dissertation verwendet werden (siehe Kapitel 3) und beschreibt die Fehlereigenschaften und deren Entstehung und Entwicklung von trägheitsbasierten Sensoren (siehe Kapitel 6.3). Darüber hinaus zeigt diese Arbeit die Zusammenhänge zwischen menschlicher Bewegung und den Fehlern der Orientierungsschätzung nach dem Stand der Technik (siehe Kapitel 2) und dient somit als Motivation für die Orientierungsschätzung mittels datengetriebener Bewegungsklassifikation (siehe Kapitel 6) der vorliegenden Dissertation. Im Rahmen dieser Forschungsarbeit wurde auch ein Patent [121] durch T. Feigl verfasst und teil-veröffentlicht.

- [126] Tobias Feigl, Christopher Mutschler und Michael Philippse. Head-to-Body-Pose Classification in No-Pose VR Tracking Systems. In *Proc. Intl. Conf. Virtual Reality and 3D User Interfaces (IEEE VR)*, Seiten 545–546. Reutlingen, Germany, 2018.

Anmerkungen: Die wissenschaftlichen Beiträge zu dieser Arbeit stammen von T. Feigl. Das Konzept und die Methoden wurden von T. Feigl erarbeitet. Die Arbeit basiert auf Daten, die von T. Feigl gesammelt wurden. T. Feigl säuberte und validierte die gesammelten Datensätze und analysierte formell deren Plausibilität. Die Grundstruktur der Software zur Datenerfassung, -verarbeitung und -auswertung wurde von T. Feigl bereitgestellt.

Die Arbeit wurde von M. Philippsen und C. Mutschler beaufsichtigt und rezensiert.

Beitrag: Diese Arbeit liefert die Grundlage für die funk- und trägheitsbasierte Orientierungsschätzung (siehe Kapitel 6) und dient als Basis der Erstkalibrierung und Neukalibrierung der Sensororientierung mittels datengetriebener Bewegungsklassifikation in Kapitel 6. Die Ergebnisse der Methode fließen in die Posenschätzung 7 ein. Im Rahmen dieser Forschungsarbeit wurde auch ein Patent [123] durch T. Feigl verfasst und teil-veröffentlicht.

- [124] Tobias Feigl, Christopher Mutschler und Michael Philippsen. Supervised Learning for Yaw Orientation Estimation. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–8. Nantes, France, 2018.

Anmerkungen: Die wissenschaftlichen Beiträge zu dieser Arbeit stammen von T. Feigl. Das Konzept und die Methoden wurden von T. Feigl erarbeitet. Die Arbeit basiert auf Daten, die von T. Feigl gesammelt wurden. T. Feigl säuberte und validierte die gesammelten Datensätze und analysierten formell deren Plausibilität. Die Grundstruktur der Software zur Datenerfassung, -verarbeitung und -auswertung wurde von T. Feigl bereitgestellt. Die Arbeit wurde von M. Philippsen und C. Mutschler beaufsichtigt und rezensiert.

Beitrag: Die Arbeit basiert auf den Ergebnissen der Arbeit von T. Feigl et al. [126], erweitert diese und verbessert deren Ergebnisse. Diese Arbeit liefert die Grundlage für die funk- und trägheitsbasierte Orientierungsschätzung (siehe Kapitel 6) und dient als Basis der Erstkalibrierung und Neukalibrierung der Sensororientierung mittels datengetriebener Bewegungsklassifikation. Die Ergebnisse der Methode fließen in die Posenschätzung (siehe Kapitel 7) ein. Im Rahmen dieser Forschungsarbeit wurde auch ein Patente [119] durch T. Feigl verfasst und teil-veröffentlicht.

- [122] Tobias Feigl, Christopher Mutschler, Michael Philippsen und Eliise Kõre. Acoustical manipulation for redirected walking. In *Proc. Intl. Symp. on Virtual Reality Software and Technology (VRST)*, Seiten 45:1–45:2. Gothenburg, Sweden, 2017.

Anmerkungen: Die wissenschaftlichen Beiträge zu dieser Arbeit stammen von T. Feigl. Das Konzept und die Methoden wurden

von T. Feigl erarbeitet. Die zugrunde liegenden Daten wurden von E. Köre im Rahmen eines Qualifizierungsprojekts erhoben. E. Köre verwaltete und wartete die technischen Ressourcen für die Durchführung der Studie. T. Feigl säuberte und validierten die gesammelten Datensätze und analysierten formell deren Plausibilität. Die Grundstruktur der Software zur Datenerfassung, -verarbeitung und -auswertung wurde von T. Feigl bereitgestellt. Die Arbeit wurde von M. Philippse und C. Mutschler beaufsichtigt und rezensiert.

Beitrag: Diese Arbeit dient als Grundlage für Metriken, die den Einfluss fehlerhafte Orientierungsschätzung auf die menschliche Bewegung bewerten (siehe Kapitel 6), die im Rahmen der Dissertation verwendet werden (siehe Kapitel 3). Darüber hinaus dient diese Arbeit als Motivation für die Schätz- und Optimierungsverfahren für Orientierung und Position der vorliegenden Dissertation (siehe Kapitel 7).

1.4.4 Beiträge zur Posenschätzung

- [133] Tobias Feigl, Andreas Porada, Steve Steiner, Christoffer Löffler, Christopher Mutschler und Michael Philippse. Localization Limitations of ARCore, ARKit, and Hololens in Dynamic Large-Scale Industry Environments. In *Proc. Intl. Conf. Computer Graphics Theory and Applications (GRAPP)*, Seiten 307–318. Valletta, Malta, 2020.

Anmerkungen: Die wissenschaftlichen Beiträge zu dieser Arbeit stammen von T. Feigl. Das Konzept und die Methoden wurden im Rahmen einer Qualifizierungsarbeit von A. Porada unter der Aufsicht von T. Feigl erarbeitet. Die Arbeit basiert auf Daten, die von A. Porada und S. Steiner gesammelt wurden. Die Grundstruktur der Software zur Datenerfassung, -verarbeitung und -auswertung wurde von T. Feigl bereitgestellt und von A. Porada und S. Steiner an das Studiendesign angepasst. Die Arbeit wurde von M. Philippse und C. Mutschler beaufsichtigt und rezensiert.

Beitrag: Diese Arbeit dient v. A. als Grundlage für Evaluationsmetriken, die im Rahmen der Dissertation verwendet werden (siehe Kapitel 3) und beschreibt unter anderem die Probleme aktueller kommerzieller kamerabasierter Lokalisierungssysteme, die diese Dissertation motivieren (siehe Kapitel 1 und 2).

- [130] Tobias Feigl, Lisa Gruner, Christopher Mutschler und Daniel Roth. Real-Time Gait Reconstruction for Virtual Reality using a Single Sensor. In *Proc. Intl. Symp. on Mixed Reality and Augmented Reality (ISMAR)*, Seiten 1–6, 2020.

Anmerkungen: Die wissenschaftlichen Beiträge zu dieser Arbeit stammen von T. Feigl. Das Konzept und die Methoden wurden von T. Feigl erarbeitet. Die zugrunde liegenden Daten wurden von L. Gruner im Rahmen einer Qualifizierungsarbeit unter Anweisung von T. Feigl erhoben. T. Feigl säuberte und validierten die gesammelten Datensätze und analysierte formell deren Plausibilität. Die Grundstruktur der Software zur Datenerfassung, -verarbeitung und -auswertung wurde von L. Gruner und T. Feigl (Lokalisierung) bereitgestellt. Die Arbeit wurde von D. Roth und C. Mutschler beaufsichtigt und rezensiert.

Beitrag: Diese Arbeit liefert die Grundlage für das Verständnis der menschlichen Bewegung (siehe Kapitel 3) und bewertet, wie genau mit den Methoden der Dissertation (siehe Kapitel 7) Voraussagen über zukünftige Ereignisse in Zeitreihen getroffen werden können. Die Arbeit wendet die Ergebnisse dieser Dissertation in einem anderen Bereich (virtuelle Realität) an. Im Rahmen von dieser und vergleichbaren Forschungsarbeiten wurden auch zwei Patente [118, 120] durch T. Feigl verfasst und teil-veröffentlicht.

- [323] Felix Ott, Tobias Feigl, Christoffer Löffler und Christopher Mutschler. ViPR: Visual-Odometry-aided Pose Regression for 6DoF Camera Localization. In *Proc. Intl. Conf. Computer Vision and Pattern Recognition (CVPR) - Workshop on Long-Term Visual Localization, Visual Odometry and Geometric and Learning-based SLAM*, Seiten 42–43. Seattle, Washington, 2020.

Anmerkungen: Die wissenschaftlichen Beiträge zu dieser Arbeit stammen von F. Ott, T. Feigl und C. Löffler. Das Konzept und die Methoden wurden von F. Ott im Rahmen einer Qualifizierungsarbeit erarbeitet und von T. Feigl beaufsichtigt. Die zugrunde liegenden Daten wurden von F. Ott und C. Löffler erhoben. F. Ott säuberte und validierten die gesammelten Datensätze und analysierte formell deren Plausibilität unter der Aufsicht von T. Feigl und C. Löffler. Die Grundstruktur der Software zur Datenerfassung, -verarbeitung und -auswertung wurde von F. Ott bereitge-

stellt. Die Arbeit wurde von C. Mutschler beaufsichtigt und rezensiert unter der Mithilfe von T. Feigl und C. Löffler.

Beitrag: Diese Arbeit zeigt, wie Bildsignale mit verschiedenen Netzarchitekturen zusammengeführt werden können, um eine Pose zu schätzen. Die Arbeit zeigt die Anwendbarkeit der Methode zur Posenschätzung mit datengetriebenen Methoden (siehe Kapitel 7) in einem anderen Anwendungsbereich (z. B. Fahrzeuge statt Menschen) mit anderen Sensoren (z. B. Bildsensoren statt Funksensoren). Weitere Anwendungen werden in Kapitel 8.4 diskutiert. Details zur Posenschätzung mit datengetriebenen Verfahren, bspw. Informationen zum Umgang mit hochdimensionalen Eingabesequenzen werden in dieser Veröffentlichung eingeführt.

1.4.5 Wissensvermittlung

Im Rahmen dieser Dissertation sind Forschungscooperationen und gemeinsame Veröffentlichungen entstanden. Das durch T. Feigl geschaffene Wissen zur Signalverarbeitung, Lokalisierung, datengetriebenen Zeitreihenanalyse und multimodaler datengetriebener Informationsfusion wurde unter Aufsicht von T. Feigl erfolgreich in folgende Forschungsarbeiten übertragen dort modifiziert und angewendet:

- [129] Tobias Feigl, Daniel Roth, Stefan Gradl, Markus Wirth, Marc Erich Latoschik, Björn Eskofier, Michael Philippsen und Christopher Mutschler. Sick Moves! Motion Parameters as Indicators of Simulator Sickness. *Trans. on Visualization and Computer Graphics (TVCG)*, 25(4):3146–3157, 2019.

Anmerkungen: Die wissenschaftlichen Beiträge zu dieser Arbeit stammen von T. Feigl. Das Konzept und die Methoden wurden von T. Feigl erarbeitet. Die Arbeit basiert auf Daten, die im Rahmen einer Qualifizierungsarbeit von T. Feigl gesammelt wurden. S. Gradl und M. Wirth begleiteten die Studie, sammelten und extrahierten Gangparameter, säuberten und validierten die gesammelten Datensätze und analysierten formell deren Plausibilität. Die Grundstruktur der Software zur Datenerfassung, -verarbeitung und -auswertung wurde von T. Feigl bereitgestellt. T. Feigl und D. Roth (Abschnitte 4.1, 4.2 und 4.3 sowie Tabellen 2, 3 und 4) validierten die Ergebnisse der Arbeit und analysierten formell deren Plausibilität. D. Roth lieferte Grafiken (Grafiken 7 und 8) zur Bewertung der Simulatorkrankheit und Präsenz in der

virtuellen Realität. Die Arbeit wurde von M. E. Latoschik, B. Eskofier, M. Philippse und C. Mutschler beaufsichtigt und rezensiert.

Beitrag: Diese Arbeit demonstriert die Anwendbarkeit der im Rahmen der Dissertation entwickelten Methoden und zeigt neben der klassischen Lokalisierung auch andere zukünftige Anwendungsbereiche auf (siehe Kapitel 8.4).

- [17] Anes Redzepagic, Christoffer Löffler, Tobias Feigl und Christopher Mutschler. A sense of quality for augmented reality assisted process guidance. In *Proc. Intl. Symp. on Mixed Reality and Augmented Reality (ISMAR)*, Seiten 1–6, 2020.
- [365] Daniel Roth, Franziska Westermeier, Larissa Brübach, Tobias Feigl, Christian Schell und Marc Erich Latoschik. Brain 2 Communicate: EEG-based Affect Recognition to Augment Virtual Social Interactions. In *Mensch und Computer 2019 - Workshopband*, Seiten 564–565. Hamburg, Germany, 2019.
- [364] Daniel Roth, Larissa Brübach, Franziska Westermeier, Christian Schell, Tobias Feigl und Marc Erich Latoschik. A Social Interaction Interface Supporting Affective Augmentation Based on Neuronal Data. In *Proc. Intl. Symp. on Spatial User Interaction (SUI)*, Seiten 1–4. New York, NY, 2019.

T. Feigl entwarf im Rahmen der folgenden Veröffentlichungen die verwendete Lokalisierungsmethode (siehe Kapitel 4) und Orientierungsschätzung (siehe Kapitel 6) und stellte diese in Form einer öffentlichen Programmierschnittstelle zur Verfügung. Die Veröffentlichungen zeigen die Anwendbarkeit der im Rahmen der Dissertation von T. Feigl entwickelten Methode in anderen Anwendungsbereichen.

- [284] Jean-Luc Lugrin, Florian Kern, Ruben Schmidt, Constantin Kleinbeck, Daniel Roth, Christian Daxer, Tobias Feigl, Christopher Mutschler und Marc-Erich Latoschik. A Location-Based VR Museum. In *Proc. Intl. Conf. Virtual Worlds and Games for Serious Applications (VS-Games)*, Seiten 1–8. Würzburg, Germany, 2018.
- [363] Daniel Roth, Constantin Kleinbeck, Tobias Feigl, Christopher Mutschler und Marc-Erich Latoschik. Beyond Replication: Augmenting Social Behaviors in Multi-User Social Virtual Realities. In *Proc. Intl. Conf. Virtual Reality and 3D User Interfaces (IEEE VR)*, Seiten 215–222. Reutlingen, Germany, 2018.

- [362] Daniel Roth, Constantin Kleinbeck, Tobias Feigl, Christopher Mutschler und Marc Erich Latoschik. Social Augmentations in Multi-User Virtual Reality: A Virtual Museum Experience. In *Proc. Intl. Symp. on Mixed and Augmented Reality (ISMAR)*, Seiten 42–43. Nantes, France, 2017.
- [285] Jean-Luc Lugrin, Florian Kern, Constantin Kleinbeck, Daniel Roth, Christian Daxer, Tobias Feigl, Christopher Mutschler und Marc Erich Latoschik. A Framework for Location-Based VR Applications. In *Virtuelle und Erweiterte Realitat: 16. Workshop der GI-Fachgruppe VR/AR (Berichte aus der Informatik)*, Seiten 148–159. Fulda, Germany, 2019.

1.5 Methode

Dieser Abschnitt führt und ordnet die Methoden entlang der Verarbeitungskette ein. In dieser Dissertation löst diese Verarbeitungskette die Herausforderungen einer datengetriebenen losen Fußgängerkoppeleingangsnavigation mit (gestörten) Funk- und (verrauschten) Trägheitssensoren in dynamischen Bewegungssituationen. Abschnitt 1.5.1 ordnet die Positionsschätzung ein. Abschnitt 1.5.2 ordnet die Geschwindigkeitsschätzung ein. Dann werden die Orientierungsschätzung (Abschnitt 1.5.3) und schließlich die Zusammenführung dieser Schätzungen zu gemeinsamen Posen (Abschnitt 1.5.4) eingeordnet.

Abbildung 1.1 zeigt eine abstrakte Darstellung der vorgeschlagenen Methode. Die Informationen fließen von links nach rechts. Die Eingangsdaten werden zur Laufzeit (engl. online) von Funk- und Trägheitsensoren bereitgestellt. Der Begriff Trägheitssensorstechnologie umfasst im Rahmen der Dissertation Beschleunigungs- und Gyroskop- sowie Magnetsensorstechnologie. Nähere Informationen zu den einzelnen Sensoren sind in den jeweiligen Versuchsabschnitten der Kapitel 4, 5 und 6 sowie in der Auswertung in Kapitel 7 zu finden. Es ist zu beachten, dass die Methoden auch ohne das Magnetfeldsensorsystem (siehe Abbildung 1.1 graues Kästchen) auskommen, das üblicherweise zur Orientierungsschätzung verwendet wird. Darüber hinaus ist es nicht erforderlich, dass die Funk- und Trägheitssensoren im gleichen Koordinatensystem auf dem zu lokalisierenden Objekt platziert sind. Bspw. können die Position des Funksenders in der Hosentasche und die Ori-

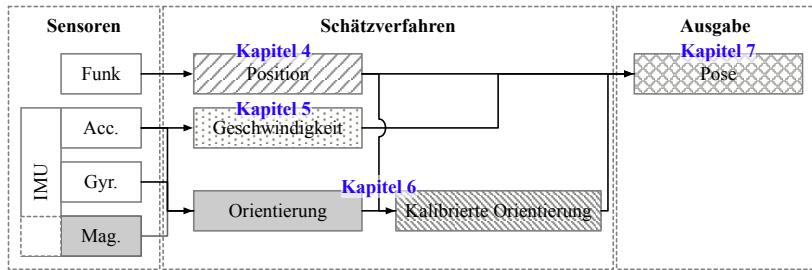


Abbildung 1.1: Verarbeitungskette der vorgeschlagenen Methode und Aufbau der Dissertation anhand der Verarbeitungskette (Positionsschätzung in Kapitel 4, Geschwindigkeitsschätzung in Kapitel 5, Orientierungsschätzung und deren Kalibrierung in Kapitel 6 und Posenschätzung in Kapitel 7). Die Datenströme der Beschleunigungs- (Acc.), Drehraten- (Gyr.) und Magnetfeld-Sensoren (Mag.) einer inertialen Messeinheit (engl. Inertial Measurement Unit, IMU) fließen von links nach rechts durch die Verarbeitungskette.

entierung des Kopfes separat betrachtet werden und eine gemeinsame - fusionierte - Pose ermittelt werden.

1.5.1 Positionsschätzung

Die Position wird typischerweise anhand der sogenannten Uplink-Ankunftszeitdifferenz (engl. Uplink-Time-Difference-of-Arrival, UT-DoA) geschätzt (siehe Abbildung 1.1 Kästchen mit grauen Linien von links unten nach rechts oben). Zu diesem Zweck werden Funksignale von zu lokalisierenden Sendern in die Umgebung gesendet. Diese werden an stationären Antennen empfangen. Die Ankunftszeiten der eindeutig identifizierten Funksignale werden synchronisiert und ein sequenzbasiertes datengetriebenes Lernverfahren schätzt die zugrunde liegende Position des Senders. Kapitel 4 beschreibt die Positionsschätzung im Detail. Um die Robustheit des Positionsschätzers in Situationen mit hochdynamischen Bewegungen sicherzustellen, wird die Methode mit Bewegungsinformationen eines Roboters und typischen sportlichen Aktivitäten bei hohen Beschleunigungen, Geschwindigkeiten und stark und schnell variierenden Orientierungen ausgewertet.

1.5.2 Geschwindigkeitsschätzung

Parallel zur Positionsschätzung wird die Geschwindigkeit des Trägheitssensors geschätzt (siehe Abbildung 1.1 Kästchen mit grauen Punkten). Das Verfahren verwendet die Informationen des Beschleunigungs- und Drehratensensors (engl. Gyroscope, Gyr.), um die Geschwindigkeit zu schätzen, die die relative Geschwindigkeit des Trägheitssensors darstellt, d. h. die Geschwindigkeit wird ohne Richtung geschätzt. Zu diesem Zweck werden zwei verschiedene datengetriebene Methoden vorgestellt: Beide Methoden liefern Geschwindigkeiten, die genauer und robuster als der Stand der Technik sind, auch wenn sie nicht fest auf dem zu lokalisierenden Objekt platziert sind. Eine Methode bietet jedoch etwas genauere Ergebnisse, kostet aber etwas mehr Rechtleistung. Kapitel 5 geht detailliert auf die Geschwindigkeitsschätzung ein. Um die Robustheit des Geschwindigkeitsschätzers in Situationen mit hochdynamischen Bewegungen zu gewährleisten, wird die Methode mit Bewegungsinformationen aus typischen sportlichen Aktivitäten bei hohen Beschleunigungen, Geschwindigkeiten und stark und schnell variierenden Orientierungen ausgewertet.

1.5.3 Orientierungsschätzung

Parallel zur Positions- und Geschwindigkeitsschätzung wird die Orientierung geschätzt (siehe Abbildung 1.1 graues Kästchen). Das Verfahren verwendet die Informationen des Trägheitssensors: Die Beschleunigung, die Rotationsgeschwindigkeit und das Magnetfeld. Grundlage der Methode ist das sogenannte Madgwick-Filter [287], das aus den drei Informationsströmen eine Anfangsorientierung schätzt, die unter bekannten Fehlern leidet: Dem thermischen Rauschen des Gyroskops, den ferromagnetischen Störungen des Magnetfeldes und der Unfähigkeit des Beschleunigungssensors, den Gierwinkel zu bestimmen. Im Rahmen der Dissertation werden daher drei verschiedene Szenarien der Orientierungsschätzung angenommen: (1) Es liegt ein ungestörtes Magnetfeld vor und die Anfangsorientierung kann mit dem Magnetfeldsensor fehlerfrei kalibriert werden; (2) Es liegt ein gestörtes Magnetfeld vor und der Magnetfeldsensor kann die Orientierung nicht fehlerfrei kalibrieren; (3) Der Sensor bewegt sich während der Initialisierungsphase und daher kann sowohl in gestörten als auch in ungestörten Fällen keine kalibrierte Anfangsorientierung geschätzt werden.

In der Dissertation wird daher eine zusätzliche Kalibrierungsmethode zur Orientierungsschätzung vorgeschlagen (siehe Abbildung 1.1 Kästchen mit grauen Linien von links oben nach rechts unten). Das Verfahren verwendet zwei aufeinanderfolgende Positionen, um eine Bewegungsrichtung des Körpers, m.a.W. des Rumpfes, zu schätzen, die mit der aktuellen Blickrichtung des Kopfes verglichen wird, wenn der Trägheitssensor bspw. an einem frei beweglichen Rotations- und Koordinatensystem wie dem Kopf angebracht wird. Der Unterschied zwischen der Bewegungsrichtung des Rumpfes und der Sensorrichtung oder Blickrichtung des Kopfes repräsentiert den Orientierungsfehler sowohl zu Beginn als auch zur Laufzeit. Das Verfahren verwendet die Orientierungsschätzung und die Positionsschätzung, um eine kalibrierte und fehlerfreie Orientierung zu liefern, die mit dem Koordinatensystem der Position ausgerichtet ist. Um zu bestimmen, wie unterschiedliche Rotations- und Koordinatensysteme miteinander ausgerichtet sind, werden zunächst robuste und eindeutige Merkmale aus den Beschleunigungs- und Drehrateninformationen der Sensoren extrahiert, und ein datengetriebenes Verfahren klassifiziert die Übereinstimmung der beiden Koordinatensysteme, m.a.W., ob die zwei Systeme, z. B. Kopf und Körper in die gleiche Richtung zeigen. Wenn das Verfahren einen hohen Grad an Übereinstimmung zwischen den beiden Systemen schätzt, kann die Richtung des Positionsvektors verwendet werden, um den Orientierungsfehler zu berechnen. Kapitel 6 geht detailliert auf die Orientierungsschätzung ein. Um die Robustheit der Methode in Situationen mit hochdynamischen Bewegungen zu gewährleisten, wird der Orientierungsschätzer mit Bewegungsinformationen aus typischen sportlichen Aktivitäten und Aktivitäten in der virtuellen Realität bei hohen Beschleunigungen, Geschwindigkeiten und stark und schnell variierenden Orientierungen ausgewertet.

1.5.4 Posenschätzung

Nachdem die aktuelle Position, die Geschwindigkeit und die absolute kalibrierte Orientierung geschätzt wurden, werden diese zusammengeführt, um die aktuelle Pose des zu lokalisierenden Objekts zu bestimmen (siehe Abbildung 1.1 graues Kästchen mit grauen Feldern). Es ist zu beachten, dass die Position in Funklokalisierungssystemen typischerweise mit einer geringen Rate bestimmt wird, i.d.R. weniger als 10 Mal pro Sekunde (<10 Hz), wobei die Geschwindigkeit und Ausrichtung in Trägheitsnavigationssystemen typischerweise mindestens 100

1 Einleitung

Mal pro Sekunde ($>=100$ Hz) geschätzt wird. Das Verfahren zur Posenschätzung wartet auf eine Anfangsposition des Positionsschätzers und verwendet dann die Geschwindigkeits- und Orientierungsschätzungen, um neue Positionen (=alte Position verkettet mit gerichtetem Distanzvektor) zu schätzen, bis eine neue Position aus den Funkdaten geschätzt und zur Kalibrierung verwendet werden kann. Zusätzlich liefert die Orientierungsschätzung kalibrierte Richtungsinformationen des zu lokalisierenden Objekts. Kapitel 7 geht detailliert auf die Posenschätzung ein. Um die Robustheit des Posenschätzers in Situationen mit hochdynamischen Bewegungen zu gewährleisten, wird die Methode mit Bewegungsinformationen aus typischen sportlichen Aktivitäten und in der virtuellen Realität bei hohen Beschleunigungen, Geschwindigkeiten und stark und schnell variierenden Orientierungen ausgewertet.

1.6 Ergebnisse

Die Dissertation befasst sich mit den Hauptherausforderungen der einzelnen Sensoren bei der Lokalisierung von Fußgängern in Innenräumen und zeigt, wie diese kompensiert werden können, um eine genauere und zuverlässige Pose und Trajektorie zu rekonstruieren. Die Methoden adressieren dabei folgende Kernforschungsfrage:

Wie können sequenzbasierte Lernmethoden die Trajektorien von Fußgängern in dynamischen Anwendungen mithilfe multimodaler Sensordatenströme vorhersagen, um eine genaue und robuste Lokalisierung zu ermöglichen, ohne auf komplexe, handgefertigte Modelle zurückgreifen zu müssen?

Die Experimente der Dissertation zeigen, dass mit der neuen Methode die Pose eines Benutzers in Situationen mit schnellen Änderungen der Bewegung, nämlich Orientierung, Beschleunigung und Geschwindigkeit aus lose am Fußgänger platzierten Sensoren über einen langen Zeitraum genauer und robuster als nach dem Stand der Technik abgeschätzt werden. Die Ergebnisse werden in Kapitel 8 diskutiert.

1.7 Aufbau

Die Abbildung 1.1 zeigt eine abstrakte Darstellung der Verarbeitungskette des vorgeschlagenen Verfahrens der Dissertation sowie den Aufbau der Dissertation auf der Grundlage der Verarbeitungskette. Der

Aufbau wird in diesem Abschnitt beschrieben.

Kapitel 2 diskutiert verwandte Arbeiten, die das Thema und den wissenschaftlichen Hintergrund dieser Arbeit aufspannen. Zunächst werden allgemeine Lokalisierungsverfahren diskutiert, die Trägheitssensoren mit Informationen von entfernten Domänen von verschiedenen Sensoren verknüpfen (Abschnitt 2.1). Anschließend werden Verfahren diskutiert, die Trägheitssensoren mit Informationen von eng verwandten Domänen von Funksystemen verknüpfen und somit vergleichbarer Ergebnisse liefern (Abschnitt 2.2). Abschließend werden verschiedene Methoden diskutiert, die Funk- und Trägheitssensoren zusammenführen (Abschnitt 2.3). Zusätzlich diskutieren die Methodenkapitel 4 bis 7 spezifische verwandte Arbeiten.

In **Kapitel 3** werden die Grundlagen vorgestellt. Zunächst werden Grundlagen zur Lokalisierung diskutiert (Abschnitt 3.1 bis 3.3). Anschließend wird der allgemeine Prozess für die Datenvorverarbeitung beschrieben, der je nach Methode und Anwendung individuell angepasst wird (Abschnitt 3.4). Abschließend werden die Grundlagen der verwendeten Methoden vorgestellt, nämlich konventionelle modellgetriebene Methoden (Abschnitt 3.5) und allgemeine datengetriebene Lernmethoden (Abschnitt 3.6) sowie spezifische sequenzbasierte Lernmethoden (Abschnitt 3.7).

Die **Kapitel 4, 5, 6 und 7** präsentieren die methodischen Beiträge der Dissertation. **Kapitel 4** beschreibt die sequenzbasierte Lernmethode zum Schätzen von Positionen in Funkumgebungen mit Mehrwegeausbreitung von Ankunftszeiten. Das Kapitel geht ausführlicher darauf ein, wie sequenzbasierte, datengetriebene Methoden mit stochastischem Rauschen und langfristigen Abhängigkeiten in Zeitreihendaten umgehen, um genaue und zuverlässige Ergebnisse zu schätzen.

Kapitel 5 beschreibt zwei sequenzbasierte Lernmethoden, eine ressourcenschonende und eine genaue, um die Geschwindigkeit aus Trägheitsmessungen von lose platzierten Sensoren an Fußgängern zu schätzen, wenn sich diese dynamisch bewegen. In diesem Kapitel wird auch näher darauf eingegangen, wie sequenzbasierte datengetriebene Methoden mit räumlichen und zeitlichen Informationen und langfristigen Abhängigkeiten in Zeitreihendaten umgehen können, um genaue und zuverlässige Ergebnisse zu schätzen.

1 Einleitung

Kapitel 6 beschreibt den sequenzbasierten Lernprozess zum Schätzen von Bewegungsmustern aus verrauschten Trägheitsmessungen von Sensoren, die am Kopf von Fußgängern angebracht sind, wenn diese sich dynamisch bewegen. Es zeigt auch, wie diese Bewegungsmuster mit bekannten Richtungsvektoren kombiniert werden können, um eine absolute Orientierung im Raum zu schätzen und ungenaue fehlerhafte Orientierungen zu kalibrieren.

Kapitel 7 beschreibt zwei Methoden, eine konventionelle modellgetriebene und eine neuartige sequenzbasierte datengetriebene, um Posen aus variablen Eingabedaten, bspw. Position, Geschwindigkeit und Orientierung, zu schätzen und Trajektorien zu rekonstruieren. In diesem Kapitel werden die Wechselbeziehungen und das Zusammenführen der vorherigen Methoden bewertet. Es werden auch die Stärken und Schwächen sequenzbasierter, datengetriebener Methoden im Umgang mit langfristigen Abhängigkeiten in Zeitreihendaten näher erläutert.

Die vier Kapitel sind einheitlich und wie folgt strukturiert: Zunächst wird der Hintergrund der jeweiligen Methode eingeführt. Dann werden die Probleme und Herausforderungen der jeweiligen Methode diskutiert. Anschließend werden eng verwandte Vorarbeiten diskutiert. Danach werden die Beiträge oder Forschungsfragen aufgelistet, bevor die tatsächlichen Methoden beschrieben werden. Im Anschluss daran werden Experimente und die Auswertung vorgestellt. Abschließend werden die Ergebnisse diskutiert und eine Zusammenfassung, Schlussfolgerung und ein Ausblick gegeben.

Kapitel 8 fasst die Dissertation zusammen, diskutiert die Ergebnisse, diskutiert die Forschungsfragen und gibt Implikationen, zeigt drei Anwendungen der Methoden, die neuen Technologien im Forschungsbereich der virtuellen Realität hervorgebracht haben und gibt einen Ausblick auf offene Fragen und zukünftige Arbeiten.

Kapitel 9 listet die eigenen Forschungsarbeiten und Forschungskooperationen in Form von Konferenzbeiträgen und Zeitschriftenartikeln auf, bietet einen Überblick über eingereichte und erteilte Patente und Eigentumsrechte und listet die Qualifizierungsarbeiten auf, die im Rahmen der Dissertation von T. Feigl betreut wurden.

2 Verwandte Arbeiten

Über die Kopplung von Trägheitssensoren mit verschiedenen anderen Sensorfamilien wird in der Literatur seit vielen Jahren berichtet. Die meisten verwandten Arbeiten verwenden Sensorkopplung und komplementäre Sensoreigenschaften, um die Genauigkeit und Zuverlässigkeit der Lokalisierung zu verbessern. Daher führt dieses Kapitel die verwandten Arbeiten, von 2015 bis 2021, auf und diskutiert die wesentlichen renommierten Arbeiten. Zunächst erörtert Abschnitt 2.1 prominente Arbeiten, die Trägheitssensoren mit Informationen von Magnetfeld-, akustischen, optischen, Bluetooth- und drahtlosen lokalen Netzwerk-Sensoren kombinieren, um Positionen zu schätzen. Diese Arbeiten können nicht direkt mit dieser Dissertation verglichen werden. Dennoch geben sie dem Leser einen Überblick darüber, was außerhalb des Fokus dieser Dissertation erforscht wird. Abschnitt 2.2 erörtert dann prominente, eng verwandte Arbeiten, die Trägheitssensoren mit Informationen von (vergleichbaren) Funksystemen kombinieren, um Positionen zu schätzen. Dann diskutiert Abschnitt 2.3 explizit prominente Methoden, die Funk- und Trägheitssensoren fusionieren. Jeder Abschnitt behandelt relevante Arbeiten, listet wesentliche Ergebnisse in einer Tabelle auf und schließt mit einer Schlussfolgerung. Darüber hinaus diskutieren Kapitel 4 bis 7 methodenspezifische verwandte Arbeiten.

2.1 Lokalisierung mit Trägheitssensoren

Die folgenden verwandten Arbeiten verknüpfen Trägheitssensoren mit exogenen Informationen von Magnetfeld- (Abschnitt 2.1.1), akustische- (Abschnitt 2.1.2), optische- (Abschnitt 2.1.3), Bluetooth- (Abschnitt 2.1.4) und drahtlose lokale Netzwerk- (Abschnitt 2.1.5) Sensoren. Da diese Sensoren in dieser Dissertation nicht näher betrachtet werden, werden Forschungsbeiträge und Ergebnisse, die diese Sensoren verwenden, nicht im Detail diskutiert und abgegrenzt.

2.1.1 Magnetfeld

Der Magnetfeldsensor misst normalerweise das Erdmagnetfeld, um die absolute Gier-Winkel-(engl. yaw-)Ausrichtung des Sensors abzuschätzen. Er zeichnet jedoch auch das lokale Magnetfeld auf, das

2 Verwandte Arbeiten

durch verschiedene Quellen verursacht wird und die Orientierungsschätzung verschlechtert. Das lokale Magnetfeld ermöglicht allerdings die Fingerabdruck-basierte Lokalisierung. Pasku et al. [332] zeigen, dass ein magnetisches Positionierungssystem in einer statischen Szene einen mittleren Positionsfehler von 0.6 m liefert, wenn die Ankerausrüstung gleichmäßig in der Umgebung verteilt ist. Qian et al. [346] fusionieren Wi-Fi- und Magnetfeldfingerabdrücke mit PDR und erzielen ähnliche Positionsgenauigkeiten. Kok et al. [240] verwenden die Informationen über das lokale Magnetfeld, um Magnetfeldkarten des Innenraumes zu erstellen. Der Vergleich der gemessenen mit den Referenz-Fingerabdrücken von Trägheitsmessungen und Magnetometer liefert den Ort und die Ausrichtung des mobilen Geräts mit einer Genauigkeit zwischen 0.2 m und 0.3 m.

Fazit: Die Magnetfeldsensoren sind genau (durchschnittlicher Fehler <1.0 m), wenn die Lokalisierungsumgebung frei von elektronischen, metallischen oder magnetischen Störquellen ist und die Aktualisierungsrate niedrig ist. Nachteile sind die träge Abtastrate und die dynamischen Schwankungen des lokalen Magnetfelds im Alltag. Da in den meisten Gebäuden Stromleitungen und Stahlträger in die Wände und Böden nichtdeterministisch eingebaut sind, die das Magnetfeld dynamisch stören, sind sowohl die Orientierungsschätzung als auch die Fingerabdruck-basierte Lokalisierung unzuverlässig und ungenau. Da der Hauptanwendungsbereich der Methoden in dieser Dissertation in Innenräumen mit Magnetfeldstörungen liegt, werden die Magnetfeldsensoren in dieser Dissertation explizit nicht berücksichtigt.

2.1.2 Akustisch

Jiang et al. [212] schätzen die akustische Reichweite zwischen einem selbstlokalisierten mobilen Roboter und einem Mobiltelefon in Innenräumen und erreichen eine Schätzgenauigkeit von 0.43 m bis 1.12 m. Fischer et al. [139] nutzen ein erweitertes Kalman-Filter (engl. Extended Kalman-Filter, EKF), um die Positionsschätzungen eines PDR-Systems und von Ultraschallbaken¹ zusammenzuführen. Um die Positionsgenauigkeit von Fischer et al. zu verbessern entwickeln Zhao et al. [513, 281] ein Sigma-Punkt Quadratwurzel-Kalman-Filter (engl. Unscented Square-Root Kalman-Filter, USRKF) und erzielen eine

¹Eine Bake (engl. beacon) ist eine feste Funksender- oder -empfängereinheit.

Genauigkeit von <0.10 m. Liu et al. [280] kombinieren Ultraschall mit IMU in einer PDR-Methode mit Mobiltelefonen. Ihr KF reduziert die Doppler-Verschiebung bei der akustischen TDoA-Positionierung auf einer Fläche von 10×6 m. Das Mobiltelefon muss am Fuß platziert werden, um eine Nullgeschwindigkeitsaktualisierung (engl. Zero Velocity Update, ZUPT)² und eine Nullwinkelgeschwindigkeitsaktualisierung (engl. Zero Angular Rate Update, ZARU)³ zu verwenden. Sie erreichen einen durchschnittlichen Fehler von 0.45 m. Yang et al. [492] nutzen ein EKF, um die Trägheitssensoren eines Mobiltelefons mit den Daten eines akustischen Lokalisierungssystems und UWB zu fusionieren. Ein Random Forest (RF) analysiert CIR-Parameter (engl. Channel Impulse Response) der UWB-Messungen, um NLoS zu identifizieren. UWB Messungen ohne NLoS werden dann mit den Positionsschätzungen der anderen Systeme unter Verwendung eines KF zusammengeführt. Dieses System liefert keine Ergebnisse, sobald es einen NLoS Pfad identifiziert, sodass es nur in Innenräumen mit expliziter LoS zwischen Sender und Empfänger funktioniert.

Fazit: Die akustischen Sensoren sind genau (durchschnittlicher Fehler <1.0 m) und die Aktualisierungsrate ist hoch. Die Abdeckung und die Reichweite sind jedoch auf wenige Meter (<5 m) begrenzt. Der Hauptnachteil ist die Abhängigkeit von der direkten Sichtlinie zur Lokalisierungsumgebung bei der Bestimmung des Standorts. Die Dissertation untersucht die Funklokalisierung unter vergleichbarer Mehrwegeausbreitung sowie konstruktive und destruktive Signalüberlagerungen. Daher können die entwickelten Methoden möglicherweise auf die Lokalisierung mit akustischen Sensoren und Trägheitssensoren übertragen werden. Aufgrund der kurzen Reichweite werden akustische Sensoren im Rahmen der Dissertation nicht näher betrachtet.

²Die Nullgeschwindigkeitsaktualisierung (engl. Zero Velocity Update, ZUPT) kalibriert akkumulierte Sensorfehler, wenn bekannte Bewegungsmomente, nämlich die Geschwindigkeitsschätzung = Null, identifiziert werden, m.a.W. wenn ein fußmontierter Sensor den Boden berührt und die Geschwindigkeit 0 misst.

³Die NullWinkelgeschwindigkeitsaktualisierung (engl. Zero Angular Rate Update, ZARU) kalibriert ebenfalls akkumulierte Sensorfehler, wenn bekannte Bewegungsmomente, nämlich die Winkelgeschwindigkeitsschätzung = Null, identifiziert werden, m.a.W. wenn ein fußmontierter Sensor den Boden berührt und nicht rotiert.

2 Verwandte Arbeiten

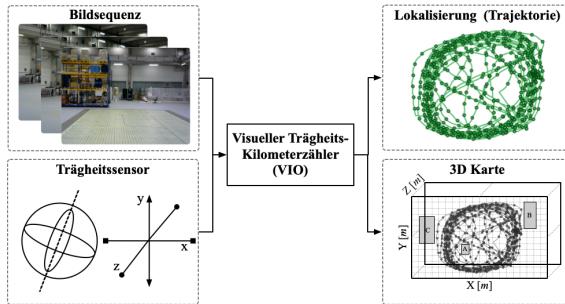


Abbildung 2.1: Visuelle Trägheitsdometrie (engl. Visual-Inertial Odometry, VIO) ist eine spezielle Instanz von SLAM (ohne Schleifenschluss), bei der visuelle Daten (Kamera) und Daten der Trägheitsmesseinheit (IMU) verwendet werden, um den Standort zu schätzen und die Umgebung mit einem deutlich höheren Durchsatz abzubilden.

2.1.3 Optisch

Benini et al. [39] schlagen eine Positionierungsmethode für fliegenden Drohnen in Innenräumen vor, bei der Kamera-, IMU- und UWB-Sensorinformationen zusammengeführt werden. Durch die Verwendung von Quick Response (QR)-Codes zur Initialisierung und Rekalibrierung an bestimmten Positionen erreichen sie eine zweidimensionale Positionierungsgenauigkeit von >0.10 m. Heute wird zur Lokalisierung von mobilen Robotern in Innenräumen die deutlich genauere kamerabasierte simultane Lokalisierung und Kartierung (engl. Simultaneous Localization And Mapping, SLAM) verwendet [303, 347]. Die meisten aktuellen Verfahren [41, 90, 491] basieren auf Oriented FAST and rotated BRIEF-SLAM-2 (ORB-SLAM2) von Mur-Artal et al. [347]. SLAM-Technologien hängen allerdings stark von Lichtverhältnissen und der Berechnungsgeschwindigkeit von Computern ab. Hess et al. [186] verwenden deshalb einen zusätzlichen LiDAR-Sensor (engl. Light Detection And Ranging, dt. optische Abstands- und Geschwindigkeitsmessung), um die Positionierungsgenauigkeit des mobilen Roboters zu verbessern. Qin et al. [347] entdecken, dass Kamera-trägheitsbasierte-Systeme (engl. Visual Inertial Navigation System, VINS) eine genauere Lokalisierung mobiler Roboter durch visuelle Verlagerung und Schleifenerkennung (engl. loop closure) als SLAM ermöglichen (durchschnittlicher Fehler von 0.13 m). Im Gegensatz dazu muss die visuelle Trägheitsdometrie (engl. Visual-Inertial Odometry,

VIO) (siehe Abbildung 2.1) keine Schleife schließen, ist etwas ungenauer, aber wesentlich ressourcenschonender als SLAM und VINS. Cadena et al. [59] zeigen, dass EKF-basierende SLAM-Systeme eine ebenso gute Leistung erzielen können. Hervorragende Beispiele für EKF-basierte SLAM-Systeme sind das Multi-Zustand-eingeschränktes-Kalman-Filter (engl. Multi-State Constraint Kalman-Filter, MSCKF) von Mourikis et al. [302] sowie die kamerabasierten Trägheitsnavigationssysteme von Kottas et al. [184] und Hesch et al. [185]. Kamerabasierte Systeme liefern sehr rauscharme Informationen, die eine hochpräzise Kalman-Filterung begünstigen, wenn die Lichtverhältnisse der Einsatzumgebung gleich bleiben und ausreichen. Dann schätzen sie Positionen sehr genau [200, 349, 350, 351]. Li et al. [263] entwickeln eine Positionierungsmethode für Roboter in Innenräumen. Sie fusionieren Informationen von 3D-Laserscanner und INS (engl. Inertial Navigation System, dt. Trägheitsnavigationssystem) und zeigen, dass die Positionierungsgenauigkeit im Vergleich zu Nur-INS- und Nur-UWB-Ansätzen erheblich verbessert wird.

Fazit: Die Selbstlokalisierung mit Kamera- und LiDAR-Sensoren ist genau (durchschnittlicher Fehler <0.10 m) und die Aktualisierungsrate ist hoch. Die Abdeckung und Reichweite ist jedoch auf wenige Meter begrenzt. Der Hauptnachteil ist die Abhängigkeit von der direkten Sichtlinie zur Lokalisierungsumgebung und ausreichenden und konstanten Lichtverhältnissen bei der Standortbestimmung. Die Verknüpfung mit Trägheitssensoren kompensiert diese Probleme und liefert die hohe Genauigkeit. Optische Sensoren zur Selbstlokalisierung (SLAM, VINS und VIO) werden im Rahmen der Dissertation nicht näher betrachtet. Allerdings werden sehr genaue und präzise hochredundante *stationäre* Systeme mit über 20 spezialisierten hochauflösenden optischen Sensoren (<0.01 m) als Referenzlokalisierungssysteme eingesetzt.

2.1.4 Bluetooth und Nahfeldkommunikation

Die Kombination von Funktechnologien wie Bluetooth und Identifizierung mithilfe elektromagnetischer Wellen (engl. Radio-Frequency Identification, RFID), und Trägheitspositionierung kann die Positionierungsgenauigkeit nur in sehr geringem Maße verbessern, da die Positionierungsgenauigkeit von Bluetooth- und RFID-Methoden niedrig ist. Li et al. [267] und Murata et al. [305] fusionieren Trägheitssensoren für mobile Geräte und die Empfangsfeldstärke (engl.

2 Verwandte Arbeiten

Received Signal Strength, RSS) von Bluetooth-Baken und erreichen in experimentellen Studien Lokalisierungsgenauigkeiten im Innenraum von 1.5 m bis 3.0 m. Abbildung 2.2 zeigt ein typisches RSS-basiertes Lokalisierungsverfahren. Die Anzahl der RFID Anker, die zur Abdeckung eines Innenraums nötig sind, nimmt mit der Raumgröße zu, da die effektive Reichweite zwischen RFID Sender und Empfänger sehr klein (<1 m) ist. Daher installieren Jimenez et al. [371] bspw. 71 RFID-Tags in einem Bereich von $60\text{ m} \times 40\text{ m}$, um eine Positionsgenauigkeit von 3 m zu erzielen. Jiménez et al. [214] kombinieren die RSS-Werte von BLE (Bluetooth Low Energy, dt. Bluetooth mit geringerem Stromverbrauch) mit IMU und erzielen einen Positionsfehler von 4.5 m in einem Museum auf einer Fläche von 7×1.5 m. Ihr UWB-Referenzsystem liefert jedoch bereits einen Fehler von 1.6 m. Zampella et al. [506] kombinieren RSS-Werte (RFID Sensor in der Hosentasche; durchschnittlicher Fehler von 4 m), IMU-Signale (Sensor am Fuß; unbekannter Fehler) und TDoA- und Ankunftsinkel-Werte (engl. Angle of Arrival, AoA) (Sensor in der Brusttasche; durchschnittlicher Fehler von 7 m). Ihr PDR-Verfahren nutzt ZUPT und führte zu Fehlern zwischen 4 m mit 12 RFID-Ankern und 2 m mit 34 RFID-Ankern. Ihre Methode ist sehr rechenintensiv und liefert keine Position, wenn keine Sichtverbindung zwischen dem Sender und einem Empfänger besteht.

Fazit: Die Signalstärkemessungen von Bluetooth und RFID sind sehr ungenau (durchschnittlicher Fehler >1 m) und die Aktualisierungsrate sowie die Abdeckung und Reichweite sind sehr gering. Die Hauptnachteile sind die variable niedrige Reichweite von 1 m bis höchstens 100 m und die Ungenauigkeit bei der Standortbestimmung. Diese Sensoren werden im Rahmen der Dissertation nicht näher betrachtet.

2.1.5 Drahtloses lokales Netzwerk

Walters et al. [451] kombinieren Wireless Local Area Network (WLAN)- (auch Wi-Fi-) Sensordaten mit Trägheitssensormessungen und konstruieren ein tragbares Navigationssystem. Eine Datenbank mit Referenzwerten (sogenannte Fingerabdrücke) der Wi-Fi- und Trägheitssensoren kann mit den Messungen eines Mobilgeräts verglichen werden (sogenannter Vergleich der Fingerabdrücke). Sie erreichen eine mittlere Positionsgenauigkeit von 0,38 m. Im Gegensatz dazu erzielen Zhang et al. [511] mit demselben Verfahren einen Lokalisierungsfehler von 5,7 m, während sich Probanden für 5 bis 10 Minuten in Innenräumen bewegen.

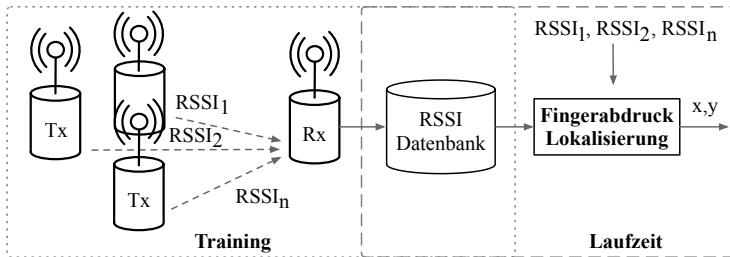


Abbildung 2.2: Fingerabdruck-basierte Lokalisierung mit Hilfe der Empfangsfeldstärke (engl. Received Signal Strength Indicator, RSSI). RSSI schätzt die Entfernung zwischen einem Sender (engl. Transmitter, Tx) und einem Empfänger (engl. Receiver, Rx). Je höher der RSS-Wert, desto kleiner ist der Abstand zwischen Tx und Rx. Die RSS-basierte Lokalisierung erfordert eine Fingerabdruckbasierte Abgleichfunktion oder Trilateration. Letztere werden üblicherweise verwendet, um den absoluten Abstand zwischen dem Benutzergerät und mindestens drei Referenzpunkten zu schätzen.

men bewegen. Obwohl Qian et al. [346] zusätzlich Fingerabdrücke des Magnetfeldes nutzen, erzielen sie vergleichbar niedrige Genauigkeiten. Daher schlagen Zhuang et al. [520] einen Kaskadenfilter-Algorithmus vor, der Wi-Fi-Fingerabdrücke und Trägheitsmessungen fusioniert, um die Genauigkeit der Positionierung in Innenräumen zu verbessern, erzielen aber auch keine genaueren Positionen. Wang et al. [455] schlagen ein Schema für die Positionierung in Innenräumen vor, indem Fingerabdrücke von RSSI-Messungen (engl. Received Signal Strength Indication, dt. Indikator für die Empfangsfeldstärke) des Wi-Fi Systems und die Trägheitsmessungen eines Mobiltelefons mit einer Karte der Innenräume zusammengeführt werden. Abbildung 2.2 zeigt ein typisches RSS-basiertes Lokalisierungsverfahren. Diese Methode kann zwar das Problem der Akkumulation von trägeitsbasierten Kursfehlern unter Verwendung von Kartenbeschränkungen lösen, allerdings ist die Auflösung und Abtastrate der Zeitstempel und somit die Genauigkeit von Wi-Fi-Messungen sehr gering. Die Versuchsergebnisse von Bachtler et al. [22] zeigen, dass mit dem Ansatz von Wang et al. [455] ein durchschnittlicher Fehler von 3.2 m und eine Genauigkeit von 4.1 m in 90% aller Fälle erreicht werden kann, wenn sich Probanden mit durchschnittlich 1.0 m/s bewegen. Deshalb schlagen Deng et al. [98] ein EKF vor, um Wi-Fi-RSS direkt mit den Trägheitssensoren des Mobiltelefons zu fusionieren. Masiero et al. [295] fusionieren

2 Verwandte Arbeiten

dieselben Sensoren ebenfalls direkt, nutzen dazu aber ein PF. Waqar et al. [465] verwenden ein KF, um dieselben Daten zu fusionieren. Diese Methoden liefern allerdings auch keine genauen Positionen bei Bewegungsgeschwindigkeiten über 1.0 m/s. Vermutlich gibt es unterschiedliche Ergebnisse, da die Genauigkeit der Fingerabdruckverfahren stark mit der Dichte der Referenzaufzeichnungen (mehr Daten liefern eine höhere Genauigkeit), der Bewegungsgeschwindigkeit (Genauigkeit nimmt mit zunehmender Geschwindigkeit drastisch ab) und der Umgebungs dynamik zusammenhängt. Wang et al. [457] entwickeln ein robustes KF für hochspezifische synthetische ToA-Werte eines Wi-Fi-Systems. Eine Simulation ermöglicht die Bewertung von LoS- und NLoS-Situationen und liefert einen Fehler von 2.1 m.

Fazit: Wi-Fi ist sehr ungenau (durchschnittlich Fehler >1 m) und die Aktualisierungsrate sowie die Abdeckung und Reichweite sind sehr gering. Der Hauptnachteil sind die niedrige Bandbreite (Intervall mit ungefähr 100 Werten zur Positionsbestimmung) und Abweichungen von bis zu 15 m bei der Standortbestimmung. Diese Sensoren werden im Rahmen der Dissertation nicht näher betrachtet. Die in der Dissertation entwickelten Methoden können möglicherweise auf die Lokalisierung mit Wi-Fi Sensoren und Trägheitssensoren übertragen werden.

Zusammenfassung

In der Tabelle 2.1 werden herausragende Arbeiten aufgelistet und die verwendeten Sensoren, Methoden, ihre Genauigkeit und Nachteile zusammengefasst. Die vorgestellte Arbeit verwendet Sensortypen, die im Rahmen dieser Dissertation nicht untersucht werden. Da diese Dissertation eine lose Kopplung unterschiedlicher Sensorinformationen ermöglicht, ist es offensichtlich, dass die Methode dieser Dissertation durch diese Sensortypen ergänzt werden kann. Zu diesem Zweck könnten bspw. die jeweiligen Sensorinformationen in Prozessen ähnlich den in den Kapiteln 4 bis 6 beschriebenen zu einer Position, Entfernung oder Orientierung verarbeitet werden. Diese könnten dann zu einer Pose, wie in Kapitel 7 beschrieben, zusammengeführt werden.

⁴Maschinelles lernen, engl. Machine Learning, ML.

Tabelle 2.1: Ergebnisse renommierter verwandter Arbeiten der letzten fünf Jahre, die Trägheitsmessungen mit verschiedenen Sensortypen fusionieren. Übersicht über verwendete Sensoren, Methoden, höchstmögliche Genauigkeit oder geringste Fehler **fettgedruckt** und Nachteile.

Autoren	Sensortyp	Methode	Fehler in [m] *	Nachteile
Qian et al. [346]	Magnetfeld	PDR	0.70	Anfällig für elektro- und ferromagnetische Störquellen in Gebäuden.
Pasku et al. [332]		Speziell	0.60	
Kok et al. [240]		Fingerabdruck	0.20 bis 0.30	
Jiang et al. [212]	Akustisch	Speziell	0.43 bis 1.12	Anfällig für akustische Störquellen; Kurze Reichweite;
Fischer et al. [139]		EKF	0.20	
Zhao et al. [513, 281]		UKF	<0.10	Die Genauigkeit ist direkt abhängig von LoS.
Liu et al. [280]		PDR und KF	0.45	
Yang et al. [492]		EKF und ML ⁴	0.90	
Qin et al. [347]	Optisch	VINS	0.13	
Benini et al. [39]		Fingerabdruck	0.10	Reichweite durch die Kameraauflösung begrenzt.
Mourikis et al. [302]		KF-SLAM	0.13	Genauigkeit ist abhängig von Licht- und Sichtverhältnissen.
Kottas et al. [184]		EKF-SLAM	0.12	
Hesch et al. [185]		EKF-SLAM	0.14	
Li et al. [267]	Bluetooth	Speziell	1.57	Hohe Ungenauigkeit der Signalstärkemessungen und niedrige Reichweite der Identifizierung mit Hilfe elektromagnetischer Wellen.
Murata et al. [305]		KF	3.01	
Zampella et al. [506]		PDR	2.00 bis 7.00	
Jimenez et al. [371, 214]		Speziell	3.00 bis 4.50	
Walters et al. [451]	Wi-Fi	Fingerabdruck	0.38	
Zhang et al. [51]		Fingerabdruck	5.70	
Qian et al. [346]		Kaskadenfilter	6.00	
Wang et al. [455]		Fingerabdruck	4.20	
Bachtler et al. [22]		Fingerabdruck	3.20 bis 4.10	Niedrige Bandbreite führt zu sehr ungenauen Positionen.
Deng et al. [98]		EKF	2.7	
Masiero et al. [295]		PF	3.22	
Waqr et al. [465]		KF	1.98	
Wang et al. [457]		KF	2.10	

*) Die Ergebnisse, die zur Bewertung der verwendeten Methoden verwendet wurden, werden bestmöglich aus der jeweiligen Darstellungsform extrahiert.

2.2 Funk- und trägheitsbasierte Lokalisierung

Die folgenden verwandten Arbeiten verknüpfen Trägheitssensoren, nämlich Beschleunigungs-, Drehraten- und Magnetfeldmessungen, mit exogenen Informationen aus ähnlichen Domäne wie diese Dissertation, nämlich Kanalstatusinformationen von Funksystemen. Zunächst wird die Fusion von Funk- und Trägheitssensoren am Beispiel des globalen Positionsbestimmungssystems (engl. Global Positioning System, GPS) motiviert (Abschnitt 2.2.1). Dann diskutiert Abschnitt 2.2.2 die Nachteile und Vorteile von Trägheits- und Funksystemen und deren Fusion. Danach werden der Einfluss der Systemarchitektur (Abschnitt 2.2.3), der Einfluss zusätzlicher Kontextinformationen (Abschnitt 2.2.4) und der Einfluss von Mehrwegeausbreitung (Abschnitt 2.2.5) auf die Genauigkeit der Fusion diskutiert.

2.2.1 Globales Positionsbestimmungssystem

Die Fusion von relativen Sensoren wie Trägheitssensoren mit absoluten Sensordatenquellen wie das GPS ist am ältesten und am weitesten verbreitet [110]. Diese Fusion kombiniert die Vorteile relativer Sensoren hinsichtlich ihrer lokalen Genauigkeit im Sensorkoordinatensystem mit der Fähigkeit absoluter Sensoren, die Unsicherheit und Ungenauigkeit der relativen Sensoren im Weltkoordinatensystem zu reduzieren. Viele Arbeiten [345, 79], allen voran Sasiadek et al. [385], basieren auf einem KF, um die von einem GPS-Empfänger und Trägheitssensoren erhaltenen Daten zusammenzuführen. Mit dem Aufkommen von Ultra-Breitband Funktechnologien (engl. Ultra-Wideband, UWB), die deutlich genauerer Positionen als GPS liefern und in Innenräumen und im Freien funktionieren, wurden die bekannten Verfahren zusätzlich mit UWB Sensoren fusioniert [447, 35, 456, 194, 195]. Z. B. verwenden Hol et al. [194, 195] ein EKF, um die sechsdimensionale (6D)-Pose (= 3D-Position + 3D-Orientierung) mit GPS, UWB und INS zu schätzen. Vydhyanathan et al. [447] nutzen neben GPS und INS Impulsfunksignale eines UWB Systems in der Umgebung, in der GPS nicht gut oder überhaupt nicht funktioniert, und ermöglichen eine horizontale Genauigkeit von 0.5 m. Ebenso fusionieren Belakbir et al. [35] GPS und INS mit UWB-Sensordaten, um eine Lokalisierung im Freien zu erreichen. Sie platzieren mehrere UWB-Sensoren auf einem Gebäude, während sich eine mobile Einheit im Gebäude bewegt und behaupten,

dass die Genauigkeit sowohl im Innen- als auch im Außenbereich genauer ist als eine Fusion, die nur GPS und INS verwendet. Wang et al. [456] entwickeln auch ein eng gekoppeltes integriertes GPS, UWB und INS-Positionierungssystem, basierend auf dem adaptiven robusten KF. Ihr System positioniert mit einem durchschnittlichen Fehler von 0.20 m, wenn GPS, UWB und IMU verfügbar sind. Die Experimente werden jedoch nur in LoS-Umgebungen im Freien durchgeführt, sodass eine Aussage über die Positionsschätzgenauigkeit in natürlichen NLoS-Umgebungen unmöglich ist. Außerdem werden nur langsame statische Roboterbewegungen ohne schnelle und abrupte Bewegungsänderungen auf einem rechteckigen Pfad untersucht.

2.2.2 Nach- und Vorteile der Fusion

Die Nachteile und Vorteile von Positioniersystemen, die auf Trägheitsmessungen oder Funkinformationen basieren, und deren Fusion werden im Anschluss diskutiert. Arbeiten, die trägheitsbasierte Systeme verwenden, werden zusätzlich in Kapitel 5 und 6 und Arbeiten, die funkbasierter Systeme verwenden, werden zusätzlich in Kapitel 4 ausführlich diskutiert.

Nach- und Vorteile von Trägheitssystemen

Üblicherweise schätzt die IMU-basierte Fußgängerkoppelnavigation [421, 5, 285] die Schrittlänge, die Anzahl der Schritte und die Richtung der Schritte eines Fußgängers. Beschleunigungsmesser messen die Intensität der Bewegung und Gyroskope messen die Änderung der Lage kontinuierlich und regelmäßig [156]. Allerdings hat jeder Fußgänger individuelle Gangmerkmale und aktuelle Verfahren können die Gangparameter nicht dynamisch an jeden Fußgänger anpassen. So können herkömmliche Verfahren keine genauen Positionen schätzen [421]. IMU-basierte Ansätze akkumulieren Orientierungs- und Positionsfehler mit zunehmender Zeit durch die Integration von Messrauschen der Gyroskop- und Beschleunigungssensorik. Die Verzerrungsinstabilitäten (engl. bias-instability) des Gyroskops führen zu Winkelfehlern. Wenn diese über einen langen Verfolgungszeitraum signifikant werden (driften⁵), verschlechtert sich auch die

⁵Drift ist eine langsame unerwünschte, oftmals unbekannte und nichtdeterministische, Änderung einer Systemeigenschaft oder der Ausgangsgröße, die nicht mit einer Änderung der Eingangsgröße zusammenhängt [50]. Die Gyroskopdrift ist hauptsäch-

2 Verwandte Arbeiten

Gesamtleistung (durchschnittlicher Fehler von 0.80 m bis 2.18 m) der Positionsschätzung [432], selbst wenn die Fehler mit anderen Sensoren, bspw. Ultraschall und Magnetometer kompensiert werden. Zampella et al. [505] zeigen, dass die Genauigkeit geschätzter Trajektorien aufgrund der Anhäufung von Richtungsschätzungsfehlern während des Abbiegens vor allem bei vielen Kurven erheblich abnimmt.

Stattdessen erzielen rein IMU-basierte Verfahren bei linearen Bewegungen und über kurze Zeiträume allerdings sehr hohe Positionierungsgenauigkeiten in Echtzeit [262].

Nach- und Vorteile von Funksystemen

Ein UWB-Subsystem schätzt die Position eines mobilen Knotens (Sender) mithilfe der Multilateration [288], bei der die Abstandsmessungen zwischen dem Sender und mindestens drei Ankerknoten (Empfängern) kombiniert werden [256]. Die daraus resultierenden Infrastrukturstarken für Funksysteme sind höher als die Kosten für reine INS-Systeme. UWB-Empfängerknoten müssen explizit in der Umgebung installiert werden und Expertenwissen ist nötig, um die Systeme optimal zu kalibrieren und zu betreiben. Die Situation ist schlimmer für große Bereitstellungsbereiche, in denen die Anzahl der erforderlichen Empfängerknoten erheblich zunimmt, um sicherzustellen, dass jeder mögliche Standort in der Umgebung von mindestens drei Empfängerknoten abgedeckt wird. Außerdem sind Funksysteme anfällig für Blockierungen und weisen Volatilität und Unregelmäßigkeiten auf, wenn Objekte in der Funkumgebung die Sichtverbindung zwischen Sender und Empfänger blockieren. Diese Störungen führen zu einer niedrigen Positionsgenauigkeit [278]. Bspw. in VR-(engl. Virtual Reality, dt. virtuelle Realität-)Themenparks

lich auf die Integration von zwei Komponenten zurückzuführen: einer sich langsam ändernden Variablen, die als Verzerrungsinstabilitäten (engl. bias-instability) bezeichnet wird, und einer stochastischen Rauschvariablen mit höherer Frequenz, die als Winkelzufallsbewegung (engl. Angular Random Walk) bezeichnet wird. Verzerrungsinstabilitäten verursachen aufgrund der Integration von inhärenten Fehlern und Rauschen des Gyr. im Laufe der Zeit eine Drift. Und die Integration der konstanten Verzerrungsinstabilitäten führt zu Winkelfehlern. Diese Fehler häufen sich, wenn gyrokopbasierte Rotations- oder Winkelschätzungen langfristig drifteten. Die konstante Drift eines Gyr. kann über den bekannten Temperaturbereich gemessen werden, indem der Ausgang über einen langen Zeitraum gemittelt wird, während sich der Sensor nicht dreht. Diese Annahmen sind in der Praxis oft ungültig, daher können Orientierungen nur mit einem Gyr. nicht zuverlässig geschätzt werden. [50]

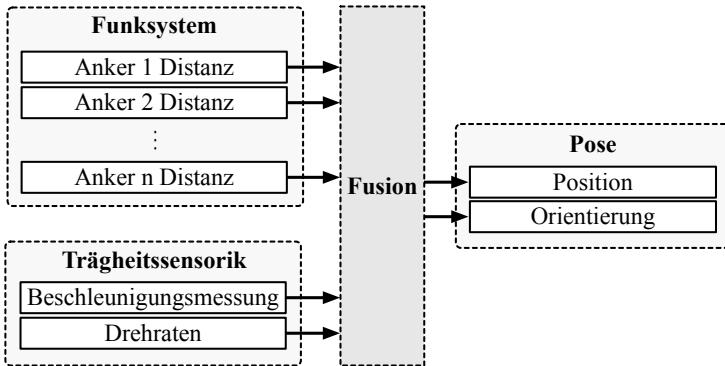


Abbildung 2.3: Fusion von Funk- und Trägheitssensordaten (engl. sensor fusion) zu einer Pose. Informationen fließen von links nach rechts im Bild.

und anderen Innenraum-Positionierungsanwendungen reduzieren viele Menschen oder sich bewegende Objekte in der Umgebung die Positionierungsgenauigkeit in dynamischen Situationen. Ungenaue und verzögerte Positionen wirken sich negativ auf das Spielerlebnis des Benutzers aus, da eine ungenaue Position fehlerhaft in der virtuellen Welt dargestellt wird und die Propriozeption, m.a.W. die Bewegungs- und Lagewahrnehmung des eigenen Körpers, verfälscht wird. Daher kann ein eigenständiges UWB-System in einem komplexen Innenraumumfeld mit vielen statischen und dynamischen Objekten in der Funkumgebung, die viele Mehrwegeausbreitungen verursachen, nur ungenaue oder gar keine Positionen liefern. Diese Nichtsichtverbindungen (engl. Non-Light-of-Sight, NLoS) werden bei der Funkübertragung in vorangegangenen Arbeiten meistens nicht berücksichtigt [486, 494, 171, 115].

Andererseits liefern Funksysteme eine langfristig stabile und sehr hohe Genauigkeit der Absolutposition, wenn eine klare und ungestörte Sichtverbindung zwischen Sender und Empfängern vorliegt [493].

Nach- und Vorteile der Sensorfusion

Die Funk und IMU Fusion besteht typischerweise aus zwei unabhängigen Subsystemen und jedes Subsystem ist für die separate Berechnung einer Positionsschätzung für die Fusion verantwortlich. So

2 Verwandte Arbeiten

erfordert die Funk- und IMU-Fusion, dass das Funksystem in der Lage ist, selbstständig eine Position zu schätzen. Dazu werden kontinuierlich mindestens drei stationäre Empfängerknoten benötigt. Dagegen leiden Subsysteme, die auf Trägheitsmessungen basieren, unter kumulativen Fehlern durch die Integraloperationen zur Geschwindigkeits- und Distanzbestimmung. Außerdem sind die meisten aktuellen Verfahren zur Rauschunterdrückung und Positions berechnung durch die Leistung der Hardware (HW) begrenzt [510].

In den meisten Fällen können die beiden Subsysteme, Funk- und IMU-Sensor, nicht direkt in eine HW-Plattform integriert werden, da bspw. bei PDR Verfahren die IMU am Fuß platziert werden muss und die Antenne des Funksystems für eine bessere Sichtverbindung am oberen Teil des Körpers platziert werden muss. Zhang et al. [510] verwenden einen Mahony-Komplementärfilter mit KF und ZUPT und identifizieren die wichtigsten Wendepunkte in der Bewegung, um den akkumulierten Fehler der IMU zu korrigieren (durchschnittlicher Fehler von 0.075 m auf einer Kreisbahn). Sie stellen fest, dass bereits die Dichte des menschlichen Körpers die Sicht zwischen Sender und Empfänger blockiert und einen kritischen Einfluss auf das Funk-Signal hat. Befindet sich die IMU und die Funkantenne bspw. an einem mobilen Computer [229], kann die IMU keine ZUPT Aktualisierungen durchführen, Fehler akkumulieren explosionsartig und die Position wird ungenau. Befinden sich beide Sensoren am Fuß [199], hat die Antenne eine schlechte Sichtverbindung, erzeugt eine erhebliche Anzahl von Ausreißern und die Position wird ungenau.

Nur wenige vorangegangene Arbeiten verwenden Techniken, die die IMU am Fuß und die Antenne an einem anderen Körperteil [352, 371] platzieren können. Diese Verfahren erhöhen die Kovarianz (engl. covariance) der Messung so drastisch, um die unbekannte relative Position zwischen Navigationspunkten zu berücksichtigen, dass die Positionsgenauigkeit der Fusion erheblich sinkt. Andere [25, 339] verwenden inverse Kinematik. Da hier jedoch die relativen Positionen der beiden Sensoren exakt sein müssen, müssen die Sensoren extrem fixiert und konstant platziert werden, sodass sie die vordefinierten (Gelenk)Winkel und (Knochen)Kanten zwischen den Sensoren exakt abbilden. Kann eine feste Platzierung nicht gewährleistet werden, sind diese Systeme nicht für Anwendungen mit einer hohen Dynamik,

bspw. in Sport oder in VR-Anwendungen, einsetzbar.

Die meisten vorgeschlagenen Positionierungstechniken werden für Anwendungen mit Fußgängern, die sich mit langsamer oder normaler Geschwindigkeit zwischen 0.0 m/s und 4.0 m/s bewegen, optimiert. Daher ist bei den meisten vorangegangen Arbeiten davon auszugehen, dass sie akkumulierende Fehler bei hohen Geschwindigkeiten nicht kompensieren können [116]. Der Hauptnachteil, der für Filtermechanismen für beide Subsysteme und deren Fusion gilt, ist die Notwendigkeit von Experten- und Domänenwissen, um diese korrekt zu konfigurieren und auf die jeweilige Anwendung spezifisch zu optimieren.

Da die IMU-basierte Lokalisierung nicht von OLoS und NLoS beeinträchtigt wird, kurzfristig sehr genaue Positionen schätzt, aber längerfristig große Fehler akkumuliert und funkbasierter Lokalisierung langfristig eine hohe Genauigkeit aufweisen, aber kurzfristig durch Mehrwegeausbreitung ernsthaft beeinträchtigt wird, ist es naheliegend, beide Technologien zu kombinieren [267]. Die Eigenständigkeit und Echtzeitfähigkeit von Trägheitssensoren können die dynamische Leistung eines integrierten Funk- und IMU-Positionierungssystems verbessern, nämlich immer dann, wenn die Positionsschätzung der Trägheitssensorik nur kurzfristig die Positionen des Funksystems stabilisieren muss, m.a.W. immer dann, wenn diese Komplementäreffekte bestmöglich ausgenutzt werden können. Funk- und IMU-Systeme werden häufig komplementär gepaart, sodass ihre jeweiligen Nachteile minimiert oder sogar beseitigt werden [195, 521, 86, 501, 113]. Viele Arbeiten haben diese Komplementäreffekte gezielt durch die Fusion von Funk und IMU untersucht [486, 494, 171, 115, 499]. Die meisten Fusionsmethoden [525, 526, 505, 72] nutzen am Fuß getragenen IMUs, um ZUPT zu nutzen.

Fazit: Die Fusion von Funk- und Trägheitssensoren ist immer dann sinnvoll, wenn das Rotations- und das Koordinatensystem bekannt sind und langfristig fehlerfrei miteinander ausgerichtet werden können. Diese Tatsache schränkt die Anwendbarkeit der Methoden der verwandten Arbeit stark ein, da die Sensoren häufig nicht starr und deterministisch an beweglichen Objekten befestigt werden können. Im Rahmen dieser Dissertation wird deshalb in Kapitel 6 eine Methode vorgeschlagen, die die Ausrichtung der Rotations- und Koordinatensysteme langfristig und zuverlässig stabilisiert, wenn Funk- und Trägheitsenso-

2 Verwandte Arbeiten

ren in unterschiedlichen dynamischen Rotations- und Koordinatensystemen angebracht werden.

2.2.3 Einfluss der Systemarchitektur

Prinzipiell sind mindestens drei stationäre Funkempfänger, sogenannte Anker, an bekannten Positionen als Infrastrukturunterstützung erforderlich, um mithilfe der klassischen Trilateration eine 2D-Position (ohne Höhe) zu erhalten. Abbildung 2.4 beschreibt die Trilateration der Ankunftszeiten bzw. Entferungen unter verschiedenen Ausbreitungsbedingungen. Diese Abhängigkeit führt zu einer logarithmischen Komplexität, die zu höheren Kosten und zu einer aufwendigeren Systembereitstellung für einen größeren Abdeckungsbereich führt, als andere Arten von drahtlosen Systemen, wie z. B. BLE und RFID [433]. Wenn die Sensorik vor dem Körper gehalten wird liefert das Funksystem durchschnittliche Positionsgenauigkeiten von 0.81 m, SD=0.70 m.

Um die Systemkomplexität zu reduzieren und dennoch die Positionsgenauigkeit zu erhöhen, gibt es einige Forschungsarbeiten [276, 431, 175, 429], die versuchen, die Anzahl nötiger Anker zu reduzieren oder deren Anordnung gegen die Genauigkeit der Positionsschätzung zu optimieren. Bspw. nutzen Hartmann et al. [171] ungleichmäßig verteilte UWB-Basisstationen mit mindestens 15 Ankerknoten für Bereiche über 150 m² und mindestens acht UWB-Ankern für kleine Bereiche. Allerdings können sie die Fehlerakkumulation in Bereichen ohne Anker nicht kontrollieren, sodass ihr System ungenau positioniert. Chen et al. [72] reduzieren die Positionierungsfehler im Vergleich zu nur IMU-basierter (Sensor am Fuß platziert) PDR und nur UWB-basierter Positionierung (Sensor an der Brust platziert) um 74.5% bzw. 43.5%. Sie erzielen diese Ergebnisse (durchschnittlicher Fehler von 0.15 m) mittels acht gleichmäßig verteilter UWB-Basisstationen. He et al. [175] und Tian et al. [429] verwenden nur UWB-Entfernungsmessungen zwischen einem mobilen Knoten und drei verfügbaren Empfängerknoten, um Multilateration mit (weniger als) drei UWB-Empfängern zu ermöglichen. Sie können den abgedeckten Bereich leicht erweitern und erreichen eine maximale Positionsgenauigkeit von 0.26 m. Ebenso ermöglichen Ruiz et al. [370] eine präzise Entfernungsmessung im Zentimeterbereich zwischen zwei Funk-Knoten für Entfernungen bis zu 100 m. Die Messfehler- und Unsicherheitsmodelle für die Abstandsmessungen

schwanken allerdings stärker ($SD > 0.71$ m) und reduzieren so die durchschnittliche Genauigkeit stärker. Tian et al. [276, 431] korrigieren mit ihrem UWB-IMU-Fußgängerverfolgungssystem mit nur einem UWB-Empfängerknoten den kumulativen Fehler von PDR (IMU am Fuß montiert). Da sie die Ankerposition nur einmal zu Beginn der Lokalisierung schätzen, wird die Positionsschätzung im Laufe der Zeit immer ungenauer (durchschnittlicher Fehler von 0.15 m bis 0.18 m). Giarré et al. [149] korrigieren die Orientierung einer UWB und IMU Fusion (Sensoren an der Taille platziert), diese wird mit abnehmender Anzahl an UWB Empfängern zunehmend schlechter.

Fazit: Die Komplexität der Systemarchitektur hat direkten Einfluss auf die Qualität der Lokalisierung. Weniger als drei LoS Sender-Empfänger-Verbindungen führen zu ungenauen Positionen. Zukünftige Erweiterungen dieser Dissertation könnten jedoch die Idee nutzen, Trägheitsmessfehler durch eine Abstandsschätzung zweier Funkanker zu korrigieren, um bspw. die Geschwindigkeitsschätzung oder die Posenschätzung direkt zu korrigieren. Um Studien übersichtlich zu halten und deren Parameter zu kontrollieren, befasst sich diese Arbeit nicht mit den Auswirkungen der Systemarchitektur auf die Genauigkeit und Zuverlässigkeit der Positions- und Posenschätzung. In Kapitel 4 werden mindestens 12 Empfänger genutzt, um eine genaue und zuverlässige Positionierung zu ermöglichen.

2.2.4 Einfluss zusätzlicher Kontextinformation

Zusätzlich zur Wahl des Verfahrens, das die höchste Positionsgenauigkeit erzielt, kann die Karte des Innenraums verwendet werden, um die Positionierungsgenauigkeit zu erhöhen. Typische Karten bestehen aus Grenzen und Positionen von Wänden und Hindernissen. Die Fusion der Karteninformation mit Sensordaten wird typischerweise unter Verwendung eines PFs erreicht. Nachdem sich die Partikel ausgebreitet haben, wird das Gewicht von Partikeln an ungültigen Positionen, die durch die Karte vorgegeben werden, z. B. außerhalb der Kartengrenze oder von Partikeln, die Wände und Hindernisse kreuzen auf Null gesetzt [346, 465]. Lee et al. [258] zeigen, dass sie mit SLAM die Informationen unbekannter UWB-Empfängerstandorte mit bekannten Bewegungsinformationen von Trägheitsmessungen fusionieren können, um so einen mittleren Positionsfehler von 2 m für menschliches Gehen auf gerader Linie zu gewährleisten. Allerdings

2 Verwandte Arbeiten

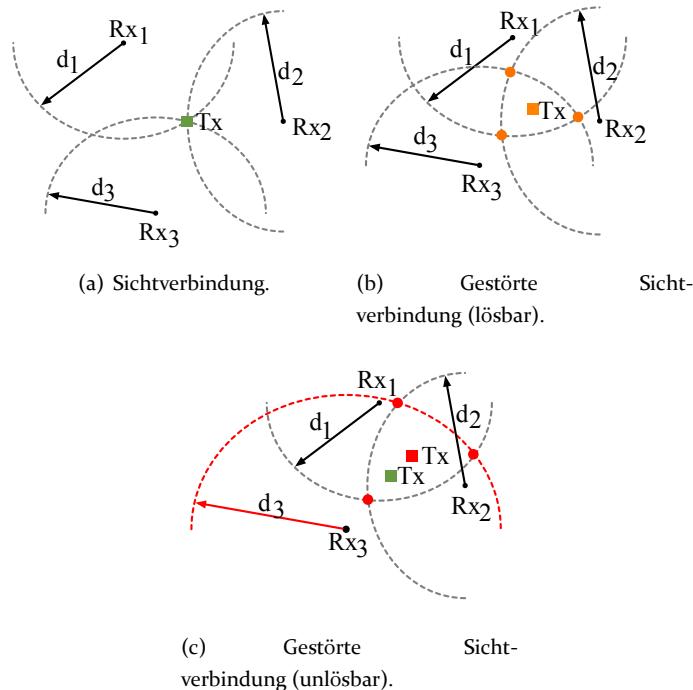


Abbildung 2.4: Positionsbestimmung mit Trilateration und Ankunftszeiten. Die Laufzeit eines Funksignals von einem einzelnen Sender (Tx) zu einem entfernten einzelnen Empfänger (Rx_i) ist ein Maß für die Entfernung (d_i) zwischen den beiden. Die Entfernung kann direkt aus der Ankunftszeit bestimmt werden, da sich ungestörte Signale mit Lichtgeschwindigkeit ausbreiten. ToA-Daten von zwei Basisstationen (Rx_1 und Rx_2) schränken die Position auf einen Positions Kreis ein. Daten von einer dritten Basisstation (Rx_3) werden benötigt, um die genaue Position auf einem Punkt zu verbessern. (a): Ohne Mehrwegeausbreitung treffen sich die Entfernungen (d_{1-3}) der Ankunftszeiten in einem gemeinsamen (grünen) Punkt. (b): Bei Mehrwegeausbreitung ändern sich die Entfernungen (d_{1-3}) bspw. aufgrund von Verzögerungen, die durch Ablenkung an Objekten entlang der Sender-Empfänger-Sichtlinie verursacht werden. Herkömmliche nichtlineare Optimierungsalgorithmen für die Trilateration treffen nicht unbedingt den gemeinsamen (orange farbenen) Schnittpunkt, da sich das optimale Zentrum, anders als hier gezeigt, als Funktion von d_3 verschieben kann. (c): reale Tx Position in grün stimmt dann nicht mehr mit der geschätzten Position Tx in rot überein da d_3 stark verzögert ist.

sind die Positionen in gültigen Kartenbereichen ungenau, da das Verfahren lediglich Ausreißer herausfiltert. Außerdem ist die Kartografie für Innenräume aufwendig und erhöht die Systemkosten erheblich.

Alternativ zur Kartografie kann die Form einer gültigen Bewegungstrajektorie in einer Innenumgebung zur Korrektur von Orientierungsfehlern verwendet werden, wenn eine ähnliche Form der Trajektorie erkannt wird [252]. Xu et al. [481] verwenden ein menschliches Bewegungsmodell für die untere Extremität des Menschen, das auf geometrischen Bewegungsmerkmalen (Denavit-Hartenberg-Parametern) basiert. Die Ergebnisse zeigen, dass der Positionierungsfehler und der Energieverbrauch niedriger sind als die der herkömmlichen Methoden. Die Sensoren müssen allerdings an den Füßen des Nutzers angebracht werden und individuell für jeden Nutzer konfiguriert werden.

Andere Vorarbeiten untersuchen explizit die Auswirkung der Sensorplatzierung auf die Positionsgenauigkeit. Chugunov et al. [80] platzieren die IMU- und UWB-Sensorik an der Taille. Allerdings sind die Ergebnisse (durchschnittlicher Fehler von 0.11 m bis 0.15 m) nicht reproduzierbar, da sie kein Referenzsystem verwenden. Im Gegensatz dazu platzieren Wang et al. [453] die Sensorik in der Hand. Sie können Geradeaus und Kurvenbewegungen in den IMU Daten klassifizieren und diese als semantische Informationen zusätzlich zu den RSSI Werte ihres UWB Systems nutzen. Allerdings liefert ihre Methode nur eine durchschnittliche Genauigkeit von 6 m. Außerdem müssen die Sensoren in der Hand vor der Brust möglichst starr gehalten werden. Xia et al. [476] platzieren die Sensorik auf der Brust und nutzen die Zeit für eine Rundreise (engl. Round Trip Time, RTT) des UWB Systems, um eine Genauigkeit von 0.13 m zu erreichen. RTT erhöht die Positionsgenauigkeit, reduziert aber die Teilnehmerzahl und führt eine höhere Latenz ein. Benzerrouk et al. [40] und Wen et al. [470] platzieren die IMU und UWB Sensorik an den Füßen, nutzen ZUPT, ZARU und KF und erzielen Positionsgenauigkeiten von 1.2 m auf 40 m Distanz bzw. 0.132 m, SD=0.05 m in reinen LoS Situationen. Xu et al. [487] platzieren die IMU und UWB Sensorik an unterschiedlichen Körperstellen (UWB im Nacken und IMU am Fuß). Ein adaptives KF nutzt ZUPT und erzielt Positionsgenauigkeiten von 0.5 m, SD=1.0 m. Zwar positionieren Methoden, die das Wissen über die Sensorplatzierung nutzen, genauer, aber sie fordern eine explizite, zeitaufwändige

und nicht alltagstaugliche Vorbereitung durch den Benutzer.

Fazit: Eine Idee dieser Dissertation basiert auf den hier vorgestellten Ideen, nämlich zusätzliche Kontextinformationen wie Umweltinformationen und Bewegungsmodelle zur Unterstützung der Lokalisierung zu verwenden. Die Lokalisierungsmethode in Kapitel 4 verwendet ToA-Werte, um die Position zu schätzen. Diese können durch Mehrwegeausbreitung gestört werden. Jede Störung spiegelt sich in veränderten Werten wider die zusätzliche räumliche Kontextinformationen bietet. Diese nichtlinearen Änderungen können als exogene Kontextinformationen verwendet werden, um implizit die räumliche Verteilung der Mehrwegeausbreitung für jede Position im Raum im datengetriebenen Modell abzubilden. Das Modell lernt implizit aus den Daten, in denen sich Objekte oder Störer im Raum befinden. Diese zusätzlichen Kontextinformationen führen möglicherweise zu einer hohen Positionsgenauigkeit. Zusätzlich verwenden die Kapitel 5, 6 und 7 implizit Bewegungsmodelle, indem der Bewegungspfad (z. B. Kreisbahn oder gerade Linie) und damit das Bewegungsmodell erfasst werden.

2.2.5 Einfluss von Mehrwegeausbreitung

Die Fusion einer IMU in einer PDR und UWB ist ein effektiver Weg, um eine hochpräzise Positionierung auch in Umgebungen ohne Sichtverbindung (NLoS) zu erreichen. Während UWB eine genaue absolute Positionierung unter Sichtlinienbedingungen bietet, sorgt PDR für eine kontinuierliche und gleichmäßige Trajektorie in Perioden mit UWB-Signalverlust. Der Hauptvorteil der Fusion von UWB und IMU liegt in der Positions korrektur in NLoS-Situationen durch die Trägheitssensoren. Die problematische langfristige Instabilität der Trägheitsnavigation aufgrund von Integrationsfehlern beeinträchtigt jedoch die Integrationsgenauigkeit effektiv und muss sorgfältig berücksichtigt werden. Ebenso einschränkend sind die NLoS-Situationen und Mehrwege des Funksystems. Die Genauigkeit der UWB-Entfernungsmessung wird hauptsächlich durch den NLoS-Zustand des UWB-Signalausbreitungspfads beeinflusst. Die meisten vorangegangenen Forschungsarbeiten konzentrieren sich auf Filteralgorithmen für die INS-Fehlerkorrektur, aber berücksichtigen nicht die durch Umwelteinflüsse verursachten Ausreißer in UWB-Systemen. Dennoch reduzieren die Ausreißer die Genauigkeit des integrierten Systems. Daher ist es notwendig, den Kopplungsgrad zu stärken und

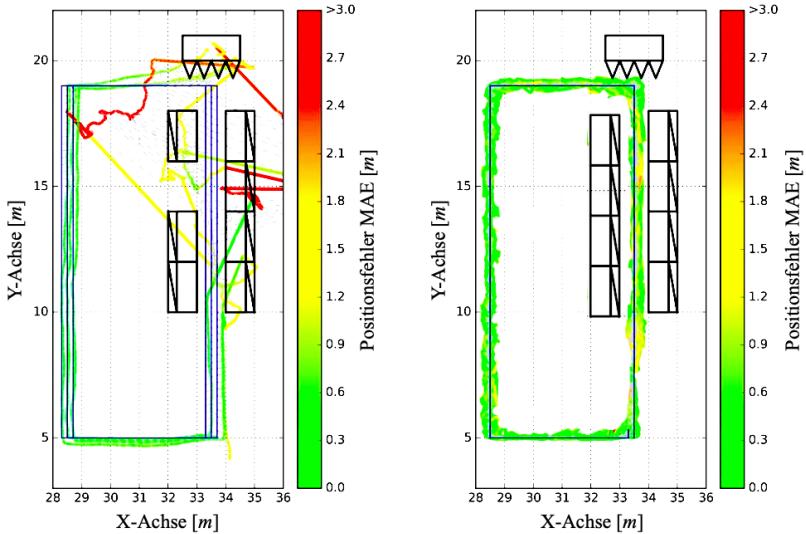


Abbildung 2.5: Positionsgenauigkeit eines Funklokalisierungssystems (in m). Links: unter Verwendung eines klassischen Bayesschen Filters. Unter massiven Mehrwegbedingungen (bis zum Verlust des Sichtkontaktes zwischen Sender und Empfänger oben rechts im Bild) unterliegt ein klassisches System zur Positionsschätzung großen Fehlern oder einem Versagen der Positions berechnung, da die Ankunftszeit des Übertragungssignals (engl. Time of Arrival, ToA) nicht korrekt bestimmt werden kann. Rechts: Ergebnisse einer Positionsbestimmung mit datengetriebenen Lernverfahren [313]: Der Ansatz kann die Mehrwegeinformationen in den ToA Werten auch bei massiven Abschattungen (rechts oben) verwenden, um die wahre Position zuverlässig und präzise zu bestimmen. Rekurrente neuronale Netze [127] lernen die Bewegung des Objekts, was zu genauen Bewegungsmodellen und damit zu genauesten Positionen führt (Kapitel 4).

die Wechselwirkungskorrektur beider Subsysteme zu realisieren [115].

Um störende Messfehler bei der UWB-Entfernungsmessung zu vermeiden, wird der KF-basierte INS- und UWB-Fusionsansatz in [115] mit einem Schema zur Entfernung NLoS-bedingter Ausreißer optimiert und in einem Bereich von $6 \text{ m} \times 8 \text{ m}$ evaluiert. Techniken zur Fehlerminde rung werden in vorangegangenen Arbeiten mit verschiedenen Messfehlermodellen untersucht [370, 100, 136, 146, 318]. Die Genauigkeit der Messfehler- und Unsicherheitsmodelle ist ein kritischer Faktor für die Positionierungsgenauigkeit, die mit der Zeit

2 Verwandte Arbeiten

abnimmt, wenn kein Korrektur-Mechanismus verwendet wird.

Um dennoch genaue Positionen in NLoS-Situationen mit verzögerten Messungen zu ermöglichen, werden die Ausreißer-Zurückweisung [194, 195] und robuste EKF-Formulierungen [167] verwendet. Ciurana et al. [82] identifizieren NLoS-bedingte Ausreißer mit einer Positionsgenauigkeit von 0.9 m. Xu et al. [486] entwickeln einen unverzerrten Filterbankalgorithmus mit endlicher Impulsantwort, um die Zeitdifferenz zwischen der INS- und der UWB-geschätzten Position zu korrigieren (Fehler >3 m). Auch Zwirello et al. [526] (Fehler >1.5 m) und Hol et al. [195] (Fehler >1.0 m) integrieren UWB- und IMU-Beobachtungen mithilfe eines EKFs und zeigen, dass das System robust und stabil ist, obwohl es von Mehrwegepfaden und NLoS beeinflusst wird. Hol et al. [195] zeigen, dass das System von Xu et al. jedoch nicht in praktischen Anwendungen unter NLoS-Bedingungen funktioniert. Sczyslo et al. [393] kombinieren UWB und IMU, um PDR auch in NLoS Situationen mit einer Genauigkeit von 0.67 m zu ermöglichen. Zhong et al. [517] erkennen NLoS Situationen bei niedrigen Bewegungsgeschwindigkeiten (2 m/s) eines Handwagens und erzielen einen durchschnittlichen Positionsfehler von 0.10 m, SD=0.12 m. Kok et al. [241] formulieren ein Optimierungsproblem, um UWB-Messungen zu entfernen, die von Mehrwegepfaden und NLoS betroffen sind. Dies eröffnet auch die Möglichkeit, nicht-Gaußsche Verteilungen auszunutzen und zusätzliche Parameter zu schätzen. Kok et al. [242] kombinieren eine IMU (am Fuß platziert) mit UWB und erzielen einen durchschnittlichen Fehler von 0.03 m in NLoS Situationen. Allerdings nutzen sie ein markerbasiertes Referenzsystem, das typischerweise deutlich höherer Positionsfehler verursacht.

Fazit: In der Tabelle 2.2 werden herausragende Arbeiten aufgelistet und die verwendeten Sensoren, Methoden, ihre Genauigkeit und Nachteile zusammengefasst. Die besten der vorgeschlagenen Methoden erreichen genaue Positionen, wenn die Sensoren starr am Fuß befestigt sind, hochkomplexe Bewegungsmodelle implementiert sind und die Sensoren eng gekoppelt sind (Abschnitt 2.3.1 erläutert die enge und lose Kopplung). Im Gegensatz zu den meisten verwandten Arbeiten werden in den Experimenten dieser Dissertation (in Kapitel 4) explizit die Mehrwegeausbreitung in der realen Welt erzwungen, um deren Auswirkungen auf die unterschiedlichen Verfahren genau zu untersuchen. Die datengetriebenen Verfahren können diese räumlichen Informatio-

nen sehr gut nutzen, um die Positionsgenauigkeit über den Stand der Technik hinaus zu verbessern, selbst wenn die verschiedenen Sensoren lose gekoppelt sind (siehe Kapitel 7). Abbildung 2.5 zeigt die Genauigkeit, mit der typische modellbasierte und datengetriebene Verfahren in NLoS-Situationen positionieren.

2.3 Methoden zur Kopplung von Funk- und Trägheitssensoren

Abschnitt 2.3.1 diskutiert zunächst die Effekte von enger und loser Kopplung auf die Fusion von Funk- und Trägheitsmessungen. Dann werden Methoden diskutiert, die Funk- und Trägheitssensoren fusionieren, um Positionen oder Posen zu schätzen: hochspezialisierte Methoden (Abschnitt 2.3.2), KF (Abschnitt 2.3.3), PF Abschnitt 2.3.4) und datengetriebene Methoden (Abschnitt 2.3.6).

2.3.1 Enge und lose Kopplung

Zur Fusion von Trägheits- und Funkmessungen wird typischerweise ein eng oder lose gekoppelter Ansatz verwendet. In der Literatur ist die sogenannte lose gekoppelte Integration beider Systeme eine beliebte Option. In diesem Fall werden die Lokalisierungslösungen zuerst separat berechnet und dann vom Fusionsfilter verwendet. Die Position, die nur die Funksignale verwendet, wird dann vom Fusionsfilter als Positionsaktualisierung verarbeitet. Die Geschwindigkeit, die nur die Trägheitsmessungen verwendet, wird dann vom Fusionsfilter als Geschwindigkeitsaktualisierung verarbeitet. Das Fusionsfilter verarbeitet beide Aktualisierungen zu einer Pose. Zwirello et al. [526] behaupten, dass dies die einfachste, aber nicht die genaueste Methode ist. Probleme treten auf, wenn nicht genügend Empfänger erreichbar sind und eine Funklokalisierung unmöglich ist. Das Hauptproblem ist die asynchrone Verarbeitung der Sensoren, die zu zeitlich divergierenden Datenströmen und einer ungenauen Fusion führt, da herkömmliche Verfahren diese a priori definieren müssen. Schon eine kleine Abweichung führt zu schwerwiegenden Fehlern. Im Gegensatz dazu ist die eng gekoppelte Integration der Sensordaten vorteilhafter, da hier beide Sensorströme direkt miteinander synchronisiert sind. Diese Variante schränkt jedoch die Vielfalt und Flexibilität der möglichen Hardwareplattformen dramatisch ein und kann nicht auf typische Anwendungsbereiche, wie

2 Verwandte Arbeiten

Tabelle 2.2: Ergebnisse renommierter verwandter Arbeiten der letzten fünf Jahre, die Trägheitsmessungen mit Breitband-Hochfrequenzfunksystemen kombinieren, die für den Innenbereich geeignet sind. Übersicht über verwendete Sensoren, Methoden, höchstmögliche Genauigkeit oder geringste Fehler **fettgedruckt** und Nachteile.

Autoren	Sensortyp	Methode	Fehler in [m] *	Nachteile
Sasiadek et al. [385]		KF	2.97	
Vydhyanathan et al. [447]	GPS	KF	0.51	Die hohe Systemkomplexität führt im Übergang zwischen Außen- und Innenbereichen zu sehr hohen Ungenauigkeiten.
Belakbir et al. [35]	UWB	KF	0.37	
Wang et al. [456]		KF	0.20	
Tian et al. [433]	UWB IMU	KF	0.81	Hohe Systemkosten. Mindestanzahl an Sichtverbindungen limitiert die Minimalkonfiguration. Genauigkeit sinkt bei Mehrwegeausbreitung stark.
Zampella et al. [505]		PDR	2.18	Orientierungsfehler akkumulieren während des Abbiegens erheblich, sodass die Positionsgenauigkeit stark sinkt.
Li et al. [262]		PDR	0.80	Hohe Genauigkeiten nur bei statischen linearen Bewegungen.
Hartmann et al. [171]	UWB	KF (8 Anker)	2.49	Hohen Ungenauigkeiten in Bereichen ohne Rx .
He et al. [175]	IMU	KF (4 Anker)	0.26	Hohe Genauigkeit nur unter LoS bei geringer Reichweite.
Ruiz et al. [370]		KF (2 Anker)	1.9	Hohe Varianz der Positionsgenauigkeit.
Chen et al. [72]		PDR (8 Anker)	0.15	Hohe Genauigkeit, wenn IMU starr am Fuß und UWB an der Brust platziert werden.
Tian et al. [276, 431]		PDR (1 Anker)	0.15 bis 0.18	IMU und UWB muss starr am Fuß befestigt werden. Die Sichtverbindung des Tx am Fuß wird häufig unterbrochen und führt zu Positionsfehlern.
Lee et al. [258]	UWB	SLAM	2.00	Aufwendige Kartographie zur Laufzeit erhöht die
Qian et al. [346]	IMU	PF	0.82	Positionsgenauigkeit nur, wenn die
Waqrar et al. [465]	Karte	PDR	0.45 bis 0.74	Sensormessungen genau sind. **
Xu et al. [481]		PF	0.08 bis 0.42	IMU muss starr an beiden Füßen platziert werden. Sehr komplexes Bewegungsmodell. **
Wang et al. [453]	UWB	KF	6.0	IMU muss starr an der Taille platziert werden. **
Chugunov et al. [80]	IMU	KF	0.11 bis 0.15	IMU muss in der Hand vor der Brust gehalten werden. **
Xia et al. [476]	Kontext	KF	0.13	IMU muss starr an der Brust platziert werden. **
Benzerrouk et al. [40]		KF	1.2	IMU muss starr an beiden Füßen platziert werden. **
Wen et al. [470]		KF	0.13	IMU muss starr an beiden Füßen platziert werden. **
Xu et al. [487]		KF	0.50	IMU muss starr am Nacken platziert werden. **
Lan et al. [252]		PF	0.48	IMU muss starr an beiden Füßen platziert werden. **
Ciurana et al. [82]		KF	0.90	IMU muss starr am Fuß platziert werden.
Kok et al. [241, 242]	UWB	EKF	0.03	IMU muss starr am Fuß platziert werden.
Sczyslo et al.	IMU	PDR	0.67	IMU muss starr am Fuß platziert werden.
Zhong et al. [517]	NLoS	Speziell	0.10	IMU muss starr an einem Rollwagen befestigt werden.
Hol et al. [195]		SINS EKF	>1.0	Keine Positionsgenauigkeiten berichtet. Sehr
Zwirello et al. [526]		EKF	>1.5	unpraktischer eng gekoppelter Ansatz.

*) Die Ergebnisse, die zur Bewertung der verwendeten Methoden verwendet wurden, werden bestmöglich aus der jeweiligen Darstellungsform extrahiert.

**) Die Ergebnisse wurden unter Laborbedingungen unter Funkausbreitungsbedingungen mit klarer Sichtlinie (LoS) erhalten.

z. B. Sensoren in der Hosentasche, am Kopf, am Fuß oder am Ohr, übertragen werden. Abbildung 2.6 stellt exemplarische Verarbeitungsketten einer engen und einer losen Kopplung mit denselben Sensoren dar.

Lose Kopplung

Bei einem lose gekoppelten Ansatz werden die Funk- und die Trägheitsmessungen getrennt und unabhängig voneinander betrachtet und für beide Sensoren jeweils separat und individuell Positionen lose geschätzt. Typischerweise werden in einem ersten Schritt Funkmessungen verwendet, um Positionsschätzungen unter Verwendung eines Multilaterationsansatzes zu erhalten. In einem zweiten Schritt werden aus den Trägheitsmessungen Positionen geschätzt. Anschließend werden diese Positionsschätzungen in einem weiteren Schritt miteinander zu einer gemeinsamen Position fusioniert, um sowohl kurz- als auch langfristige Positionsgenauigkeit zu gewährleisten [135, 393, 340].

Typische Verfahren, die lose gekoppelt fusionieren, platzieren die UWB und Trägheitssensoren getrennt voneinander, bspw. an unterschiedlichen Körperstellen. So werden UWB Sensoren für eine möglichst freie Sichtverbindung häufig am Kopf angebracht und IMU Sensoren an den Füßen, um ZUPT Verfahren zu ermöglichen. Fan et al. [115] und Sczyslo et al. [393] entwickeln lose gekoppelte Fusionsmethoden basierend auf dem EKF, um die Fußgängerbewegung zu verfolgen. Für diese Methoden sind jedoch mindestens drei hochgenaue UWB-Entfernungsmessungen erforderlich, die unter den NLoS-Bedingungen möglicherweise nicht verfügbar sind. Sczyslo et al. [393] zeigen, dass die Trägheitsnavigationsschätzungen einer losen Kopplung nicht genau genug sind, um notwendigerweise den korrekten Pfad der UWB-Kanalimpulsantwort zu bestimmen. Daher entwickeln sie eine Mischung aus enger und loser Kopplungsarchitektur an, um die Informationen von UWB und IMU zusammen zu führen und die Genauigkeit des Systems unter NLoS-Bedingungen zu verbessern. Auf dieser Basis wurde eine Vielzahl weiterer loser Kopplungsarchitekturen erforscht [38, 39, 494, 522]. Die Genauigkeit der losen Kopplung ist bei allen vorgeschlagenen Systemen (durchschnittliche Fehler von mindestens >0.52 m) deutlich niedriger als mit enger Kopplung.

Enge Kopplung

Bei einem eng gekoppelten Ansatz werden dieselben Arbeitsschritte zur Fusion wie bei einem lose gekoppelten Ansatz ausgeführt, nämlich Funkpositions-, Geschwindigkeits- und gemeinsame Posenschätzung. Allerdings werden immer zusätzlich die rohen Messwerte zur Fusion zu einer gemeinsamen Position verwendet und die Sensorik befindet sich auf einer gemeinsamen Plattform, deren technischer Aufbau die Datenströme implizit zeitlich synchronisiert.

Typische Verfahren, die lose gekoppelt fusionieren, platzieren die UWB und Trägheitssensoren in einem gemeinsamen Sensor an einer Körperstelle, um Übertragungszeiten und -aufwand möglichst kurz und gering zu halten. Viele Arbeiten entwickeln eng gekoppelte Methode, die die UWB- mit den IMU-Messungen kombinieren [484, 485, 114]. Xu et al. [484, 485] nutzen die enge Kopplung und liefern im Freien mit langsam Bewegungen, ohne schnelle und abrupte Bewegungsänderungen, einen durchschnittlichen Fehler von über 0.20 m, wenn GPS, UWB und IMU verfügbar sind. Ascher et al., Zampella et al. verwenden die enge Kopplung, um die Positionierung von Fußgängern in Innenräumen zu realisieren [21, 505]. Fan et al. dagegen realisieren die 2D-Positionierung mobiler Roboter in Innenräumen [114]. Cheng et al. [74] schätzen eine 3D-Position in Innenräumen. Aber diese vorgeschlagenen Methoden sind aufgrund ihrer unhandlichen großen Hardwarestrukturen nicht alltagstauglich. Wang et al. [456] entwickeln ein eng gekoppeltes experimentelles GPS, UWB und INS System, das auf einem adaptiven robusten KF basiert und sowohl in Innen- als auch in Außenräumen positioniert (durchschnittlicher Fehler >0.71 m). Die meisten vorangegangenen Arbeiten [526, 195, 97] integrieren UWB- und IMU-Messungen mithilfe eines EKFs in enger Kopplung und zeigen, dass das Verfahren auch in NLoS Situationen robust und stabil ist. Der Vorteil des eng gekoppelten Ansatzes besteht darin, dass er nicht unter dem Informationsverlust leidet, der durch die lose Vorverarbeitung (engl. pre-processing) der UWB-Messungen entstehen kann. Häufig werden einzelne ToA-Messungen eng gekoppelt [195, 97]. Allerdings werden durch die enge und starre Kopplung häufig Messungen ignoriert, bspw. wenn nicht genügend ToA-Messungen in NLoS Situationen für die Multilateration vorhanden sind [526]. So kann es dazu kommen, dass, obwohl die Positioniergenauigkeit und -stabilität durch eng gekoppelte Verfahren prinzipiell verbessert werden, das Signalrauschen und die Unterbre-

2.3 Methoden zur Kopplung von Funk- und Trägheitssensoren

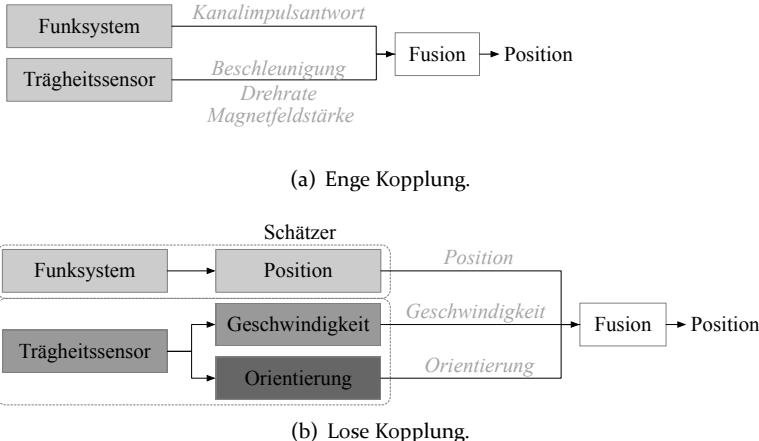


Abbildung 2.6: Enge und lose Kopplung: Die enge Kopplung fusioniert die rohen Sensorinformationen in einer zentralen Komponente zu einer gemeinsamen Position. Die lose Kopplung fusioniert die rohen Sensorinformationen unabhängig voneinander direkt am Sensor und fusioniert diese in einer zentralen Komponente zu einer gemeinsamen Position. (Unterschiedliche Grautöne deuten die Unabhängigkeit der Sensoren und Informationen an).

chung in NLoS Situationen nicht berücksichtigt werden und gar keine Position geschätzt werden kann.

Vergleich lose und enge Kopplung

In einigen Studien wird das UWB-System mit dem INS-System unter Verwendung eng gekoppelter [195, 521] und lose gekoppelter [86] Methoden fusioniert, um Trägheitsverfolgungsfehler mit den fehlerfreien Schätzungen des UWB-Systems zu beheben und Fehler des UWB-Systems über kurze Zeiträume durch eine hohe Abtastrate der Trägheitsverfolgung zu korrigieren, um eine kontinuierliche Trajektorie bereitzustellen. Ein hybrider Ansatz identifiziert Funksignale a priori lose, die NLoS Komponenten enthalten, und koppelt dann UWB- und IMU-Sensoren eng zusammen [82]. Zwirello et al. [525] vergleichen die EKF-basierte lose und eng gekoppelte Integration von UWB und INS basierend auf dem PDR-Algorithmus. Bei diesem Verfahren ist die Positionierungsgenauigkeit jedoch weitgehend von der Genauigkeit des

2 Verwandte Arbeiten

PDR-Algorithmus selbst abhängig. Hartmann et al. [171] vergleichen ein enges und ein loses UWB-INS-Lokalisierungssystem und zeigen, dass das lose System viel genauere Positionen als das enge schätzt (durchschnittlicher Positionierungsfehler des schmalen Systems von 3.06 m im Vergleich zu 2.68 m des losen Systems). Eine spezielle Variante, die sogenannte Tiefenkopplung, führt die Ergebnisse der endgültigen Fusion in die Komponenten der gelieferten Funkpositions- und Geschwindigkeitsschätzung zurück, um diese zu stabilisieren. In einem lose gekoppelten System ist die Tiefenkopplung in der Praxis schwierig zu implementieren, da jeder Knoten senden und empfangen muss und wurde in verwandten Arbeiten nicht berücksichtigt.

Fazit: Aus der verwandten Arbeit ist nicht eindeutig ersichtlich, ob eine lose oder enge Kopplung zu einer optimalen Fusion der Pose führt. Eine enge Kopplung bietet Vorteile, wenn die Sensoren zentral und lokal auf einer Plattform synchronisiert und fest definiert werden können. Eine lose Kopplung ermöglicht andererseits mehr Flexibilität in der Systemstruktur und erfordert erheblich mehr Vorsicht beim Umgang mit nicht synchronisierten und fehlerhaften Datenströmen. Moderne Methoden, die mit Bayesschen Filtern fusionieren und Systeme erfordern, die im Voraus fest definiert sind, müssen daher eng gekoppelte Strukturen verwenden. Im Rahmen der Dissertation wurde ein Verfahren entwickelt, das sowohl eine lose als auch eine enge Kopplung ermöglicht. Die Schätzungen von Positionen (Kapitel 4), Geschwindigkeiten (Kapitel 5) und Orientierungen (Kapitel 6) werden auf unabhängigen und entfernten Sensoren geschätzt, die lose gekoppelt sind. Z. B. schätzt ein Mobiltelefon in der Hosentasche die Geschwindigkeit und ein Hauptrechner in der Umgebung schätzt die Funkposition. Die Schätzung der Pose (Kapitel 7) wird dann auf einem gemeinsam genutzten oder einem anderen unabhängigen Sensorknoten durchgeführt. Z. B. fusioniert dasselbe Mobiltelefon die gesammelten Geschwindigkeiten mit den Funkpositionen die der Hauptrechner versendet.

2.3.2 Hochspezialisierte Methoden

In diesem Abschnitt werden hochspezialisierte Methoden der adaptiven Filterfamilie, Methoden zur Graphoptimierung und anwendungsspezifische Methoden diskutiert, die typischerweise zur Fusion von Sensordatenströmen verwendet werden.

Adaptivfilter

Der adaptive Filteralgorithmus von Sage-Husa et al. [379] schätzt und korrigiert statistische Rauschmerkmale. Liu et al. [278] verwenden ein Sage-Husa-Fuzzy-Adaptivfilter, um das zeitvariable Rauschproblem zu lösen. Guo et al. [161] wenden das adaptive Sage-Husa KF auf ein integriertes System für Trägheits- und GPS-Messungen an. Auf Basis von Sage-Husa führen Song et al. [409] einen gewichteten adaptiven Filteralgorithmus für autonome Funksysteme ein, der das Problem der negativen Rauschkovarianzmatrix adressiert. Der Algorithmus schätzt jedoch aufwendig die Systemrausch-Kovarianzmatrix und die Messrausch-Kovarianzmatrix für jeden einzelnen Filterprozess, sodass er nicht echtzeitfähig ist.

Obwohl alle bekannten adaptiven Algorithmen die Schätzgenauigkeit im direkten Vergleich zu herkömmlichen KF verbessern, erhöhen sie mit ihrer Komplexität jedoch den Rechenaufwand dramatisch und befinden sich aufgrund der hohen Dimension der Informationen in einem gekoppelten Sensorsystem häufig in einem kritischen, instabilen Zustand, der letztendlich immer zu Filterdivergenzen führt.

Graphoptimierung

Alle Bayessche Filter, der Kalman- und Partikel-Filter Familien, berücksichtigen nur die Beziehung zwischen dem aktuellen Zustand und dem vorherigen Zustand. Da sie die Informationen aus allen vorherigen Zuständen und die entsprechenden Beobachtungen nicht nutzen, nimmt folglich die Positionsgenauigkeit bei kontinuierlicher größerer Beobachtung oder systematischem Fehler schnell ab [525]. Um dieses Problem zu lösen, schlagen Kaess et al. [219] einen Graphoptimierungsalgorithmus vor, der keine Markov-Annahme trifft, sondern direkt den vorherigen Zustand beibehält. Die Fehlerfunktion des Algorithmus betrachtet alle Zustände als Variablen und Bewegungsgleichungen und Beobachtungsgleichungen als Einschränkungen zwischen Variablen. Eine nichtlineare Optimierung minimiert dann den Fehler.

Obwohl der Graphoptimierungsalgorithmus die Positionierungsgenauigkeit eines Fusionsalgorithmus verbessern kann, ist er mit einem relativ hohen Rechenaufwand verbunden und ermöglicht keine echtzeitnahe Positionierung [186]. Kaess et al. [220] erforschen einen Algorithmus zur adaptiven inkrementellen Glättung von Verzögerungen (engl. Incremental Smoothing and Mapping 2, ISAM₂). Im Vergleich

2 Verwandte Arbeiten

zur aufwendigen Berechnung und Aktualisierung aller historischen Zustände bei der Optimierung von Stapel-Graphen aller herkömmlichen Graphoptimierungsalgorithmen kann ISAM2 adaptiv beurteilen, ob die aktuelle Messung einen bestimmten historischen Zustand beeinflusst und somit die Notwendigkeit einer Berechnung und Aktualisierung entscheiden, wodurch unnötige Berechnungen verringert werden [463]. Strasdat et al. [420] zeigen, dass der ISAM2-Algorithmus genauer und effektiver Positionsschätzungen bei Fusion mehrerer kamerabasierter Sensoren ermöglicht, als nichtlineare EKF und PF. Diese Methoden können jedoch nur für Anwendungen mit endlichem Zustands- und Suchraum verwendet werden und sind somit nicht zur Positionierung zufälliger (undefinierbarer) menschlicher Bewegungen geeignet.

Anwendungsspezifische Filter

Die einfachsten Filter von Xu et al. [484, 485] und Fan et al. [114] nutzen einfache Schwellenwertmethoden um abnormale Messungen aus ihrem Fusionsverfahren zu entfernen und verbessern die Positionierungsgenauigkeit von herkömmlichen Verfahren nicht.

Ein ebenso einfaches Filter entwickeln Fan et al. [115]. Ihr komplementäres Filter korrigiert die Ausrichtung der Trajektorie. Eine besonders kostspielige Kombination aus einem KF und einem antimagnetischen Filteralgorithmus eliminiert Ausreißer aus einem UWB-System in NLoS-Umgebungen. Obwohl sie die Genauigkeit der Position effektiv verbessern (durchschnittlicher Fehler 0.09 m, SD=0.21 m), sind die abnormalen Daten immer noch vorhanden und erzeugen Akkumulationsfehler im UWB-System. Der Ansatz muss für jede Anwendung und jedes verfolgbare Objekt sorgfältig und kostspielig angepasst werden und ist nicht echtzeitfähig. Es können nur langsame Roboterbewegungen ohne schnelle und abrupte Bewegungsänderungen auf einem rechteckigen Pfad lokalisiert werden.

Zampella et al. [505] begrenzen das Fehlerwachstum ihrer UWB-(Sensoren am Kopf, durchschnittlicher Fehler RMSE⁶: 0.3 m bis 1.25 m)IMU-(Sensoren am Fuß) und PDR-Fusion mit ZUPT und erreichen eine Genauigkeit zwischen 0.2 m und 0.5 m. Das Verfahren ist sehr rechenintensiv, berücksichtigt keine Nichtsichtverbindungen und die Sensoren bewegen sich mit einer konstanten Geschwindigkeit von 0.1 m in einer konstanten Höhe von 1.8 m, was für die mensch-

⁶Mittleres Abweichungsquadrat, engl. Root-Mean-Square Error, RMSE.

2.3 Methoden zur Kopplung von Funk- und Trägheitssensoren

Tabelle 2.3: Ergebnisse renommierter verwandter Arbeiten der letzten fünf Jahre, die Trägheitsmessungen und Funksignale mit speziellen Methoden fusionieren. Übersicht über verwendete Sensoren, Methoden, höchstmögliche Genauigkeit oder geringste Fehler **fettgedruckt** und Nachteile.

Autoren	Sensortyp	Methode	Fehler in [m] [*]	Nachteile
Sage-Husa et al. [379]	Statistisches Rauschen	Sage-Husa (SH-KF)	-	Hohe Komplexität; Hoher Rechenaufwand; Hohe Dimension der Daten führt zu Filterdivergenz.
Liu et al. [278]	Zeitvariables Rauschen	SH-Fuzzy Adaptivfilter	-	
Guo et al. [161]	GPS IMU	SH-KF	-	
Song et al. [409]		Gewichtetes Adaptivfilter	-	
Kaess et al. [220, 219]	Kameras	ISAM2 Graph-Optimierung	-	Hoher Rechenaufwand. Nicht echtzeitfähig.
Strasdat et al. [420]			0.37	Genauer aber rechenaufwendiger als KF.
Xu et al. [484, 485]		Schwellenwert	3.18	Sehr ungenaue Schätzungen.
Fan et al. [114]		Schwellenwert	2.76	Hohe Positionsgenauigkeit.
Fan et al. [115]		Komplementär-Filter	0.09	Aufwendige individuelle Konfiguration; Nur bei langsam Bewegungen genau.
Zampella et al. [505]	UWB IMU	Spezielle PDR	0.3 bis 1.25	Komplexe Systemarchitektur; Nur bei langsam Bewegungen genau.
Gentner et al. [147]		Spezielle PDR	0.20 **	
Zeng et al. [508]		Spezielle PDR	2.38	Hohe Positionsgenauigkeit.
Xu et al. [482]		Spezielles SINS	0.81 **	
Xu et al. [484]		Spezielles SINS	0.21 **	
Tanigawa et al. [425]		Spezielles SINS	0.23 **	
Fan et al. [115]		Speziell	0.98	Aufwendiger unpraktischer An-satz.

^{*}) Die Ergebnisse, die zur Bewertung der verwendeten Methoden verwendet wurden, werden bestmöglich aus der jeweiligen Darstellungsform extrahiert.

^{**}) Die Ergebnisse wurden unter Laborbedingungen unter Funkausbreitungsbedingungen mit klarer Sichtlinie (LoS) erhalten.

liche Bewegung untypisch ist, da sich die Kopfposition während eines Gangzyklus ändert. Darüber hinaus berücksichtigen sie nicht, dass Kopf und Körper unterschiedlich zueinander ausgerichtet werden können und sich Bewegungen schnell und abrupt ändern können. Gentner et al. [147] verwenden einen iterativen Fußgängerlokalisierungsalgorithmus der keine Informationen über die UWB-Ankerpositionen benötigt und Genauigkeiten von 0.20 m bei direkten Sichtverbindungen ermöglicht. Fan et al. [115] nutzen einen aufwendigen dreistufigen Ansatz zur Verbesserung der Gesamtleistung (0.98 m) ihres UWB-IMU-Systems. Zeng et al. [508] nutzen einen infrastrukturfreien Ansatz. Sowohl IMU- als auch UWB-Knoten wurden an Schuhen angebracht. Ihre Schätzung nutzt ZUPT, ZARU und eine heuristische Orientierungsfehlerreduktion, mit der die IMU- und UWB-Messfehler kompensiert werden. Obwohl das System akkumulierenden Fehler des Beschleunigungsmessers reduziert, liefert es ungenaue Positionen 2.38 m.

Im Gegensatz zu diesen speziellen PDR-basierten Verfahren nutzen Xu et al. [482] eine SINS-basierte Optimierungsmethode basierend auf dem Chebyshev-Zentrum, um die Fehlerakkumulation der IMU weitgehend zu kompensieren (Fehler 0.81 m) und diese zusammen mit ToA-Informationen zur Positionierung zu nutzen. Die Genauigkeit in NLoS Situation wird leider nicht untersucht. Xu et al. [484] und Tanigawa et al. [425] schlagen ein integriertes UWB-IMU-System nur mit Reichweitemessungen der UWB vor und erreichen Genauigkeiten von 0.20 m unter Sichtverbindungen. Da SINS Modelle sehr eng gekoppelten sind und starr a priori definiert werden müssen, lassen sich kaum auf alltägliche Anwendungen generalisieren.

Zhao et al. [514] nutzen ein außergewöhnliches FIR-Filter (Finite Impulse Response). Xu et al. [486] machen das Verfahren von Zhao et al. robuster, indem sie einen unverzerrten Filter mit endlicher Impulsantwort (engl. Unbiased Finite Impulse Response, UFIR) nutzen. Beide Verfahren liefern sehr ungenaue Positionen (Fehler > 0.76 m), selbst bei direkten Sichtverbindungen.

Guo et al. [162] lokalisiern gleichzeitig mehrerer mobile Roboter kooperativ. Allerdings funktioniert ihr Ansatz nur, solange sich die Roboter bewegen und liefert dann genaue Positionen (Fehler < 0.33 m).

Fazit: Die vorgestellten Verfahren sind hochgradig anwendungsspezifisch und für den praktischen Einsatz ungeeignet. Die meisten dieser Filter sind darüber hinaus auch noch sehr komplex und die Reproduktion und Wiederverwendbarkeit sind unmöglich. In der Tabelle 2.3 wer-

den herausragende Arbeiten aufgelistet und die verwendeten Sensoren, Methoden, ihre Genauigkeit und Nachteile zusammengefasst. Der Vergleich der Fehler in Tabelle 2.2 mit Tabelle 2.3 zeigt, dass die hochspezifischen Methoden in Tabelle 2.3 keine präzisere Fusion ermöglichen als herkömmliche Methoden.

2.3.3 Kalman-Filter

Das KF ist das am häufigsten verwendete Verfahren, um Funk- und Trägheitsmessungen zu fusionieren. Vorangegangene Arbeiten verwenden meistens das klassische lineare Kalman-Filter (LKF), um lineare Systeme oder das erweiterte Kalman-Filter (EKF), um nichtlineare Systeme zu schätzen. Letzteres wird in einigen wenigen Ansätzen für hochgradig nichtlineare Systeme auch um spezielle Funktionen für spezifische Anwendungen erweitert, bspw. Sigma-Punkt Kalman-Filter (engl. Unscented Kalman-Filter, UKF) oder Kubatur Kalman-Filter (engl. Cubature Kalman-Filter, CKF).

Lineare Kalman-Filter (LKF)

Corrales et al. [85] schätzen mit einem LKF die Bewegungen von Gliedmaßen eines Menschen. Allerdings ist die Lokalisierung des Menschen sehr ungenau. Ihre Sensorplattform ist sehr sperrig und für mobile Anwendungen ungeeignet. Chen et al. [72] erreichen eine durchschnittliche Genauigkeit sowohl unter statischen als auch unter dynamischen Bedingungen von 0.10 bis 0.15 m. Allerdings liefert ihr Verfahren genauere Ergebnisse als ihr Referenzsystem. Zhong et al. [516] erreichen mit einem LKF vergleichbare Genauigkeiten in LoS-Fällen. Xu et al. [486] optimieren das Filter von Yudanto et al. [501], um Schätzungen in Echtzeit und mit einer Genauigkeit von 0.2 m zu liefern. Allerdings berücksichtigt das LKF nur den Korrektureffekt von Funk auf IMU, während die Korrektureffekte von IMU auf Funk ignoriert werden, sodass das Verfahren in NLoS Situationen ungenaue Positionen liefert. Zihajehzadeh et al. [522] schätzen genaue Positionen (< 0.21 m) von sehr statischen Bewegungen mit zwei kaskadierten LKF. Diese benötigen allerdings 20 s, um eine Position zu schätzen und können daher nicht in dynamischen Situationen genutzt werden. Fan et al. [116] entwickeln einen vielversprechenden Ansatz, der verschiedene Gangarten von Fußgängern mit unterschiedlichen Geschwindigkeiten (langsam und schneller) erfassen kann und adaptiv ein passendes KF

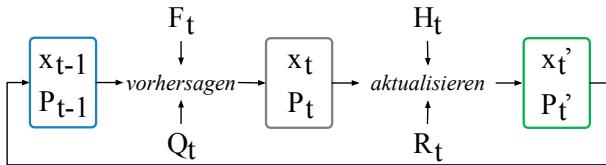


Abbildung 2.7: Vereinfachte Darstellung der KF-Verarbeitungskette über die Zeitschritte $t - 1$ bis t . Iterativer Optimierungsprozess: Das Filter sagt aus dem aktuellen Zustand x_{t-1} und dessen Unsicherheit P_{t-1} mit der Zustandsübergangsfunktion und deren Prozessunsicherheit Q_t den nächsten Zustand x_t mit der Unsicherheit P_t vorher. Das Filter misst und überprüft mit der Messmatrix H_t und deren Messunsicherheit R_t , ob die Vorhersage des Systemzustands zu den neuen Messwerten passt und korrigiert und aktualisiert dementsprechend, um den optimalen Zustand x'_t und dessen Unsicherheit P'_t zu liefern. [222] (Die Verarbeitungskette wird in Abschnitt 3.5.1 im Detail erläutert.)

verwendet (IMU Sensor am Fuß ermöglicht ZUPT). Bereits mit einem durchschnittlichen Fehler von 0.13 m, SD=0.15 m divergiert die Trajektorien stark von der Referenz, was auf einen hohen Orientierungsfehler zurückzuführen ist. Sie zeigen, dass ZUPT bei hohen Geschwindigkeiten nicht funktioniert: Selbst bei einer Geschwindigkeit von 4.65 m/s wird kein ZUPT-Ereignis erkannt. Ähnlich wie diese Dissertation zeigen sie, dass eine Fensterlänge der IMU-Signale von 1.25 s am besten einen Gangzyklus einer Person mit unterschiedlichen Geschwindigkeiten darstellen kann. Allerdings ist unklar, ob das Verfahren auch in NLoS Situationen oder mit dynamischen Bewegungen funktioniert.

Erweiterte Kalman-Filter (EKF)

Seit 1960 das KF vorgeschlagen wurde [222], wurden verschiedene verbesserte Algorithmen erforscht. Basierend auf dem KF Algorithmus wurde ein EKF entwickelt [265]. Sczyslo et al. [461], Fan et al. [115] und andere [321, 241, 114, 268, 456] verwenden das EKF, um Trägheitsdaten mit Funkdaten von Fußgängern zu fusionieren. Yao et al. [494] schätzen ungenaue (Fehler > 0.83 m) Positionen eines Radroboters mit Odometriedaten mit einem EKF. Basierend darauf lokalisieren Benini et al. [38] einen mobilen Agenten in einem Innenraum mit einer Fläche von 9.5 m \times 7.5 m mit einer Genauigkeit von 0.15 m. Foxlin et al. [142] nutzen EKF und ZUPT, um den Geschwindigkeitsfehler nach jedem Schritt linear zur Anzahl der Schritte zu korrigieren, um das kubisch-zeitliche Fehlerwachstum zu unterbrechen. Allerdings können sie die Orientierung

nicht korrigieren. Zwirello et al. [525, 505] untersuchen explizit die lose und eng gekoppelte Integration von Funk und IMU durch das EKF. Um ZUPT zu nutzen, muss allerdings die IMU an einem Fuß montiert werden und der Funksender muss am Kopf oder an der Schulter montiert werden, um die Blockierung des Funksignals durch den menschlichen Körper zu verringern. Li et al. [269, 268] berechnen den Mahalanobis-Abstand von der Beobachtung zur vorherigen Verteilung, um den Einfluss abnormaler Funksignale zu unterdrücken. So ist das EKF gegenüber intermittierendem Rauschen, kontinuierlichem Rauschen, Signalunterbrechung und anderen Abnormalitäten der Funkdaten robust. Bei Sichtverbindungen ist der Richtungsfehler schwer zu bestimmen und bei Nichtsichtverbindungen kann gar keine Position bestimmt werden. Mit laufenden und rennenden Testpersonen erreichen sie eine Genauigkeit von 0.45 m, jedoch nur in LoS-Situationen. Ebenso wie Li et al. [268] zeigen Pittet et al. [340], dass ein EKF in Situationen mit Nichtsichtverbindungen keine genauen Positionen erzielen kann und zu Filterdivergenz des nichtlinearen Systems führt. Außerdem kann das EKF den Orientierungsfehler nicht korrigieren.

Obwohl das EKF die Mahalanobis-Entfernung nutzt, um das Beobachtungsvertrauen von Funksystemen für die Unterdrückung von nicht-Gaußschem Rauschen, bspw. Fehler durch Mehrwege, zu korrigieren [270, 269, 268, 340], wird die Beobachtung nur auf der Grundlage der Information des vorherigen Zeitschritts bestimmt. Wenn im Systemstatus ein großer Fehler vorliegt, wird die Zuverlässigkeit der UWB-Beobachtung beeinträchtigt und die Positionsschätzung wird ungenau.

Spezifische Kalman-Filter

Zwirello et al. [525] zeigen, dass Bayessche Filter wie LKF und EKF nur die Beziehung zwischen dem aktuellen Zustand und dem vorherigen Zustand berücksichtigen und deshalb die Informationen aus allen vorherigen Zuständen und die entsprechenden Beobachtungen nicht ausreichend nutzen. Folglich nimmt die Schätzgenauigkeit bei kontinuierlicher größerer Beobachtung oder systematischem Fehler ab.

Um das EKF zu verbessern, korrigiert das rechtsinvariante EKF (engl. Right Invariant Extended Kalman-Filter, RIEKF) von Li et al. [266] das Beobachtungsvertrauen durch Integration der vorherigen Informationen in alle Beobachtungen in allen Filteriterationen. Während jeder Iteration wurde das Beobachtungsvertrauen des Funksystems basierend auf der Mahalanobis-Entfernung berechnet. Aufgrund der

2 Verwandte Arbeiten

wiederholten Konstruktion und Lösung der Verteilung wird angenommen, dass das Endergebnis eine Schätzung ist, die auf allen Beobachtungen und vorherigen Verteilungen basiert. So können in NLoS Situationen bei sehr langsamen Bewegungsgeschwindigkeiten, Positionsgenauigkeiten von 0.21 m , $SD=0.13\text{ m}$ erreicht werden. Im Vergleich zu EKF, das sich explizit auf die vorherigen Informationen stützt, kann RIEKF die Positioniergenauigkeit erheblich verbessern.

Li et al. [268] entwickeln ein komplementäres Kalman-Filter (engl. Complementary Kalman-Filter, KKF), um den Einfluss einer abnormalen Beobachtung oder eines abnormalen Systemzustands auf die Positionierungsergebnisse zu verringern. Die Genauigkeit der Positionsschätzung ist allerdings selbst in LoS Situationen sehr niedrig.

Shi et al. [399] entwickeln ein Fehlerzustands-Kalman-Filter (engl. Error-State Kalman-Filter, ESKF), um neue Funkempfänger selbst zu lokalisieren (Fehler $> 1.13\text{ m}$), indem eine Sensorplattform mit Funk- und IMU-Sensoren im Raum frei bewegt wird. Das ESKF wurde explizit für den Umgang mit vielen dynamischen, neuen Hypothesen entwickelt. Die Genauigkeit des ESKF ist allerdings sehr niedrig.

Julier et al. [218] entwickeln ein hochspezifisches UKF. Das UKF schätzt die Positionen in LoS Situation genauer und korrigiert Orientierungsfehler genauer als vergleichbare EKF (Fehler $< 0.64\text{ m}$). Allerdings ist die Parametrisierung des UKF deutlich aufwendiger und Expertenwissen und domänenspezifisches Wissen sind erforderlich.

Arasaratnam et al. [18] und Haykin et al. [19] entwickeln das Kubatur-Kalman-Filter (CKF). Das CKF berechnet das mehrdimensionale Gewichtsintegral mit einer sphärisch-radialen Regel und eignet sich gut für nichtlineare Systeme höherer Ordnung, bspw. wenige Mehrwegepfade. Wenn das Systemmodell jedoch ungenau ist, bspw. viele Mehrwegepfade oder sich der Systemzustand abrupt ändert und dynamisch ist, bspw. schnelle Richtungsänderungen, liefert das CKF ungenaue Positionen. Genauso wie alle anderen KF Varianten basiert CKF auf dem Minimum Mean Square Error Kriterium. Wenn das Signalmodell nicht Gaußsch ist, sind die Verfahren unwirksam und nicht mehr anwendbar.

2.3 Methoden zur Kopplung von Funk- und Trägheitssensoren

Um das Problem des nicht-Gaußschen Signalmodells zu lösen, schlagen Chen et al. [65, 67] und Liu et al. [67] einen Filteralgorithmus vor, der auf dem maximalen Korrentropiekriterium (engl. Maximum Correntropy Criterion, MCC) basiert und als maximales Korrentropie-Kalman-Filter (engl. Maximum Correntropy Kalman-Filter, MCKF) bezeichnet wird [66]. Das Modell liefert genaue Positionen für Gauß-verteilte Signale (Fehler 0.42 m) und liefert selbst für nicht Gauß-verteilte Signale genaue Positionen (Fehler $>0.58\text{ m}$). Das MCKF kann grobe Fehler (bspw. Mehrwege) im Signal effektiv filtern. MCKF liefert unter Laborbedingungen genaue Positionen, aber sehr ungenaue in starken Mehrwegesituationen oder bei dynamischer Bewegung.

Fazit: Im Allgemeinen stützt sich jede Variante des KFs kritisch auf ein genaues Systemmodell und der statistischen Verteilung des Messrauschens und hat große Schwierigkeiten, die durch Umwelteinflüsse verursachten Ausreißer, wie z. B. NLoS in Funksystemen, zu berücksichtigen [86, 283, 113]. KF benötigen Experten- und Domänenwissen, um ein genaues Systemmodell und genaue statistische Eigenschaften des Messrauschens zu definieren, um ein optimales Filter zu entwickeln. Darüber hinaus sind KF nicht immun gegen Messdatenfehler und plötzliche Sensorausfälle, die die Filtergenauigkeit erheblich verringern und Filterdivergenzen verursachen [115]. Ein weiteres Problem sind die unterschiedlichen und variierenden Aktualisierungsraten von Trägheitssensoren (100 Hz bis 1000 Hz [156]) und Funksensoren (10 Hz bis 100 Hz [493]). Die a priori erstellten dynamischen stochastischen Fehlermodelle können keine unbekannten Schwankungen der Aktualisierungsraten abbilden und führen zu ungenauen Positionen. Ähnlich wie das LKF approximiert das EKF die Beobachtungsgleichung im wesentlichen linear und kann daher die Fehler des stark nichtlinearen Beobachtungsmodells eines Funksystems nicht approximieren [150] und liefert genau wie das LKF ungenaue Positionen in NLoS-Situationen oder in Situationen mit hochdynamischen Bewegungen. Bspw. bestimmt ein EKF die Beobachtung nur auf der Grundlage der Informationen aus dem vorherigen Zeitschritt. Unter solchen Umständen wird die Zuverlässigkeit der UWB-Beobachtung beeinträchtigt, wenn ein großer Fehler aufgrund von Mehrwege im Systemstatus vorliegt. Das Hauptproblem aller KF Varianten, einschließlich EKF, UKF RIEKF, ESKF, und M/CKF ist, dass alle auf dem minimalen mittleren quadratischen Fehlerkriterium basieren, d. h., wenn das Signalmodell nicht Gaußsch ist, sind

2 Verwandte Arbeiten

die Methoden unwirksam, divergieren und sind nicht mehr praktikabel.

In der Tabelle 2.4 werden herausragende Arbeiten aufgelistet und die verwendeten Sensoren, Methoden, ihre Genauigkeit und Nachteile zusammengefasst. Die Übersicht über die Genauigkeiten der verschiedenen Kalman-Filtervarianten in Tabelle 2.4 zeigt, dass die Variante und Komplexität des Algorithmus keinen oder nur geringen Einfluss auf die Genauigkeit der Positionsschätzung hat. Der Auswahl des entsprechenden Filters zur Verarbeitung der Eingabedaten wird implizit vorgegeben. Viel wichtiger ist jedoch die optimale Konfiguration und Parameterisierung des Filters für die jeweilige Anwendung. In den Kapiteln 4, 5 und 7 werden Varianten des EKF, basierend auf Kok et al. [241, 242], implementiert, die für die jeweiligen Eingabedaten optimiert werden, um optimale Mess-, Beobachtungs- und Übergangs-Fehlermodelle abzuleiten, um optimale Schätzungen zu ermöglichen.

2.3.4 Partikel-Filter

Die Herausforderungen von PF in realen Anwendungen sind die ordnungsgemäße Modellierung der Systemfunktionen und der Rauschverteilung, die Echtzeitleistung, da der Algorithmus sehr rechenintensiv ist, und das Problem der verlorenen Trajektorie, wenn nur wenige Funkempfänger vorhanden sind und die Position verloren geht. Das Hauptproblem, die verlorenen Trajektorie, führt dazu, dass alle Partikel im Vergleich zum tatsächlichen Systemzustand fehlerhaft sind und alle Gewichte vernachlässigbar klein sind [438]. Das PF verliert die Position und Zuordnung vollständig und muss neu initialisiert werden.

Mit der kontinuierlichen Erhöhung der Rechenleistung untersuchen Wang et al. [461, 462] das PF, um Funk und IMU Daten zu fusionieren. Die Genauigkeit der Positionsschätzung ist allerdings deutlich ungenauer als vergleichbare KF (Fehler > 0.68 m). Um die Kosteneffizienz bei der Skalierung des Lokalisierungssystem zu verbessern, fusionieren Tian et al. [429] INS- und UWB-Abstandsmessungen mit einem PF. Das PF unterstützt eine beliebige Anzahl von UWB-Ankerknoten, benötigt allerdings einen erheblichen Optimierungsaufwand und liefert dennoch ungenaue Positionen (Fehler > 0.49 m).

Ebenso positionieren Arulampalam et al. [20] und Wang et al. [462] Fußgänger mit einer hohen Ungenauigkeit von über 0.7 m (SD=0.66 m, selbst bei einer langsamen Bewegungsgeschwindigkeit von 1.5 bis

2.3 Methoden zur Kopplung von Funk- und Trägheitssensoren

Tabelle 2.4: Ergebnisse verwandter Arbeiten der letzten fünf Jahre, die Trägheitsmessungen und Funksignale mit KF Methoden fusionieren. Übersicht über verwendete Sensoren, Methoden, höchstmögliche Genauigkeit oder geringste Fehler **fettgedruckt** und Nachteile.

Autoren	Sensortyp	Methode	Fehler	Nachteile in [m] *
Corrales et al. [85]	Kamera	LKF	-	Ungenaue Posenschätzung und unpraktischer Systemaufbau.
Chen et al. [72]	UWB IMU	LKF	0.10 bis 0.15	Widersprüchliche Ergebnisse: das Verfahren liefert genauere Positionen als das verwendete Referenzsystem. **
Zhong et al. [516]	UWB IMU	LKF	-	
Xu et al. [486]	UWB IMU	LKF	0.20	Methode korrigiert IMU mit Funk, aber nicht Funk mit IMU. **
Zihajehzadeh et al. [522]	UWB IMU	LKF (Kaskade)	<0.21	Benötigt 20 s zur Berechnung einer aktuellen Position von statischen Bewegungen.
Fan et al. [116]	UWB IMU	LKF ZUPT	0.13	IMU muss starr am Fuß platziert werden. Verfahren divergiert stark bei dynamischen Richtungswechseln. **
Yao et al. [494]	UWB IMU	EKF Odometrie	0.83	Sensoren müssen starr am Radroboter angebracht werden. **
Benini et al. [38]	UWB IMU	EKF Odometrie	0.15	Benötigt Odometriemessungen. **
Foxlin et al. [142]	UWB IMU	EKF ZUPT	>0.24	Hohe Orientierungsfehler erlauben nur statische Bewegungen. **
Zwirrello et al. [525, 505]	UWB IMU	EKF ZUPT	>1.5	Die enge und die lose Kopplung positionieren sehr ungenau, IMU muss am Fuß und Funk am Kopf oder Schultern montiert werden.
Li et al. [269, 268]	UWB IMU	EKF NLoS	0.45	Robust gegen unterschiedliches Sensorrauschen in LoS Situationen. **
Pittet et al. [340]	UWB IMU	EKF NLoS	>0.98	Irreparabler Orientierungsfehler. Hohe Ungenauigkeit in NLoS Situationen. **
Li et al. [266]	UWB IMU	RIEKF NLoS	0.21	Liefert genaue Positionen in NLoS Situationen, bei langsamem, eher statischen Bewegungen.
Shi et al. [399]	UWB IMU	ESKF	1.13	Explizit für den Umgang mit unbekannten Hypothesen entwickelt. Nicht direkt vergleichbar mit anderen Verfahren. **
Julier et al. [218]	UWB IMU	UKF	<0.64	Hohe Genauigkeit in LoS Situationen. Benötigt Experten- und Domänenwissen. **
Li et al. [268]	UWB IMU	KKF	>0.46	Schnelles KF, aber niedrige Genauigkeit selbst in LoS Situationen. **
Arasaratnam et al. [18]	UWB IMU	CKF NLoS	0.28	Ermöglicht hochdimensionale Hypothesen; Ungenau, wenn die Gauss-Annahme, bspw.
Haykin et al. [19]	UWB IMU	CKF NLoS	0.34	Mehrwege, von der Messung abweicht.
Chen et al. [65, 67]	UWB IMU	MCKF NLoS	>0.58	Liefert im Labor für (Nicht)Gauss-Annahmen die genauesten Positionen. In realistischen
Liu et al. [67]	UWB IMU	MCKF NLoS	0.42	Mehrwegesituationen und dynamischer Bewegung divergiert das Filter.

*) Die Ergebnisse, die zur Bewertung der verwendeten Methoden verwendet wurden, werden bestmöglich aus der jeweiligen Darstellungsform extrahiert.

**) Die Ergebnisse wurden unter Laborbedingungen unter Funkausbreitungsbedingungen mit klarer Sichtlinie (LoS) erhalten.

2 Verwandte Arbeiten

2.5 m/s) mit UWB (am Kopf) und IMU (am Fuß) und einem markerbasierten Referenzsystem. Liu et al. [279] entwickeln ein PF, um eine IMU- und UWB-basierte Positionierung der menschlichen Hand ohne ZUPT und ZAUR zu ermöglichen. Ihre sehr rechenintensive Methode erreicht einen durchschnittlichen Fehler von 2.2 m, obwohl ihr markergestütztes Referenzkamerasytem eine hohe Varianz aufweist. Elbes et al. [108] entwickeln ein PF das mit simulierten Daten eines IMU- (am Fuß platziert) und eines UWB- (am Kopf platziert, um permanent Sichtverbindung zu gewährleisten) Sensors eine durchschnittliche Positionsgenauigkeit von 4.5 m (4 Anker) und 0.3 m (50 Anker) erreicht.

Tian et al. [430] entwickeln ein PF, das akkumulierte Fehler zurücksetzt. Allerdings erzielt ihr Verfahren nur eine Positionsgenauigkeit von 2 m bis 4 m (SD=1.5 m). Nummiaro et al. [317] zeigen, dass PF in einer nicht-Gaußschen Rauschumgebung eine hohe Genauigkeit erzielen kann, wenn hoher Rechenaufwand toleriert werden kann. Allerdings kann die Genauigkeit durch starke Interferenzen stark beeinträchtigt werden (Fehler von 0.76 m bis 3.02 m). Um die Positionsgenauigkeit weiter zu verbessern, fusionieren Gonza'lez et al. [150] Funk, IMU und Odometer und erzielen selbst unter NLoS-Bedingungen eine hohe Positionsgenauigkeit (Fehler < 0.21 m) eines Roboters. Jedoch leidet das PF unter einer sehr hohen Rechenkomplexität.

Fazit: In der Tabelle 2.5 werden herausragende Arbeiten aufgelistet und die verwendeten Sensoren, Methoden, ihre Genauigkeit und Nachteile zusammengefasst. Die Übersicht über die Fehler der PF in Tabelle 2.5 zeigt, dass PF keine genauere Position als das KF schätzt. Die Tatsache, dass PF für die Positionsschätzung in der Literatur wesentlich seltener verwendet wird als KF, bestätigt diese Erkenntnis. Der hohe Rechenaufwand und der explodierende Positionsfehler, wenn die Partikel die Trajektorie verlieren, führen dazu, dass PF im Rahmen der Dissertation nicht genauer untersucht wird und die Methoden in den Kapiteln 4 bis 7 nicht mit PF sondern mit KF bzw. EKF verglichen werden.

2.3.5 Kalman- und Partikel-Filter im Vergleich

Canedo-Rodríguez et al. [60] vergleichen die Genauigkeit und Leistung von KF, EKF und PF. Prinzipiell nutzen KF, EKF und PF Fusionsalgorithmen mehrere unabhängige Quellen von Standortinformationen [216, 431]. Im Vergleich zu EKF erzielt PF keine genaueren Schätzungen bei hochdimensionalen und rechenaufwendigen Proble-

2.3 Methoden zur Kopplung von Funk- und Trägheitssensoren

Tabelle 2.5: Ergebnisse renommierter verwandter Arbeiten der letzten fünf Jahre, die Trägheitsmessungen und Funksignale mit Partikel-Filter Methoden fusionieren. Übersicht über verwendete Sensoren, Methoden, höchstmögliche Genauigkeit oder geringste Fehler **fettgedruckt** und Nachteile.

Autoren	Sensortyp	Methode	Fehler	Nachteile in [m] *
Wang et al. [461, 462]	UWB IMU	PF	0.68	Deutlich ungenauer als vergleichbare KF. **
Tian et al. [429]	UWB IMU	PF	0.49	Hoher Optimierungsaufwand; Niedrige Positionsgenauigkeit; Hohe Rechenkosten. **
Arulampalam et al. [20]	UWB IMU	PF	0.70	IMU muss am Fuß platziert werden. Niedrige Genauigkeit, selbst bei statischen Bewegungen.
Liu et al. [279]	UWB IMU	PF	>2.20	Hohe Positionsfehler bei dynamischen Bewegungen. **
Elbes et al. [108]	UWB IMU	PF	0.30 bis 4.50	Trotz hoher Systemredundanz $Rx=50$ liefert PF im Vergleich zu KF sehr ungenaue Positionen.
González et al. [150]	UWB IMU	PF Odometrie	0.21	Sehr hohe Rechenkomplexität. Erfordert Odometriesensorik. Nicht vergleichbar. **
Tian et al. [430]	UWB IMU	PF	2.0 bis 4.0	Hohe Ungenauigkeit.
Nummiaro et al. [317]	UWB IMU	PF	0.76 bis 3.02	Hohe Genauigkeit bei nicht-Gaußschem Raum; Enorm hoher Rechenaufwand.
Rodríguez et al. [60]	UWB IMU	PF	0.58	PF kaum genauer als EKF selbst bei langsam Bewegungen; PF deutlich rechenaufwändiger als EKF.
Yu et al. [500]	UWB IMU	PF	0.98 bis 2.04	Sensoren starr an den Füßen oder Taille; Optimierte PF positioniert sehr ungenau.

*) Die Ergebnisse, die zur Bewertung der verwendeten Methoden verwendet wurden, werden bestmöglich aus der jeweiligen Darstellungsform extrahiert.

**) Die Ergebnisse wurden unter Laborbedingungen unter Funkausbreitungsbedingungen mit klarer Sichtlinie (LoS) erhalten.

2 Verwandte Arbeiten

men, wie z. B. Funkausbreitung mit NLoS. Bei niedrigdimensionalen Problemen, wie z. B. Funkausbreitung mit LoS und langsamen Bewegungen, kann PF im Vergleich zu EKF die gleiche Schätzgenauigkeit erzielen (durchschnittlicher Fehler von 0.58 m) [94, 181, 431]. Im Vergleich zu PF rechnet EKF effizienter, ist jedoch für Störungen durch nicht-Gaußsches Rauschen anfälliger. Die Erkenntnisse decken sich mit Fehlern für KF und PF der Tabellen 2.4 und 2.5.

Yu et al. [500] führen einen umfangreichen Vergleich von KF und PF Varianten durch. Sie implementieren die prominentesten modellgetriebenen Methoden Madgwick und Mahony zur Orientierungsschätzung und LKF, EKF und PF zur Positions- und Orientierungsschätzung sowohl in PDR- als auch in SINS-Verfahren. Sie vergleichen die Positionsgenauigkeit von LKF, EKF und PF, wenn Sensoren an Füßen, Taille und Hand angebracht sind. Sie finden heraus, dass jedes dieser Modelle für jede Platzierung eine individuelle Leistung liefert, wobei PDR die SINS Verfahren im Allgemeinen leicht übertrifft. Insgesamt ist die Positionierungsgenauigkeit des Taillensystems am höchsten (höchste Genauigkeit durchschnittlich 0.98 m für Kurvenbewegungen mit PDR-Filter) und stabiler als die des Fußsystems (höchste Genauigkeit durchschnittlich 1.37 m für Kurvenbewegungen mit Mahony-Filter). Das LKF-Filter mit Daten eines Mobiltelefons, das in der Hand gehalten wurde, schnitt am schlechtesten ab (höchste Genauigkeit durchschnittlich 2.04 m für Kurvenbewegungen). Yu et al. [500] finden heraus, dass Bewegungen ohne scharfes Drehen, z. B. sanfte Kurven statt Rechtecke zu genaueren Trajektorien führen. Allerdings weist ihr UWB-Referenzsystem jedoch eine erhebliche Standardabweichung von über SD=0.20 m auf, weshalb die Ergebnisse möglicherweise noch etwas schlechter sind. Darüber hinaus erfordern die Methoden, die am besten funktionieren, eine aufwendige handgefertigte Parameteroptimierung und ermöglichen deshalb nur eine explizite Anwendung, einen bestimmten Benutzer und eine bestimmte Bewegung, ohne abrupte und scharfe Änderungen der Orientierung. Yu et al. [500] geben an, dass alle modellgetriebenen Verfahren langfristig enorm drifteten und schlagen vor, Fehler durch die Verwendung externer Sensoren wie Wi-Fi und UWB zu kompensieren.

2.3.6 Datengetriebene Methoden

In den Bereichen bildbasierte und trägheitsbasierte Lokalisierung gibt es einige datengetriebene Ansätze zur Schätzung der Pose der

Kamera oder Fußgängern und Fahrzeugen. Die Ansätze der Bildverarbeitung können allerdings nicht auf die Fragen dieser Dissertation übertragen werden und werden daher nicht weiter diskutiert. Die Ansätze zur trägeitsbasierten Lokalisierung werden in Kapitel 5 detailliert erörtert. Gegenwärtig, 2021, gibt es keine öffentlich bekannte Methode, die vergleichbare datengetriebene Methoden verwendet, um vergleichbare multimodale Signale von Funk- und Trägheitssensoren zu fusionieren. In den Kapiteln 4 bis 7 werden explizit eng verwandte Arbeiten diskutiert. In diesem Abschnitt werden die wenigen engsten verwandten Arbeiten diskutiert. Zunächst werden Verfahren des ML und dann Verfahren des DL diskutiert.

Sensorfusion mit ML.

Ferreira et al. [137] entwickeln eine lose gekoppelte Fusion von IMU- und UWB-Sensorik, unterstützt durch einen Entscheidungsalgorithmus. Ein Random Forest Algorithmus erkennt zunächst NLoS Situationen in UWB Messungen und stützt die nachgelagerte Lokalisierung von Einsatzkräften (durchschnittlicher Fehler von 1.50 m).

Im Gegensatz dazu schlägt diese Dissertation eine datengetriebene Positionsschätzung vor, die die NLoS Eigenschaften implizit aus den Ankunftszeitstempeln extrahiert, um eine genaue Position zu schätzen. Der Ansatz von Ferreira et al. ist somit nicht mit den Verfahren dieser Dissertation vergleichbar.

Xu et al. [488] entwickeln ein Entscheidungsbaum-basiertes Verfahren, das ein FIR Filter nutzt, um eng gekoppelte IMU (am Fuß platziert) und UWB (am Kopf platziert) Signale zu kombinieren. Das UWB Signal kompensiert den Fehler der IMU. Auf kreisförmigen Trajektorien erreichen sie eine Genauigkeit von 0.53 m. Leider untersuchen sie ihr Verfahren nicht in NLoS Situationen.

Im Gegensatz zu dem Verfahren von Xu et al. schlägt diese Dissertation Verfahren vor, die nicht auf die starre und definierte Platzierung der Sensoren angewiesen ist und schätzt mit einer losen modularen Verarbeitungskette deutlich genauere Positionen.

Sensorfusion mit DL.

Li et al. [273] nutzen ein Fuzzy neuronales Netz, um GPS-, UWB-, IMU- und Odometrie-Sensorik zu kombinieren. Ihr System liefert auf GPS- und IMU-Signalen durchschnittliche Fehler von 0.65 m bis 1.83 m und auf GPS- und IMU- und UWB-Signalen durchschnittliche Fehler

2 Verwandte Arbeiten

von 0.19 m bis 0.56 m. Allerdings verschlechtert sich die Genauigkeit drastisch außerhalb der Laborbedingungen. Die sehr ungenauen Ergebnisse sind aufgrund der verwendeten Sensorsignale nicht direkt mit den Methoden dieser Dissertation vergleichbar.

Im Gegensatz dazu werden in dieser Dissertation Methoden vorgeschlagen, die direkt aus den Daten lernen und keinen Fuzzy-typischen Einschränkungen und Regeln unterliegen. Da es KF-Methoden gibt, die viel genauer positionieren als die Methode von Li et al. wird diese Methode auch nicht als Referenz verwendet.

Zyner et al. [527] präsentieren eine Methode zur Vorhersage der zukünftigen Trajektorie eines Fahrzeugs, die ein LSTM verwendet. Die Trajektorie des Fahrzeugs wird anhand der Position, der Orientierung und der Geschwindigkeit aus den vom Fahrzeug gesammelten GPS-, Trägheits- und Kilometerzähler-Daten gelernt. In diesem Zusammenhang haben Zyner et al. [527] mit ihrem Ansatz eine Vorhersagegenauigkeit der Trajektorie von durchschnittlich 90.66 % erreicht.

Im Gegensatz zu dem in der Dissertation vorgeschlagenen Regressionsansatz, klassifizieren Zyner et al. mit zusätzlicher Odometriesensorik eine Trajektorienform und schätzen keine Pose. Das Konzept, die Verlustfunktionen und die gesamte Verarbeitungskette unterscheiden sich von dieser Dissertation und sind nicht vergleichbar. Die Arbeit von Zyner et al. motiviert die Forschungsansätze dieser Dissertation.

Yao et al. [495] reduzieren mit ihrem umfassenden Lernansatz das Sensorrauschen durch verschiedene Signalerfassungsmodalitäten wie GPS und IMU. Sie verwenden CNN und RNN Verfahren, um lokale Interaktionen zwischen ähnlichen mobilen Sensoren, bspw. Beschleunigung und Gyroskop zu nutzen, lokale Interaktionen verschiedener Sensorsmodalitäten zu globalen Interaktionen mit Hilfe von GPS zusammenzuführen und zeitliche Beziehungen zu extrahieren, um Signalstörungen zu modellieren. Sie haben die Wirksamkeit ihrer Methode anhand von drei repräsentativen und anspruchsvollen Aufgaben demonstriert: Fahrzeugverfolgung mit Bewegungssensoren, Erkennung heterogener menschlicher Aktivitäten und Benutzeridentifikation mit biometrischer Bewegungsanalyse. Sie erzielen einen mittleren absoluten Fehler von 40.43 m und eine Varianz von 5.24 m, während bei der klassischen Sensordatenfusion ein mittlerer absoluter Fehler von 606.59 m und eine Varianz von 56.57 m erzielt wurden.

2.3 Methoden zur Kopplung von Funk- und Trägheitssensoren

Tabelle 2.6: Ergebnisse renommierter verwandter Arbeiten der letzten fünf Jahre, die Trägheitsmessungen und Funksignale mit datengetriebenen Methoden fusionieren. Übersicht über verwendete Sensoren, Methoden, höchstmögliche Genauigkeit oder geringste Fehler **fettgedruckt** und Nachteile.

Autoren	Sensortyp	Methode	Fehler in [m] *	Nachteile
Li et al. [273]	GPS UWB IMU Odometrie	Fuzzy NN	0.19 bis 0.56	Die Genauigkeit verschlechterte sich drastisch außerhalb der Laborbedingungen. Die Ergebnisse sind nicht vergleichbar. **
Ferreira et al. [137]	UWB IMU NLoS	KF ML	1.50	Sehr ungenaues Verfahren, das eine komplexe Konfiguration durch Experten benötigt. NLoS Situationen müssen explizit aufwändig etikettiert werden. Die Ergebnisse sind nicht vergleichbar. **
Xu et al. [488]	UWB IMU	FIR ML	0.53	IMU muss starr am Fuß angebracht werden. Hohe Genauigkeit nur bei kreisförmigen Trajektorien. Die Ergebnisse sind nicht vergleichbar. **
Zyner et al. [527]	GPS IMU Odometrie	ML	-	- Benötigt Odometriedaten; Klassifiziert Bewegungsstrajektorien ungenau; Die Ergebnisse sind nicht vergleichbar. **
Yao et al. [495]	GPS IMU	CNN RNN	40.43	Obwohl der Ansatz Karteninformation zur Kalibrierung nutzt, liefert er unpraktikabel ungenaue Positionen ($SD=5.24$ m). Die Ergebnisse sind nicht vergleichbar. **

*) Die Ergebnisse, die zur Bewertung der verwendeten Methoden verwendet wurden, werden bestmöglich aus der jeweiligen Darstellungsform extrahiert.

**) Die Ergebnisse wurden unter Laborbedingungen unter Funkausbreitungsbedingungen mit klarer Sichtlinie (LoS) erhalten.

Im Gegensatz zu dem in der Dissertation vorgeschlagenen Ansatz haben Yao et al. [495] einen Ende-zu-Ende-Ansatz gewählt, der im Rahmen der Dissertation ähnlich auf verschiedene Signalkompositionen angewendet wurde. In Kapitel 7 werden Kombinationen von CNN und RNN ausgewertet und es wird gezeigt, dass diese Kombinationen nicht die genauesten Posen liefert. Darüber hinaus unterscheiden sich das Konzept, die Verlustfunktionen und die gesamte Verarbeitungskette (von den Eingangssignalen über die Vorverarbeitung bis zur Verlustfunktion und Rekonstruktion der Trajektorie) erheblich von den Methoden dieser Dissertation und sind nicht vergleichbar. Die Arbeit und die hohen Ungenauigkeiten in den Ergebnissen von Yao et al. motivieren die Forschungsansätze dieser Dissertation.

Fazit: Die wenigen verwandten Arbeiten, die die Fusion von Funk- und Trägheitsmessungen mit datengetriebenen Methoden untersu-

2 Verwandte Arbeiten

chen, motivieren die Notwendigkeit der Methoden dieser Dissertation. Keine vorhandene datengetriebene Methode ist direkt mit den in dieser Dissertation vorgeschlagenen Verfahren vergleichbar. Daher werden eng verwandte und vergleichbare Verfahren in jedem Kapitel 4 bis 7 separat diskutiert und ausgewertet. In der Tabelle 2.6 werden herausragende Arbeiten aufgelistet und die verwendeten Sensoren, Methoden, ihre Genauigkeit und Nachteile zusammengefasst. Der Überblick über die Genauigkeit der datengetriebenen Methoden in Tabelle 2.6 zeigt, dass die datengetriebenen Methoden des Stands der Technik wesentlich ungenauere Positionen als das herkömmliche KF liefern. Die Tatsache, dass datengetriebene Methoden für die Positionsschätzung mit Funk- und Trägheitsmessungen in der Literatur kaum oder überhaupt nicht verwendet werden, motiviert diese Dissertation.

Schlussfolgerung

Die Ergebnisse der gesamten Literaturrecherche motivieren die Notwendigkeit der neuen Methoden dieser Dissertation, die die Messungen unterschiedlicher Sensoren entrauschen, mit nichtlinearen Ereignissen umgehen, dynamische Bewegungen zuverlässig und korrekt schätzen, unabhängig von der Art und Weise der Platzierung am Objekt sind, Sensoren nicht starr befestigt werden müssen, einfach konfigurierbar sind, echtzeitnahe aktualisieren, möglichst einfach erweiterbar sind und hochgenaue und präzise Trajektorien rekonstruieren können.

3 Grundlagen

Das Kapitel stellt die Grundlagen vor. Zunächst werden Grundlagen zur Lokalisierung eingeführt (Abschnitt 3.1 bis 3.3). Dann wird der allgemeine Datenvorverarbeitungsprozess beschrieben, der je nach Methode und Anwendung individuell angepasst wird (Abschnitt 3.4). Abschließend werden die Grundlagen der verwendeten Methoden vorgestellt, nämlich konventionelle modellgetriebene Methoden (Abschnitt 3.5), allgemeine datengetriebene Lernmethoden (Abschnitt 3.6) und spezifische sequenzbasierte Lernmethoden (Abschnitt 3.7).

3.1 Kamerabasierte Lokalisierung

Kamerabasierte Verfahren erlauben die Detektion von Objektpositionen und deren räumliche Ausrichtung (Orientierung) [249]. Hierzu werden im Messfeld mehrere Kameras in äquidistanten Abständen fest installiert, die mit einer hohen Frequenz Lichtsignale an die Umgebung abgeben [249]. Wie Abbildung 3.1 veranschaulicht werden am Objekt, dessen Pose bestimmt werden soll, kugelförmige Marker angebracht, die mit einem retro-reflektierenden Material beschichtet sind, um das in der Nähe des Kameraobjektivs erzeugte Licht zu reflektieren [249].

Anhand von sogenannten Time of Flight (dt. Signallaufzeit, ToF)-Messungen wird unter Einbezug der Ausbreitungsgeschwindigkeit des Lichts ($299.792.458 \text{ m/s}$) die Dauer zwischen der Erzeugung und Aufnahme des Lichtsignals an der Kamera gemessen, wodurch die Objektpose im Zeitverlauf bestimmt wird [249]. Kamerabasierte Verfahren weisen annähernd echtzeitfähige Eigenschaften auf und stellen Submillimeter und Sekundengrad genaue Posen bereit [68, 128].

Jedoch sind solche Verfahren anfällig für wechselhafte Umgebungsbedingungen, wie etwa extreme Helligkeit (Linsenreflexionen) oder Dunkelheit (Sensorrauschen) und ebenso resultieren ungünstige Witterungsbedingungen wie Hagel oder Schnee in einer zeitweise oder gänzlich nicht zu erfassenden Objektpose, aufgrund von Objektiv-Verschmutzungen [249]. Aufgrund zuerst genannter Problematik müssen kamerabasierte Systeme regelmäßig und insbesondere bei sich verändernden Umgebungsbedingungen erneut kalibriert werden [249].

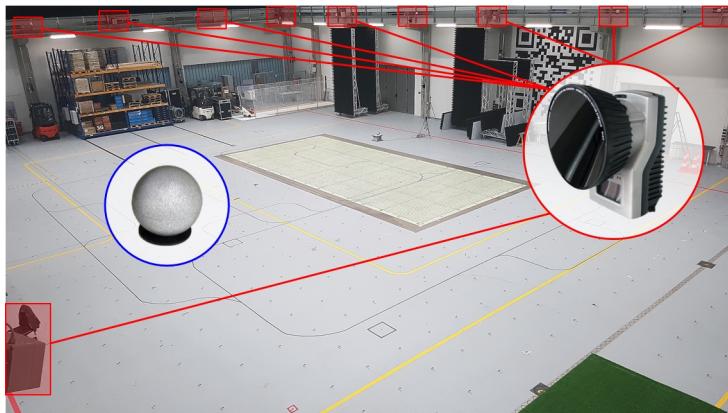


Abbildung 3.1: Seitenansicht eines kamerabasierten Lokalisierungssystems, das in dieser Dissertation als Referenzsystem verwendet wird. Rote Kästchen zeigen 10 von 25 Befestigungspunkten der 12-Megapixel-Kameras der Qualysis 7+ -Serie. (Roter Kreis: eine Kamera; blauer Kreis: ein reflektierender Marker).

Ebenso erfordern stark reflektierende Materialien wie glänzende Oberflächen in der Messumgebung die Software-seitige Maskierung dieser Areale und schränken die Objekterfassung zusätzlich ein [249]. Insbesondere zeitweilige Sichtverdeckungen (NLoS) zwischen Kamera und Objekt führen zu Signalblockaden [289, 249]. Ferner können falsch angebrachte Marker, Körperteile oder Kleidung zur Verdeckung führen und ebenso Lichtsignale blockieren [289].

Aufgrund der einzigartig hohen Genauigkeit und Präzision kamerabasierter Methoden werden in dieser Dissertation kamerabasierte Systeme als Referenzsysteme verwendet.

3.2 Trägheitsbasierte Lokalisierung

In diesem Abschnitt werden Grundlagen der trägheitsbasierten Lokalisierung eingeführt. Abschnitt 3.2.1 führt die wesentlichen Bewegungsgleichungen ein. Abschnitt 3.2.2 erörtert das (fahrzeugfeste)Trägheitsnavigationssystem. Und Abschnitt 3.2.3 beschreibt die (Fußgänger)Koppelnavigation.

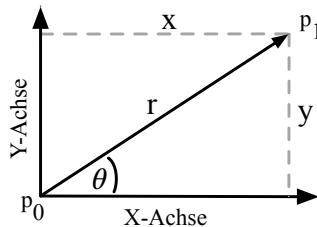


Abbildung 3.2: 2D-Transformation der Position p aus dem Kartesischen- in das Polarkoordinatensystem.

3.2.1 Bewegungsgleichungen

Mithilfe der klassischen Mechanik [238] lassen sich unbekannte kinematische Größen, wie z. B. Position, Distanz, Geschwindigkeit, Beschleunigung und Orientierung, ermitteln, die von den verwendeten Sensoren nicht direkt bereitgestellt werden können. Zu diesem Zweck wird jedem Sensor, insbesondere Sensoren, die nicht starr am Körper befestigt sind, ein eigenes Koordinatensystem zugewiesen [437]. Jedes dieser relativen, lokalen Koordinatensysteme kann in das absolute, globale Koordinatensystem der Erde transformiert werden. Messfehler bei der sensorischen Erfassung oder Fehler bei der Umwandlung zwischen den Koordinatensystemen, verzerren die Variablen [243].

Werden die objektbezogenen Bezugssysteme jeweils in einem 3D-Koordinatensystem unter Einbezug der Zeit t betrachtet, so lässt sich eine Folge von Positionen ($p = p_0, p_1, \dots, p_n$), Distanzen ($d = d_0, d_1, \dots, d_n$), Geschwindigkeiten ($v = v_0, v_1, \dots, v_n$), Beschleunigungen ($acc = acc_0, acc_1, \dots, acc_n$) und Orientierungen ($\theta = \theta_0, \theta_1, \dots, \theta_n$) mathematisch als Vektoren der Länge n beschreiben, wobei $n = t$ gilt. Jedes dieser Elemente repräsentiert einen 3D Vektor mit x -, y - und z -Koordinaten im euklidischen Raum. Ist die Position p eines Objektes gegeben, so lässt sich p zum Zeitpunkt t wie folgt ausdrücken:

$$p = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \text{ mit } x, y, z \in \mathbb{R}_0. \quad (3.1)$$

3 Grundlagen

d lässt sich aus konsekutiven p differenzieren:

$$d = \sqrt{(\partial p)^2} = \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix}, \text{ mit } d_x, d_y, d_z \in \mathbb{R}_0^+ \quad (3.2)$$

oder aus konsekutiven v integrieren:

$$d(t) = d_0 + \int_0^t v(t) dt = \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix}, \text{ mit } d_x, d_y, d_z \in \mathbb{R}_0. \quad (3.3)$$

d bzw. der Radius $r=\sqrt{x^2+y^2}$ lässt sich auch aus zwei konsekutiven Positionen p_0 und p_1 gemäß Formel 3.2 bestimmen, siehe Abbildung 3.2.

v lässt sich aus konsekutiven p differenzieren:

$$v = \frac{\partial p}{\partial t} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}, \text{ mit } v_x, v_y, v_z \in \mathbb{R}_0 \quad (3.4)$$

oder aus konsekutiven acc integrieren:

$$v(t) = v_0 + \int_0^t acc(t) dt = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}, \text{ mit } v_x, v_y, v_z \in \mathbb{R}_0. \quad (3.5)$$

acc lässt sich aus konsekutiven v differenzieren:

$$acc = \frac{\partial v}{\partial t} = \begin{pmatrix} acc_x \\ acc_y \\ acc_z \end{pmatrix}, \text{ mit } acc_x, acc_y, acc_z \in \mathbb{R}_0. \quad (3.6)$$

θ (um die Vertikalachse, z) lässt sich aus konsekutiven Winkelgeschwindigkeiten ω integrieren:

$$\theta(t) = \theta_0 + \int_0^t \omega(t) dt = (\theta_z), \text{ mit } \theta_z \in \mathbb{R}_0. \quad (3.7)$$

θ lässt sich auch über den Winkel $\theta=\arctan 2(y, x)$ zwischen zwei Positionen p_0 und p_1 bestimmen, siehe Abbildung 3.2.

$$(3.8)$$

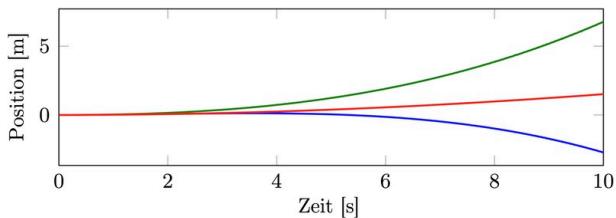


Abbildung 3.3: Mathematisch integrierte Position für Rotationen um die x- (blau), y- (grün) und z-Achse (rot) (adaptiert nach Kok et al. [241, 242]).

3.2.2 (Fahrzeugfestes) Trägheitsnavigationssystem

Die Trägheitsnavigation ist eine in sich geschlossene Navigationstechnik, bei der typischerweise Messungen von Beschleunigungssensoren, Gyroskopen und Magnetometern verwendet werden, um die Position und räumliche Ausrichtung einer Vorrichtung relativ zum Objekt, an dem es angebracht ist, mit Bezug zu einer bekannten Initialposition, -orientierung und -geschwindigkeit zu verfolgen [475, 68]. Inertiale Trägheitsmesseinheiten (IMU) enthalten typischerweise je Sensorsortyp (Beschleunigungssensor, Gyroskop und Magnetometer) drei orthogonal ausgerichtete Sensoren, welche die lineare, relative Beschleunigung, relative Winkelgeschwindigkeit und absolute erdmagnetische Flussdichte in einem neun achsigen Freiheitsgrad (engl. nine Degrees of Freedom, 9DoF) messen [475].

Die Trägheitsnavigation wird in einer Vielzahl von Anwendungen eingesetzt, darunter die Navigation von Fahrzeugen, Flugzeugen und Schiffen [68]. Inertiale Navigationssysteme in militärischen Anwendungen wie Luftfahrt und Marine benötigen eine zeitaufwendige Systeminitialisierung, einschließlich Sensorkalibrierung und Orientierungsinitialisierung und verwenden hoch-präzise, teure Sensoren mit großen Abmessungen, um Messfehler gering zu halten [68]. Diese Eigenschaften sind jedoch für die Fußgängernavigation und Robotik ungeeignet [46]. Daher werden mit den jüngsten Fortschritten im Bereich der Mikro-Elektro-Mechanische Systeme-(MEMS-)Technologie zunehmend kompaktere und kostengünstigere IMU-Sensoren entwickelt, die sich platzsparend auf Platinen, wie sie in einem Mobiltelefon verwendet werden, unterbringen lassen [286].

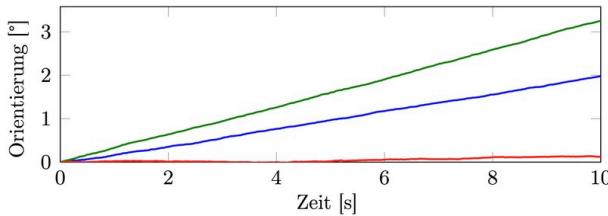


Abbildung 3.4: Mathematisch integrierte Orientierung der Position in x- (blau), y- (grün) und z-Richtung (rot) (adaptiert nach Kok et al. [241, 242]).

Jedoch sind die Messungen von MEMS-IMU durch verschiedene Fehlerquellen wie Skalierungsfaktor, Achsenversatz, thermomechanisches weißes Rauschen und zufälligem Rauschen beeinträchtigt [107]. Darüber hinaus führt insbesondere die lose Platzierung oder Befestigung der IMU am Objekt, wie etwa in der Hosentasche eines Nutzers zu zufälligen nichtdeterministischen Bewegungskräften und Gravitationsanteilen in den jeweiligen Beschleunigungssensor- und Gyroskopachsen. Diese dynamischen Bewegungen erschweren die korrekte Bestimmung der Lage und Bewegung des Sensors [128], Kapitel 5 erörtert Details. Ebenso stören bspw. metallische Gegenstände in der Umgebung des Inertialsensors die Schätzung der absoluten Orientierung mittels Magnetometer und führen zu verfälschten Messungen der absoluten räumlichen Orientierung [475]. Diese Faktoren führen zu einer Verzerrung (engl. bias) der von den IMU-Sensoren bereitgestellten kinematischen Größen [475].

Darüber hinaus beeinflussen diese Fehlerquellen die Ergebnisse der mathematischen Integration im Zeitverlauf und führen zu Drift der linearen, relativen Beschleunigung, relativen Winkelgeschwindigkeit und absoluten erdmagnetischen Flussdichte. Sodass die relative Bewegungsgeschwindigkeit, der relative Positionsversatz und die relative bzw. absolute Orientierung im Zeitverlauf von der tatsächlichen stark abweicht [243]. Schon kleine Orientierungsfehler führen zu einer falschen Projektion des Gravitationsvektors [68]. Ein Einstellungsfehler von einem Grad verursacht bspw. eine zusätzliche Beschleunigung von ungefähr 0.17 m/s^2 auf der horizontalen Ebene, der zu einem Geschwindigkeitsfehler von ungefähr 1.70 m/s , Positionsfehler von ungefähr 8.56 m (siehe Abbildung 3.3) bzw. einem Orientierungsfehler von ungefähr 3.23° (siehe Abbildung 3.4) innerhalb von 10 s führt [68, 243].

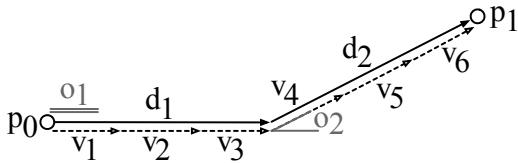


Abbildung 3.5: Beispielhafte Darstellung der Koppelnavigation über sechs Zeitschritte t . Die Geschwindigkeit v_t , die Orientierung θ_t und daraus die vektorielle (gerichtete) Distanz d_t werden in jedem Zeitschritt bestimmt. Mit einer bekannten Anfangsposition p_0 kann eine neue aktuelle Position p_1 durch einfaches Verknüpfen von p_0 und d_1 und d_2 bestimmt werden. Zur besseren Übersicht zeigt die Abbildung nicht alle Distanzen, Orientierungen und Positionen.

3.2.3 (Fußgänger) Koppelnavigation

Die (Fußgänger)Koppelnavigation (PDR), deren Ursprünge auf die Seefahrt zurückzuführen sind, ist Grundlage für viele heutige Navigationsanwendungen, darunter die Navigation von Fußgängern, Fahrzeugen und Flugzeugen [58]. Ist die Ursprungsposition p_0 eines Objektes bekannt, so lässt sich unter Zuhilfenahme der Orientierung θ und der Geschwindigkeit v oder der zurückgelegten Distanz d die laufende näherungsweise absolute Position p_1, p_2, \dots, p_n bestimmen [58]. Alternativ lässt sich die zukünftige absolute Position ebenso anhand mehrerer vergangener Positionen p , wie in Abschnitt 3.2.1 und Abbildung 3.2 beschrieben, ermitteln. Verschiedene Umgebungseinflüsse, wie etwa Sensorinterferenzen, darunter (zeitweilige) Signalverluste, -dämpfungen, -reflektionen, -drift und Sensorrauschen führen hierbei zu einer Orientierungs- bzw. Positionsabweichung. Abbildung 3.5 zeigt eine Beispielhafte Koppelnavigation über sechs Zeitschritte, wie sie in Kapitel 7 verwendet wird.

Um die Koppelnavigation hinreichend störungs- und ausfallsicher zu betreiben, werden verschiedene Lokalisierungs- und Sensorsysteme fusioniert. So werden bei der Koppelnavigation häufig Funk- oder GNSS-basierte Lokalisierungssysteme mit IMUs kombiniert [68]. Der Vorteil bei dieser kombinierten Lokalisierungstechnologie liegt darin, dass die Koppelnavigation aufgrund der Messungen der Trägheitsmesseinheit in Kombination mit den Messungen, die das Lokalisierungssystem bereitstellt, genauer und ausfallsicherer sind [211]. So können gleichermaßen die trägeitsbasierten Schätzungen die funk-

basierten Schätzungen korrigieren und umgekehrt [269]. So werden die IMU-Messungen, die bereits nach kurzer Zeit zu driften beginnen, in regelmäßigen Abständen anhand der komplementär eingesetzten Funklokalisierungstechnologie re-kalibriert (siehe Abschnitt 3.2) [269]. Da die Messdaten von funkbasierter Lokalisierungssystemen ebenfalls Signalinterferenzen (siehe Abschnitt 3.3.1) unterliegen, können inertiale Trägheitsmesseinheiten bei Funk-Signalverlust die Koppelnavigation weiterhin aufrecht erhalten [269].

Aufgrund der einzigartigen, flexiblen und modularen Architektur wird in dieser Dissertation eine PDR-ähnliche Methode vorgeschlagen. Die neuartige Verarbeitungskette umgeht oder kompensiert Schwächen herkömmlicher PDR- und SINS-Methoden, nämlich die starre und a priori definierte Sensorplatzierung, die Abhängigkeit von ZUPT und ZAUR Mechanismen und das naive Integrieren von Sensorfehlern.

3.3 Funkbasierte Lokalisierung

In diesem Abschnitt werden Grundlagen der funkbasierter Lokalisierung eingeführt. Abschnitt 3.3.1 führt die Hochfrequenz-Funklokalisierung ein. Abschnitt 3.3.2 erörtert Funkkanalparameter, die zur Lokalisierung genutzt werden. Abschnitt 3.3.3 diskutiert die Einschränkungen dieser Funkkanalparameter. In Abschnitt 3.3.4 wird die Methode zur Schätzung des Ankunftszeitparameters erläutert. Abschnitt 3.3.5 beschreibt die Lokalisierung mit Ankunftszeiten.

3.3.1 Hochfrequenzlokalisierung

Die Hochfrequenzpositionierung (HF, engl. radio frequency, RF) basiert auf Systemen, die eine geometrische Beziehung zwischen den Positionen von Sendern und Empfängern herstellen. Aktuelle 5G-Telekommunikationsstandards [1] unterstützen Ortungstechnologien auf der Basis von Funkzugangsnetzen (engl. Radio Access Network, RAN), die eine effiziente Kombination von Kommunikations- und Lokalisierungstechniken ermöglichen. Es gibt drei Hauptansätze [477, 298]: Ankunftsinkel (engl. Angle of Arrival AoA), Empfangssignalstärke (RSS) und Laufzeitmessung (engl. Time of Flight, ToF).

Während AoA teure Richtantennen oder Antennenarrays erfordert [366, 503], kämpft RSS mit erheblichen Schwankungen über kurze Entfernung [366, 320, 164] nutzt ToF die (hohe) Bandbreite, die den

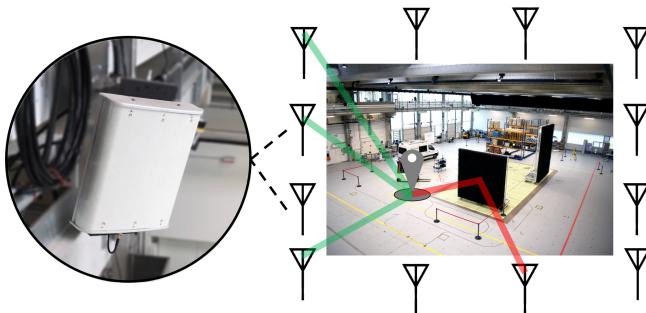


Abbildung 3.6: Systemarchitektur eines typischen Funklokalisierungssystems in Innenräumen. Empfangsantennen werden auf drei vertikalen Ebenen um eine (freie) Fläche verteilt. Die Standorte sind bekannt und a priori definiert.

5G Diensten zugewiesen ist, besser aus. Dies ermöglicht eine Positionsgenauigkeit im Zentimeterbereich unter guten Ausbreitungsbedingungen. ToF Systeme sind häufig die erste Wahl, wenn die Anwendung eine hohe Positionsgenauigkeit erfordert [503].

Bei der funkbasierten Ortung wird die absolute Position des Objektes von kleinen, tragbaren, am Objekt lose oder rigide befestigten Funksendern (engl. beacons) bestimmt. Hierbei gibt der Funksender in regelmäßigen Zeitabschnitten sogenannte Burst-Signale mit fester Frequenz und Dauer an seine Umgebung ab, welche von umliegenden Antennen empfangen werden [307]. Die Burst-Signale repräsentieren im Wesentlichen individuelle Funksignalcharakteristiken, um den jeweiligen Sender an den Empfangsantennen später zu identifizieren [307]. Die Empfangsantennen werden, wie Abbildung 3.6 veranschaulicht, rund um das Messfeld stationär an definierten Orten montiert [307].

Verarbeitungskette

In einem typischen RF-Positionierungssystem liefern mehrere synchronisierte Sende-Empfangsleitungen (engl. Transceiver-Pipelines) Sätze¹ von Kanalimpulsantworten (engl. Channel Impulse Response, CIR). Aus jedem dieser CIRs wird ein ToA geschätzt, um eine Position zu multilaterieren (siehe Abbildung 2.4). Eine genaue und robuste

¹Ein Satz CIRs enthält zu jedem Empfangszeitpunkt n CIRs, mit $n =$ Anzahl der synchronisierten Antennen des Funksystems, d. h. ein CIR pro Antenne.

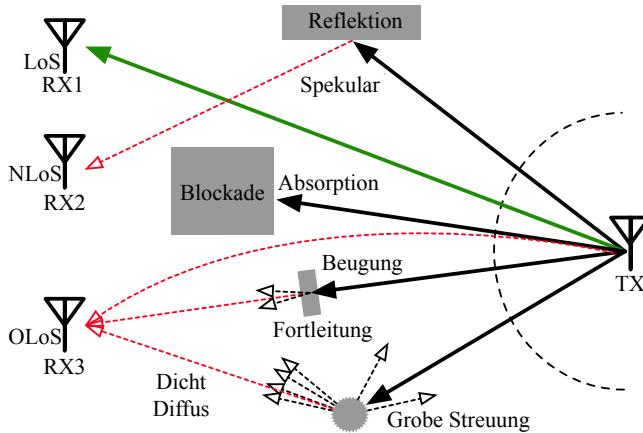


Abbildung 3.7: Mehrwege-Funkausbreitungsszenarien mit spiegelnden und dichten Mehrwegen und den entsprechenden LoS, OLoS und NLoS Szenarien (rot: betroffen; grün: nicht betroffen).

te ToA-Schätzung ermöglicht ein genaues funkbasiertes Echtzeitlokalisierungssystem (engl. Real-time Locating Systems, RLTS). Dies bleibt jedoch eine herausfordernde Aufgabe, da in der Regel ein Kompromiss zwischen der zulässigen (oder realisierbaren) Sendeleistung, der verfügbaren Bandbreite, dem Zielabdeckungsbereich und folglich der theoretisch möglichen Genauigkeit der Schätzungen von ToA besteht.

Herausforderungen

Abbildung 3.7 zeigt drei typische Szenarien: Ein Sender (rechts) sendet einen Funkstoß aus, der sich durch die Umgebung bewegt und verschiedene Empfänger (links) erreicht, d. h. LoS (Sichtlinie), NLoS (Nicht-Sichtlinie) und OLoS (Sichtbehinderung, engl. Obstructed-Line-of-Sight). Für den LoS Fall ist das Schätzen des ToA so einfach wie das Schätzen der ersten und stärksten Spitze in einem CIR, da der direkte Pfad nicht von Mehrwegkomponenten (engl. Multipath Components, MPC) beeinflusst wird. Das Signal kommt mit einer Verzögerung von mehr als $1/f_{Smin}$, d. h. länger als die Abtastperiode $T=1/f_{Smin}$ der inversen minimalen Nyquist-Shannon Abtastfrequenz f_{Smin} , an. Hier reichen einfache Algorithmen aus, um den korrekten ersten direkten Ankunftszeitpunkt (engl. First Direct Path of Arrival, FD-

PoA) und ToA [9, 177] zu identifizieren. In den Fällen OLoS und NLoS wird dieser Pfad jedoch durch Effekte beeinträchtigt, die aus Beugung, schwachen Signalen oder Reflexion resultieren. Während im Fall von OLoS möglicherweise noch ein direkter Pfad vorhanden ist, der schwer von MPC-Komponenten zu unterscheiden ist, kann im Fall von NLoS der erste ankommende Pfad im Vergleich zur ToF eines LoS-Signals erheblich verzögert sein, da diese erste Komponente viele zusätzliche Verzögerungen aufweisen kann. In solchen Fällen ist es schwierig, den wahren ToA aus dem CIR zu schätzen. In der Praxis sind sowohl die Fälle NLoS als auch OLoS sehr häufig [271, 261].

Frühere Arbeiten schätzen den ToA aus mehreren Spitzen im CIR, indem sie die maximale Spitzte [166, 165] auswählen und Schwellenwerte zusammen mit dem Signal-Rausch-Verhältnis (engl. Signal-to-Noise Ratio, SNR) und der Leistung in Bezug auf die stärkste Spitzte anwenden [419, 112] (PEAK) oder durch Schätzen der Spitzte basierend auf dem maximalen Gradienten entlang der ersten ansteigenden Flanke, d. h. der Wendepunktmethode (engl. Inflection Point Method, IFP) [193]. Solche Methoden funktionieren jedoch nicht gut, da die Mehrwegeausbreitung gegen ihre Grundannahme, dass der korrekte ToA die erste höchste Spitzte repräsentiert, verstößt und sich ihre Leistung mit abnehmendem SNR ebenfalls verschlechtert.

Darüber hinaus wurden mehrere hochauflösende ToA-Schätzer untersucht, hauptsächlich im Frequenz- und Zeitbereich in Mehrwege-szenarien [93, 257, 166]. Solche Ansätze umfassen die Mehrfachsignal-klassifikation (engl. Multiple Signal Classification, MUSIC) [360, 343] und die Schätzung von Signalparametern mit Hilfe von Rotationsinvarianztechnik (engl. Estimation of Signal Parameters via Rotational Invariance Technique, ESPRIT) [360, 441]. Sie benötigen jedoch a priori Informationen über den Kanal, und ihre Rechenkomplexität nimmt mit zunehmender Anzahl von Pfaden zu, was sie für (Echtzeit-) Anwendungen nicht mehr anwendbar macht.

Neuere datengetriebene-Ansätze, z. B. des tiefen Lernen, engl. Deep Learning, DL, verwenden Faltungs-neuronale Netze (engl. Convolutional Neural Networks, CNNs) [312, 313], um die Position direkt aus einer Reihe von CIRs zu schätzen. Die Methoden funktionieren jedoch nur für die Umgebung, für die sie trainiert wurden, und können nicht auf unbekannte Kanalbedingungen verallgemeinern.

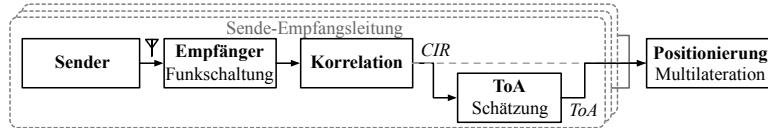


Abbildung 3.8: Typische RF-basierte Positionierung: Mehrere synchronisierte Sende-Empfangsleitungen bieten einen Satz von Daten ToA oder CIR. Von dort aus multilateriert ein Optimierer für kleinste Quadrate eine Position.

Im Rahmen der Dissertation wird ein verbesserter DL-basierter ToA-Schätzer und Trainingsstrategien für synthetische und natürliche Daten erforscht. Diese Forschungsarbeit von Feigl et al. [134] wurde noch nicht veröffentlicht. Erste Experimente zeigen, dass der DL-Ansatz einfache ToA-Schätzer (wie PEAK und IFP, aber auch MUSIC) bei SNRs unter -10 dB um durchschnittlich etwa 17% übertrifft. Im Gegensatz zu Spitzen- und Schwellenwert-basierten Methoden erfordert die DL-Methode keine *a priori* Informationen, und im Gegensatz zu Super-Resolution-Methoden verschlechtert sich ihre Leistung nicht mit der Anzahl von MPCs. Gleichzeitig lässt sich die Methode gut auf unbekannte Ausbreitungsumgebungen verallgemeinern.

Im Rahmen der Dissertation wird nicht explizit auf die Schätzung optimaler ToA Werte, die optimale FDPOAs aus CIR Daten abbilden, eingegangen. Die Datengrundlage der Positionsschätzung in Kapitel 4 bilden ToA Werte, die mit der IFP Methode (Abschnitt 3.3.4) ermittelt wurden. Somit enthalten die ToA Werte möglicherweise Schätzfehler, die nachfolgend genauer beschrieben werden.

Signal- und Kanalparameter

Funkübertragungen werden durch Funksignal- und Funkkanalparameter beschrieben, die auch den Schwund (engl. Fading) definieren. Schwund beschreibt Schwankungen der Funkempfangsstärke durch (destruktive) Interferenz (engl. interference), Abschattungen (engl. shadowing), Mehrwegeausbreitungen (engl. multipath) und den Doppler-Effekt. Es gibt zwei Arten von Schwund: Schwund in großem Maßstab (engl. Large-Scale-Fading, LSF) und Schwund im kleinen Maßstab (engl. Small-Scale-Fading, SSF).

Schwund in großem Maßstab, LSF. LSF stellt die durchschnittliche Dämpfung der Signalleistung oder den Pfadverlust aufgrund von Funkausbreitung über große Übertragungsentfernung dar und wird durch die räumliche Umgebung zwischen Sender und Empfänger sowie über die Entfernung, in der Größenordnung von Kilometern, beeinflusst, die zu einer stetigen Leistungsabnahme führen.

Schwund in kleinen Maßstab, SSF. Im Rahmen der Dissertation werden explizit Funkausbreitungen in Innenräumen mit Entfernungen < 100 m untersucht, die durch SSF beschrieben werden. SSF bezieht sich auf die schnellen Änderungen der Amplitude und Phase eines Funksignals über einen kurzen Zeitraum, in der Größenordnung von Sekunden, oder eine kurze Entfernung, einige Wellenlängen. Durch SSF kann die empfangene Signalleistung um bis zu 40 dB variieren, wenn der Empfänger nur um einen Bruchteil einer Wellenlänge bewegt wird.

In einer Funkumgebung hat jeder Pfad seine eigene Doppler-Verschiebung, Zeitverzögerung und Wegdämpfung, sodass die Mehrwegeausbreitung zu einem zeitlich variierenden Signal führt, wenn der Sender seine Position verändert. Ein solcher Kanal und seine Parameter variieren daher zeitlich. SSF wird als Rayleigh-Fading bezeichnet, wenn keine Sichtlinie zwischen Sender und Empfänger vorhanden ist. Wenn es eine Sichtlinienkomponente gibt, wird diese durch eine Rician-Verteilung beschrieben.

SSF hängt von der Art des übertragenen Signals, z. B. von Signalparametern wie der Bandbreite und der Symbolperiode, in Bezug auf die Eigenschaften des Kanals ab, z. B. Kanalparameter, wie die Kohärenzzeit, die Doppler-Spreizung, die Kohärenzbandbreite und die Verzögerungsspreizung. Ist die Ausbreitung nicht ideal, ideal wäre eine ungestörte Ausbreitung des freien Raums, können Funksignale von Objekten, wie z. B. Hochregalen, Wänden und Fußböden, gebrochen oder reflektiert werden. Es tritt eine Mehrwegeausbreitung auf, die das Signal an der Empfangsanenne verfälscht: Bei destruktiven Interferenzen kann die Phasenposition der eingehenden Signale auf mehreren Wegen verschoben werden, die Ankunftszeit verzögert sich, sodass im schlimmsten Fall sogar eine gegenseitige Signalauslöschung erfolgt. [151]

3.3.2 Kanalparameter zur Lokalisierung

Ein Funkkanal, der die Umgebung charakterisiert, wird normalerweise durch eine Reihe von Parametern definiert, die seine MPCs ma-

3 Grundlagen

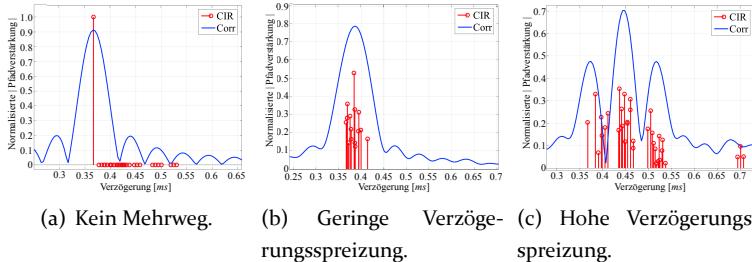


Abbildung 3.9: Kanalimpulsantworten (CIR) und zugehörige Korrelatorausgabe (Corr) verschiedener Verzögerungsspreizung (rot: unbegrenzte Bandbreite; blau: begrenzte Bandbreite).

thematisch beschreiben. Somit beschreibt die Kanalschätzung (engl. Channel Estimation, CE) die Ausbreitung eines Signals und wird für die Empfänger verwendet, um die betreffenden Signale zu rekonstruieren.

Die Eingabe in einen ToA-Schätzer ist normalerweise ein CIR, der durch Dekorrelation eines empfangenen Signals $y(t)$ mit einer bekannten Pseudozufallssequenz $s(t)$ extrahiert wird. Die CIR beschreibt den Einfluss von Pfadverlust und Materialwechselwirkungen für einen bestimmten Signalpfad. Der ToA-Schätzer extrahiert eine Signalausbreitungszeit aus einer Reihe von CIR-Beobachtungen synchronisierter Empfänger. Somit muss er die Deltazeit identifizieren, d. h. die relative ToA $\delta(t - T_i)$ in $h(t)$, die dem FDPOA entspricht. Der Abstand zwischen Sender und Empfänger wird dann mit der Lichtgeschwindigkeit bestimmt, um eine Position zu multilaterieren, siehe Abbildung 3.8.

Unter der Annahme eines (idealen) LoS-Kanals mit kleinen MPCs und unbegrenzter Bandbreite wird eine Korrelation erhalten, indem das empfangene Signal mit dem gesendeten Signal korreliert wird:

$$h(t) \approx y(t) * s(-t), \quad (3.9)$$

wobei $*$ eine Faltung ist. Das CIR wird als eine Summe von N_s einzelnen Signalpfaden modelliert, dargestellt durch Dirac-Pulse $\delta(t - T_i)$, die mit dem CIR der bandbreitenbegrenzten Filter mit einem ausgerichteten komplexen Gewicht α_i gefaltet werden. Somit bezieht sich T_i auf die

Signalausbreitungszeit und α_i beschreibt den Einfluss von Pfadverlust und Materialwechselwirkungen für einen bestimmten Signalpfad:

$$h(t) \approx \sum_{i=0}^{N_s} \alpha_i \delta(t - T_i). \quad (3.10)$$

Die tatsächlich beobachteten CIRs können sich jedoch in einem bandbegrenzten Kanal mit der Bandbreite B_w erheblich unterscheiden. Die Einschränkungen werden als Multiplikation mit einer Rechteckfunktion im Frequenzbereich beschrieben. Dies entspricht einer Faltung des empfangenen Signals mit einer normalisierten *sinus cardinalis* (SINC)-Funktion im Zeitbereich, was zu einer signifikanten Überlappung der in der CIR enthaltenen räumlichen Informationen führt. Somit mischt die Faltung die verschiedenen Impulse miteinander. Das resultierende bandbegrenzte CIR $h_{B_w}(t)$ ist:

$$h_{B_w}(t) \approx \left(\sum_{i=0}^{N_s} \alpha_i \delta(t - T_i) \right) * 2 B_w \text{ SINC}(2B_w t). \quad (3.11)$$

Die Abbildungen 3.9 (a-c) zeigen den Einfluss verschiedener Ausbreitungsbedingungen auf den Betrag der CIRs. Das CIR ganz links enthält nur einen einzigen Impuls (rot). Die Faltung mit der Funktion SINC führt zu dem blau dargestellten Korrelationssignal. Wenn es mehrere Spitzen gibt, vgl. Abbildung 3.9 (b) und (c), ergibt sich das korrelierte Signal aus einer Überlappung der SINCs, die durch die Faltung mit den Dirac-Impulsen erzeugt wird. Die Spitzen sind schlechter unterscheidbar und die FDPOA kann nicht korrekt identifiziert werden.

3.3.3 Einschränkungen von Kanalparametern

Das Extrahieren der Informationen, die den richtigen ToA darstellen, ist eine Herausforderung. Bei Mehrwegeausbreitung enthält das CIR viele verschiedene MPCs. Darüber hinaus begrenzt die begrenzte zeitliche Auflösung, insbesondere in Schmalbandkanälen, die Genauigkeit von ToA. Die Kreuzkorrelation des empfangenen mit dem gesendeten Signal setzt sich aus vielen SINC-Funktionen zusammen: Abbildung 3.9 (b) zeigt einen einzelnen Cluster von CIRs, das eine einzelne Spurze in der Korrelationskurve bildet, während Abbildung 3.9 (c) mehrere Clusters zeigt, die drei Spitzen in der Korrelationsfunktion bilden.

3 Grundlagen

Die Cluster in Abbildung 3.9 (c) zeigen zwei typische Beispiele für 1) einen OLoS-Fall, in dem der FDPOA, der erste Cluster, weniger verzögert, aber auch schwächer ist aufgrund von Beugung und Transmission, siehe auch die Pfade zwischen Tx und Rx_4 in Abbildung 3.7; und 2) mehrere NLoS-Fälle, der zweite und der dritte Cluster, sind stärker verzögert aber leistungsfähiger aufgrund von Reflexion und Streuung siehe die Pfade zwischen Tx und Rx_1 , Rx_2 und Rx_4 in Abbildung 3.7.

Daher hängt die erwartete ToA-Schätzleistung von der gesamten Kanalstatistik ab, die wiederum von der Umgebung und den Bereitstellungsparametern abhängt. Das quadratische Mittel (engl. Root-Mean-Square) der Verzögerungsspreizung (engl. *Delay Spread*, DS) und der *Ricean k-Faktor* (kF) werden normalerweise zur Charakterisierung eines CIR verwendet. Diese statistischen Modellparameter werden aus Kanalmessungen abgeleitet und beschreiben am besten die Ausbreitungsbedingungen innerhalb einer Umgebung: (1) kF ist definiert als:

$$kF = P_{\text{LoS}} / P_{\text{NLoS}}, \quad (3.12)$$

wobei P_{LoS} die Potenz des direkten Pfads und P_{NLoS} die Summe der Potenz aller Mehrwegkomponenten ist und (2) DS, definiert als:

$$DS = \sqrt{\frac{1}{P_i} \cdot \sum_{l=1}^L P_l \cdot (\tau_l)^2 - \left(\frac{1}{P_i} \cdot \sum_{l=1}^L P_l \cdot \tau_l \right)^2}, \quad (3.13)$$

wobei P_i die Gesamtleistung darstellt, P_l die Leistung des Clusters l und τ_l die normalisierte Clusterverzögerung.

DS und kF sind beide umgebungsabhängig und können durch statistische Verteilungen modelliert werden. Wenn DS im Vergleich zu $1/fSmin$ niedrig ist, überlappen sich mehrere SINC-Funktionen und bilden eine neue Form der Korrelationsfunktion. Die Wirkung von DS, wenn die Bandbreite begrenzt ist, ist in Abbildung 3.9 dargestellt: (von links nach rechts) Bei höheren DS gibt es mehr MPCs Cluster, siehe die drei Cluster von MPCs in Abbildung 3.9 (c), und es wird schwieriger, die richtigen FDPOA zu identifizieren. Die DS gibt eine grobe Schätzung, wie weit MPCs verzögert sind, d. h. die Verzögerung zwischen dem MPC, der das FDPOA am positivsten beeinflusst, und dem MPC, das es am negativsten beeinflusst. Im Gegensatz dazu, gibt der kF eine grobe Schätzung, wie viele MPCs die FDPOA Schätzung negativ beeinflussen. Hohe DS- und niedrige kF-Werte stellen eine Worst-Case-

Situation dar, um einen korrekten FDPOA in einem bandbegrenzten Kanal zu schätzen. Dies wird schlimmer mit niedrigeren SNRs.

Umgang mit großem DS, niedrigem kF und niedrigem SNR

In Bezug auf die Interaktion mit der Umwelt erhöhen städtische, industrielle und andere Umgebungen mit vielen streuenden, absorbierenden und reflektierenden Objekten die Vielfalt der Ausbreitungspfade und MPCs und damit der Schätzfehler ToA. Dies wird besonders kritisch, wenn die Bandbreite und die Sendeleistung ebenfalls begrenzt sind. Der unveröffentlichte datengetriebene Ansatz [134] identifiziert FDPOAs auch in diesen komplizierten Szenarien, da er ihre komplexe räumliche Korrelation von Merkmalen aus Schnappschüssen von CIRs lernt. In Kapitel 4 wird implizit die ToA-Fehlervarianz genutzt, um die Positionsgenauigkeit zu verbessern.

3.3.4 Ankunftszeiten Schätzer

Dieser Abschnitt stellt die drei Hauptmethoden zum Schätzen der Ankunftszeiten anhand von CIRs vor. Die naive PEAK-Methode, die Super-Auflösungs-Methoden und die IFP-Methode, die im Rahmen der Dissertation zur ToA-Schätzung verwendet wird.

Spitzen- und Schwellwert (PEAK)

Die einfachste Technik schätzt den ToA, indem die maximale Spitze der Korrelationsfunktion [165] ermittelt wird. Andere ToA-Schätzer basieren auf Schwellenwerten und verwenden feste SNR- und Leistungswerte [419, 112]. Sie berücksichtigen alle Spitzen oberhalb einer Schwelle in Bezug auf die stärkste Spitze. Der Schwellenwert muss jedoch von Hand eingestellt werden und funktioniert nur in LoS-Situationen.

Super-Auflösung

Zusätzlich zu PEAK und IFP werden mehrere Super-Auflösungs-(engl. Super-Resolution-)Algorithmen vorgeschlagen, die neben ToA auch verschiedene andere Kanalparameter [398, 271, 459] im Frequenz- und Zeitbereich in Mehrwegeausbreitungsszenarien schätzen (engl. Channel Estimation, CP) [271]:

Minimale Norm [446], lineare Vorwärts-Rückwärts-Vorhersage [138], Singulärwertzerlegung [473] und (normalisierte) minimale Varianz

eines CIR [445]. Häufig wird die Erwartungsmaximierung verwendet, um die Kanalzustandsinformation (engl. Channel State Information, CSI) einzelner MPCs mithilfe von Subraumanpassungsalgorithmen wie Multiple Signal Classification (MUSIC) [271, 208, 360] und Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) [360] zu schätzen. Diese CP-Schätzer benötigen eine Fourier-Transformation, die das empfangene Signal in den Frequenzbereich übersetzt. Diese verursacht jedoch einen hohen Rechenaufwand. Da diese Methoden a priori Informationen erfordern, decken sie möglicherweise nicht alle möglichen Ausbreitungsszenarien zur Laufzeit ab.

Stattdessen basieren iterative Maximum-Likelihood-Schätzer wie Space Alternating General Expectation (SAGE) und Richter MAXimum likelihood (RiMAX) [360, 169] auf maximaler Wahrscheinlichkeit und weisen eine geringere Rechenkomplexität auf. Solche Schätzer sind genauer und robuster als PEAK und IFP CP-Schätzer. Während Superauflösungstechniken die Zeitbereichsauflösung erhöhen und somit die ToA in Umgebungen mit mehreren Pfaden genauer schätzen können, nimmt ihre Rechenkomplexität in solchen Umgebungen erheblich zu [360, 381]. Daher sind sie für reale Anwendungen unpraktisch und werden im Rahmen dieser Dissertation nicht untersucht.

Wendepunkt Verfahren (IFP)

Das IFP Verfahren [193] ist ein ToA-Schätzer, der anstelle des Maximums der Korrelatorausgabe nach dem Index mit dem maximalen Gradienten der ansteigenden Flanke (linken Kante) der Korrelationskurve sucht. Um den ToA mit IFP zu schätzen, wird zunächst der maximal tolerierbare Abstand d definiert zwischen dem Maximum des Korrelationspeaks und dem am weitesten links liegenden Wendepunkt $\delta = \max(d) = -10$, mit Upsampling- oder Interpolationsfaktor $L = 32$, dem Startindex der Wendepunktsuche $i = 1$ und der Gruppenverzögerung des Filters $\tau = 0$ im absoluten Korrelationssignal $c = |\text{Corr}|$.

IFP findet dann durch einen relativen Schwellenwert entweder die erste relevante Spitze p_0 oder die ungefähre Position von p_0 , wobei zum ersten Mal der Schwellenwert p_τ bezüglich des Maximalbetrags $\max(\text{val})$ erreicht wird. Der Spitzenschwellenwert p_τ wird gemäß dem SNR von c ausgewählt: wenn $\text{SNR} > -10$ dann $p_\tau = -5$; oder wenn $\text{SNR} > -20$ dann $p_\tau = -3$; oder wenn $\text{SNR} \leq -20$ dann $p_\tau = -2$.

Damit die Methode nicht mit p_τ auf eine sekundäre Welle aufläuft, muss der erste Pfad kleiner als $10 \cdot \log_{10}(p_\tau)$ sein:

$$\mathbf{p}_o = \text{find}(c \geq \max(val) * p_\tau). \quad (3.14)$$

Das Ende der Spitzensuche wird folgendermaßen bestimmt:

$$\mathbf{p_n} \lceil (p_0 - (2 * \delta) * L). \quad (3.15)$$

Der Betrag der Korrelationsfunktion hat vor jeder Spitze einen Wendepunkt, d. h. einen Punkt mit einem Maximum in seiner ersten Ableitung. Dann wird nach dem Index links vom linken Wendepunkt gesucht, d. h. dem letzten Element mit einer positiven zweiten Ableitung. Der Wendepunkt an der ansteigenden Flanke (linke Kante) wird gesucht, solange die zweite Ableitung von $p_0 \geq 0$ ($\frac{\partial p_0^2}{\partial t^2}$) eine u-konvexe Lösung ist. Die Suche endet nach dem Wendepunkt und bestätigt den korrekten Index vor dem Wendepunkt.

Stattdessen gilt für n-konvexe Lösungen: Wenn die zweite Ableitung von $p_0 \leq 0$, $\frac{\partial p_0^2}{\partial t^2}$, d. h. wenn der Wendepunkt links ist, fährt der Algorithmus links fort. Die Position der max. Spitze p_i ist bestimmt durch:

$$\mathbf{p_i} = \lceil (p_0 - \delta * L). \quad (3.16)$$

Wenn die erste Ableitung von $p_i \geq 0$ ($\frac{\partial p_0}{\partial t}$) ist, ist das Maximum rechts von der aktuellen Position. Wenn die erste Ableitung von $p_i < 0$ ($\frac{\partial p_i}{\partial t}$) ist, befindet sich das Maximum links von der aktuellen Position. p_i repräsentiert den Index des Punktes unmittelbar links vom Wendepunkt der ansteigenden Flanke (linke Flanke, le):

$$\mathbf{le} = p_i + \frac{\partial p_i^2 / \partial p_i}{\partial t^2} - \frac{\partial p_{i+1}^2 / \partial p_i}{\partial t^2}. \quad (3.17)$$

Die genaue Position des Wendepunkts wird durch lineare Interpolation berechnet. Die Gruppenverzögerung (τ) des Filters wird von le subtrahiert und ToA_{rel} wird bestimmt:

$$\mathbf{ToA_{rel}} = \frac{le - \tau}{L} - \delta. \quad (3.18)$$

Das IFP Verfahren wird im Rahmen der Dissertation eingesetzt, um die ToA Werte zu schätzen, die in Kapitel 4 zur Positionierung verwendet werden [134]. Die ausgereifte Wendepunktmethode [193] schätzt die

3 Grundlagen

ToA basierend auf dem maximalen Gradienten entlang der ersten ansteigenden Flanke. Dies hilft, die durch MPCs verursachten Spitzenerkennungsfehler mit ähnlichen Verzögerungen zu berücksichtigen. Obwohl IPF eine Verbesserung gegenüber PEAK darstellt, leidet es immer noch unter Mehrwegeausbreitung und niedrigen SNRs.

3.3.5 Lokalisieren mit Ankunftszeiten

Da die Position der Empfangsantennen sowie die Ausbreitungsduer von elektromagnetischen Signalen vorab bekannt ist, lässt sich die ToA und ferner die Zeitdifferenz zwischen den Signalankunftszeiten (TDoA) oder alternativ der Einfallswinkel der Signale (AoA) an jeder zeitsynchronisierten Antenne in regelmäßigen Abständen messen [307]. Mittels Multilaterationsverfahren [515], unter Einbezug mehrerer (mindestens drei) Empfangsantennen, lässt sich der Abstand zwischen tragbarem Funksender und den Empfangsantennen bestimmen und sich so die absolute Position schätzen [179]. Die zwei $TDoA_2$ und $TDoA_3$ Werte mit drei Empfängern sind gegeben durch $TDoA_2 = ToA_2 - ToA_1$ und $TDoA_3 = ToA_3 - ToA_1$, wobei ein ToA_1 eines Empfängers 1 wird zufällig ausgewählt, um die Zeitdifferenz zwischen den beiden anderen Empfängern 2 und 3 zu bestimmen. Die Entfernung kann mit Hilfe der Konstante der Lichtgeschwindigkeit bestimmt werden. Mit einem Optimierungsprozess, wie z. B. Bancroft [26] oder Levenberg-Marquardt [260], kann der gemeinsame Schnittpunkt, die Position, im Koordinatensystem für bekannte Empfängerstandorte bestimmt werden. Die Orientierung des Senders lässt sich bei funkbasierten Ortungssystemen anhand mehrerer konsekutiver Positionsdaten schätzen, siehe Abschnitt 3.2.1, Abbildung 3.2.

In LoS Szenarien, wenn die Funksignale den mobilen Sender und alle involvierten Antennen ungehindert verbinden [509], lässt sich die absolute Objektposition nach der Korrektur möglicher Ausreißer durch Kalman-Filterung mit einer Positionsgenauigkeit von weniger als 0.25 m ermitteln [269]. Wird der direkte Signalausbreitungsweg jedoch durch Hindernisse wie Personen oder anderen Gegenständen behindert, erreicht das Signal die Empfangsantennen über reflektierte, gebogte oder gestreute Wege zeitverzögert [509]. In solchen OLoS- oder NLoS-Fällen wird das Funksignal mehrfach von den Empfangsantennen zeitverzögert empfangen und führt zu abweichenden Reichweitendifferenzberechnungen, welche wiederum zu einem Positionsversatz und

ungenauen Positionen oder zu einem zeitweiligen Ausfall des Lokalisierungssystems führen [269].

3.4 Datenvorverarbeitung

In diesem Abschnitt werden Grundlagen der Datenvorverarbeitung eingeführt. Abschnitt 3.4.1 führt die Signalerfassung ein. Abschnitt 3.4.2 erörtert Verfahren zur Datenaggregation. Abschnitt 3.4.3 beschreibt Verfahren zur Reinigung verrauschter Daten. In Abschnitt 3.4.4 werden Methoden zur Synchronisation verschiedener Sensordatenströme beschrieben. Abschnitt 3.4.5 erläutert Verfahren zur Datenreduktion und -erweiterung. Abschnitt 3.4.6 führt die Transformation, Skalierung und Normalisierung der Daten ein.

Abbildung 3.10 veranschaulicht den Zusammenhang zwischen Signalerfassung (links), der anpassbaren und skalierbaren Datenvorverarbeitung (mittlerer Block) und der Zeitreihenanalyse (rechts). Die unterschiedlichen Farben und Architekturen der Würfel deuten den jeweiligen Arbeitsschritt an. Jeder Würfel entspricht einem Datenpunkt oder Merkmal. Die Datenvorverarbeitung bildet das Bindeglied zwischen Signalerfassung und Datenanalyse. Die mit einer oder mehreren Sensormodalitäten aufgezeichneten Zeitreihendaten werden aggregiert, Fehler in den Messdaten bereinigt, Zeitreihendaten unterschiedlicher Herkunft synchronisiert, Datenattribute aus den Zeitreihendaten ausgeschlossen bzw. neue erzeugt und schließlich die Zeitreihendaten in ein homogenes, für die Datenanalyse kompatibles Zielformat transformiert [83, 373]. Durch die Maßnahmen zur Fehlerminimierung wird die Datenqualität und damit die Datenanalyse für die Laufzeit-basierte Posenschätzung verbessert und andererseits die Genauigkeit und Zuverlässigkeit zur Laufzeit des Datenvorverarbeitungs- und analyseprozesses verbessert. Die Datenvorverarbeitung wird durch Permutation, Weglassen oder Einbeziehen der Verarbeitungsschritte optimiert [83].

3.4.1 Signalerfassung

Sensoren erfassen analoge (multimodale) Informationen der Umgebung, wie z. B. elektromagnetische Wellen (Funksignale), Trägheitskräfte (Beschleunigung), Winkelgeschwindigkeiten (relative Orientierung) oder magnetische Flussdichten (absolute Orientierung) als Zeitreihendaten [187]. Eine Zeitreihe ist eine chronologische Folge von ein-

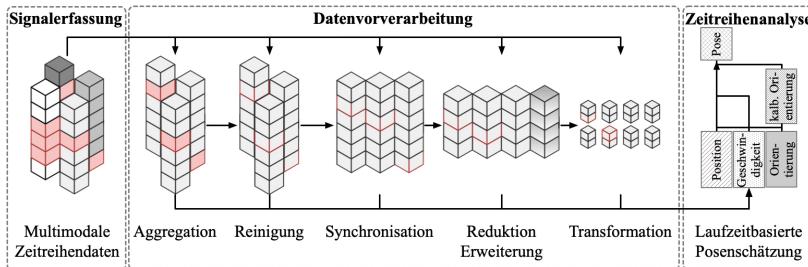


Abbildung 3.10: Verarbeitungskette vom Sensor zur Eingabesequenz: Signalerfassung, Datenvorverarbeitung und Zeitreihenanalyse.

oder mehrdimensionalen konsekutiven numerischen Werten [144]. Um digitale Zeitreihendaten zu erzeugen, wird das analoge Signal in festen Zeitintervallen abgetastet und in eine zeit- und wertdiskrete Repräsentation per Quantisierung transformiert [424].

Während die Auflösung des Wertebereichs von der Wortbreite und der Präzision des Sensors abhängt, wird die zeitliche Auflösung durch die Abtastrate des Sensors bestimmt [407]. Die Präzision von Sensoren hängt im Wesentlichen von der Qualität des Fertigungsprozesses sowie der werksseitigen Sensorkalibrierung ab und wird zusätzlich durch zufällige physikalische Störeinflüsse, z. B. (nicht)deterministisches Rauschen durch menschliche Bewegung, Temperaturschwankungen und Materialermüdung beeinflusst. Fertigungsbedingte Unterschiede sind selbst bei teuren Sensoren unvermeidbar und führen bspw. zu divergierenden Skalierungsfaktoren und zu Sensorverzerrung (engl. bias). Zufällige physikalische Störeinflüsse, die auf die Messung einwirken, sind nicht-deterministisch, schwierig zu bestimmen und führen zu Sensorrauschen oder Signalausfällen. [210] Diese Fehler werden zum Teil vorab durch Sensorkalibrierungen oder in der Datenvorverarbeitung korrigiert und werden in klassischen Laufzeit-basierten Bayes-Verfahren in die Modellierung einbezogen [197, 210].

3.4.2 Aggregation

Um die Grundcharakteristiken, nämlich Wertebereiche, Längen, Ausreißer, Lücken und Rauschverhalten, der aufgezeichneten Zeitreihendaten zu bestimmen, werden statistische Kenngrößen, wie z. B. Standardabweichung, Varianz, Minimum und Maximum ermittelt. Da-

nach werden unterschiedliche metrische Einheiten per Faktormultikplikation standardisiert. Redundante oder fehlerhaften Datenwerte und Zeitstempel in den Zeitreihendaten werden entfernt oder markiert und abschließend in einem Datenformat strukturiert. [83, 373]

3.4.3 Bereinigung

Verunreinigte Zeitreihen, die bspw. mit unterschiedlichen Raten abgetastet (engl. samplingrate) wurden oder Lücken oder Ausreißern enthalten, führen zu einer Verschlechterung der Datenanalyse und müssen korrigiert werden [83].

Unterschiedliche Abtastrate

Da Sensoren unterschiedlicher Systeme mit unterschiedlicher Abtastrate aufzeichnen, muss die Abtastrate angeglichen werden, um eine einheitliche, zeitkritische Datenanalyse zu ermöglichen. Unter Berücksichtigung des Nyquist-Shannon-Abtasttheorems werden alle gegebenen Zeitreihendaten auf eine gemeinsame Zeitbasis per Stichprobewiederholung und Interpolation angeglichen. Typischerweise werden die Zeitreihenserien zunächst in eine höhere Abtastrate durch Hochtaktung (engl. upsampling) und Interpolation überführt, Datenströme synchronisiert und weiter verarbeitet und anschließend das Heruntertakten (engl. downsampling) auf die Zielabtastrate durchgeführt. [373]

Datenlücken

Zeitweilige Signalverluste werden korrigiert indem die Zeitreihe zunächst per Stichprobewiederholung vergrößert und anschließend die wertmäßige Interpolation der fehlenden Datenpunkte durchgeführt wird. Typischerweise erfolgt die Interpolation anwendungs- und situationsspezifisch anhand von Polynomzügen n -ten Grades (Splines) oder etwa durch Einfügen eines Fehlerwertes, einem benachbarten Wert oder linear anhand dem Mittelwert oder Median des vorangegangenen und nachfolgenden Wertes [373].

Sensorrauschen

Sensorrauschen führt zu driftenden Zeitreihendaten mit Ausreißern und falschen Werten, die mit spezifischen Filtern korrigiert werden. Im Folgenden werden drei renommierte Filtervarianten erörtert: gleitende

3 Grundlagen

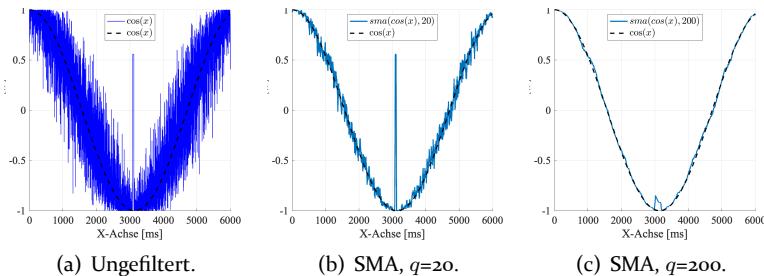


Abbildung 3.11: Gleitendes Mittelwert Filter (engl. Simple Moving Average, *sma*) am Beispiel ungefilterter Daten (10 dB additives weißes Gaußsches Rauschen, engl. Additive White Gaussian Noise, AWGN) und gefilterter Daten.

statistische Maße, Exponential- und Tiefpassfilter [373].

Gleitende statistische Maße. Die gleitenden statistischen Maße betrachten für jeden Wert x_k einer Zeitreihe der Beschaffenheit $k = 1, 2, 3, \dots, n$, die unmittelbare Nachbarschaft zu diesem Wert und geben zu den n Vorgängern bzw. Nachfolgern ein statistisches Maß wie den Median oder den Mittelwert als gefilterten Wert y_k aus [373]. Abbildung 3.11 (links) zeigt eine Kosinusfunktion als Zeitreihe, die mit Rauschen und Ausreißern (bei $x=3000$) behaftet ist [373]. Ein asymmetrisches gleitendes Mittelwert Filter verringert das Rauschverhalten sowie die Ausreißer der Ursprungsfunktion mit dem Filterparameter $q = 21$ (Mitte) und $q = 201$ (rechts) [373].

Exponentialfilter. Das Exponentialfilter ist eine Fensterfunktionen, die Zeitreihendaten glättet und bspw. als Tiefpassfilter hochfrequentes Rauschen entfernt. Es ist definiert als $y_k = y_{k-1} + \eta * (x_{k-1} - y_{k-1})$, wobei $k = 1, 2, 3, \dots, n - 1$ ist. Der momentane Filterausgang y_k wird berechnet, indem der vorhergehende Filterausgang y_{k-1} vom vorhergehenden Filtereingang x_{k-1} abgezogen und anschließend mit dem Korrekturwert $\eta \in [0, 1]$, multipliziert und mit dem vorhergehenden Filterausgang y_{k-1} addiert wird. Da das Filter am effektivsten bei langsamem Datenänderungen ist, sollte jeder Wert y_k des Filterausgangs dem vorhergehenden Wert y_{k-1} ähnlich sein.

Abbildung 3.12 (rechts) zeigt die Ausgabe von unterschiedlich parametrisierten Exponentialfiltern (mit $\eta = 0.1, 0.01$ und 0.001)

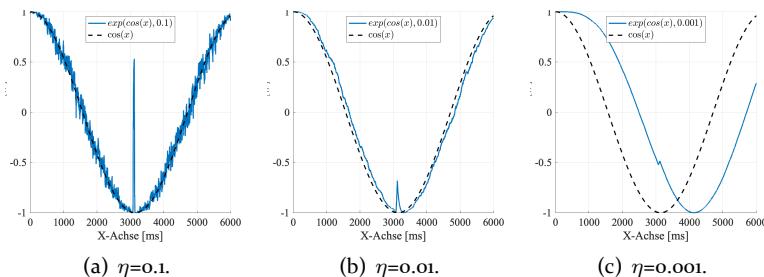


Abbildung 3.12: Exponentialfilter (\exp) am Beispiel gefilterter Daten, siehe Abbildung 3.11, mit unterschiedlichen Filterparametern η .

für dieselbe Kosinusfunktion wie in Abbildung 3.11 (links). Je kleiner η , desto mehr wird das Rauschen und Ausreißer entfernt, als mit einem Median-Filter. Allerdings führt ein kleines η zu zeitlichem Versatz zwischen Ein- und Ausgangsdaten, sodass der Filterausgang, dem Filtereingang, bspw. für $\eta = 0.001$, nicht mehr folgen kann. Daher muss der Wert für η bei einem Exponentialfilter sorgfältig und anwendungsspezifisch gewählt werden. [373]

Butterworth-Tiefpassfilter. Abbildung 3.13 zeigt die Ausgabe eines Butterworth-Tiefpassfilters zweiter Ordnung. Für $\omega_g = 0.01$ werden weißes Rauschen und seltene Ausreißer stark reduziert. Kleinere Grenzfrequenzen verursachen Effekte, eines exponentiellen Filters (siehe Abbildung 3.12). Der Filterausgang wird mit kleinerem ω_g immer stärker verzögert und wird für $\omega_g \rightarrow 0$ zu 0.

3.4.4 Synchronisation

Grundsätzlich unterliegen Micro-Controller in Funksystemen oder IMUs, thermo-mechanischen und fertigungsbedingten Einflüssen, die dazu führen, dass die Systemzeit mittelfristig von der tatsächlichen abweicht, sogenannter Uhrendrift [390]. Daher muss die lokale Zeitbasis eines Micro-Controllers auf eine allgemeingültige universale Zeitbasis in regelmäßigen Abständen angeglichen werden [390]. Dann ist der Informationsgehalt in den uni- bzw. multivariaten Zeitreihen unterschiedlicher Quellen hinreichend zeitlich synchron und beginnt und endet gleichzeitig zu einem definierten Zeitpunkt [408]. Die lo-

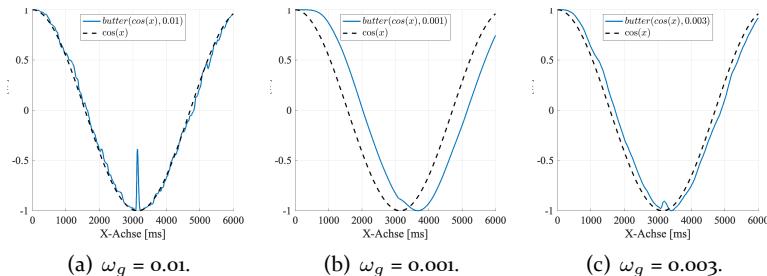


Abbildung 3.13: Butterworth-Tiefpassfilter (*butter*) zweiter Ordnung am Beispiel gefilterter Daten, siehe Abbildung 3.11, mit Filterparameter ω_g .

kale Zeitbasis der verschiedenen Micro-Controller der verschiedenen Lokalisierungs-, Ortungs- oder Sensorsysteme werden zur Sensorfusion vorab mit jeweils einer globalen Zeitbasis, wie der koordinierten Weltzeit (engl. Coordinated Universal Time, UTC) über das Network Time Protocol (NTP) synchronisiert [408].

3.4.5 Datenreduktion und -erweiterung

Datenreduktion entfernt einzelne Datenpunkte aus der multivariaten Zeitreihe oder verringert deren Informationsgehalt per Heruntersaktung. Ziel ist es, den Rechenaufwand und die Komplexität zu reduzieren, ohne wichtige Informationen zu verlieren. Bspw. schließt die schrittweise Rückwärtseliminierung unwesentliche Daten, die (keinen) Einfluss auf die Datenanalyse haben, aus dem Datensatz aus. [83]

Im Gegensatz dazu, werden durch die Reduktion der Dimension der Daten, bspw. durch Extraktion von statistischen, empirischen oder spektralen Merkmalen, die Zeitreihendaten reduziert oder sogar um zusätzliche Informationen erweitert. Diese Merkmalsvektoren enthalten dann bspw. Rohdaten und daraus extrahierte Merkmale. [231]

Zeitreihendaten werden erweitert, um die Länge, Größe und Informationsgehalt der Zeitreihe oder der Merkmalsvektoren zu vergrößern. Diese Mechanik kann die Genauigkeit der Zeitreihenanalyse verbessern. Bspw. können Datenpunkte, die durch Heruntersaktung verworfen werden per Indexverschiebung als neuer zusätzlicher Merkmalsvek-

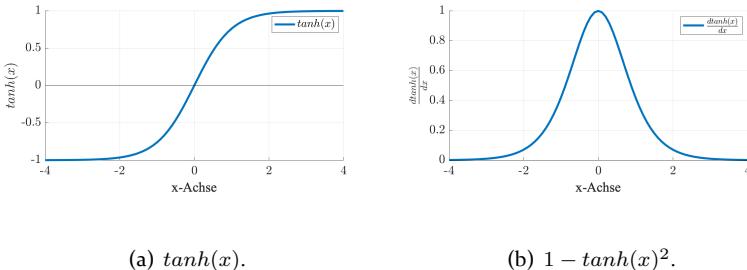


Abbildung 3.14: Tangens hyperbolicus $\text{TanH}(x)$ -Aktivierungsfunktion (links) und Ableitung, bzw. Gradient, $1 - \text{TanH}(x)^2$ (rechts).

tor verwendet werden und so der Informationsgehalt des gesamten Datensatzes erhöht werden. [417]

3.4.6 Transformation

Die Datentransformation wandelt die vorverarbeiteten Daten in eine für die Analysemethode passende Struktur [83]. Dabei werden Merkmalsvektoren zusammengefasst oder gefenstert und unterschiedliche Wertebereiche normiert.

Der Signalgrößenvektor (engl. Signal Magnitude Vector, SMV) [128] fasst mehrere Merkmalsvektoren, wie z. B. die drei Beschleunigungsachsen, zusammen, um bspw. die lineare Beschleunigung zu ermitteln:

$$SMV = \sqrt{x^2 + y^2 + z^2}. \quad (3.19)$$

Die Fensterung der Zeitreihendaten reduziert die Komplexität und Rechenzeit pro Verarbeitungsschritt und extrahiert seltene Merkmale. Zeitreihendaten der Länge n werden in kleinere Repräsentationen der Länge k überführt, mit $k < n$ und $n \bmod k = 0$. Die Fensterüberlappung der Länge k modelliert, aufgrund redundanter Informationen, längerfristige Beziehungen in den Ursprungsdaten [127].

Da unterschiedliche Wertebereiche die Datenanalyse beeinträchtigen können, werden sämtliche Datenpunkte auf festgelegte Intervall-

le, wie z. B. $[0, 1]$, $[-1, 1]$ oder $[-2, 2]$ normiert [83]. Die (Min-Max-)Normalisierungsmethode verschiebt und skaliert die Werte im Intervall $[0, 1]$. Die Standardisierungsmethode skaliert die Daten auf den Mittelwert $\mu = 0$ und die Standardabweichung $\Sigma = 1$ (Einheitsvarianz). Abbildung 3.14 veranschaulicht, dass die $TanH(x)$ -Aktivierungsfunktion, die typischerweise in RNN-Zellkonzepten (siehe Abschnitt 3.7.4) verwendet wird, für Eingabewerte über $+2$ und unter -2 , quasi konstante Prädiktionsergebnisse (siehe Abflachung des Gradienten unter -2 und über $+2$ in Abbildung 3.14 unten) liefert und zu verschwindenden Gradienten führt, siehe 3.7. Daher ist es empfehlenswert die Eingabedaten für RNN-basierte Analysemethoden auf einen Bereich von ± 2 zu skalieren.

3.5 Modell-getriebenen Verfahren

In diesem Abschnitt werden die Grundlagen der Modell-getriebenen Verfahren eingeführt. Abschnitt 3.5.1 beschreibt das Kalman-Filter. Und Abschnitt 3.5.2 beschreibt das erweiterte Kalman-Filter.

3.5.1 Kalman-Filter

Im Rahmen dieser Arbeit wird eine Trajektorie als eine Reihe von Positionen p_0, p_1, \dots, p_n definiert, die zeitdiskret abgetastet werden. Jeder Wert p_t ist ein 2D-Vektor, der die Position x - und y zum Zeitpunkt t darstellt. Die Trajektorienvorhersage ist definiert als Vorhersage des Wertes von p_t bei p_{t-1} . Zusätzlich zu p können andere exogene Variablen wie Distanz, Geschwindigkeit und Beschleunigung die Berechnung stützen. Ein optimaler deterministischer Algorithmus zur Vorhersage der Trajektorie wird als Funktion f definiert:

$$p_t = f(p_0, p_1, \dots, p_n). \quad (3.20)$$

In der höchst nichtdeterministischen realen Welt kann diese optimale Funktion (Gleichung 3.20) nur angenähert werden. Daher entwickelt Rudolf E. Kálmán eine rekursive Lösung für das Problem der linearen Filterung diskreter Daten [222]. Das KF wird zur Navigation von Fußgängern, Fahrzeugen, Schiffen und Flugzeugen beforscht und eingesetzt [469]. KF schätzen vergangene, aktuelle und zukünftige Zustände, auch wenn die genaue Systemmodellierung unbekannt ist. Im Gegensatz zu Filtern mit einer unendlichen Impulsantwort (engl. Infinite

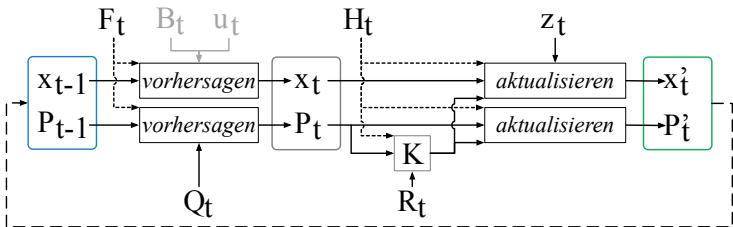


Abbildung 3.15: Detaillierte Kalman-Filter-Verarbeitungskette über die Zeitschritte $t - 1$ bis t . Iterativer Prozess: Das Filter sagt aus dem aktuellen Zustand x_{t-1} und dessen Unsicherheit P_{t-1} mit der Zustandsübergangsfunktion F_t und deren Prozessunsicherheit Q_t den nächsten Zustand x_t mit der Unsicherheit P_t vorher. Optional steuern exogene Effekte der Umgebung mit der Steuermatrix B_t und der Steuergröße u die Vorhersage. Das Filter misst und überprüft mit der Messmatrix H_t und deren Messunsicherheit R_t , ob die Vorhersage des Systemzustands zu den neuen Messwerten passt, korrigiert mit dem Kalman Gewinn K und aktualisiert die aktuellen Messungen z_t entsprechend, um den optimalen Zustand x'_t und dessen Unsicherheit P'_t zu liefern. [222]

Impulse Response, IIR) wie dem gleitenden Mittelwert Filter und Filtern mit einer endlichen Impulsantwort (engl. Finite Impulse Response, FIR) wie den Exponential- und Butterworth-Tiefpass-Filttern, basiert das KF auf einer Zustandsraummodellierung, bei der explizit zwischen der Dynamik des Systemzustands und dem Prozess seiner Messung unterschieden wird. Dadurch lassen sich zum einen anhand von vergangenen Daten Fehler in realen Messdaten korrigieren und in einem nächsten Schritt zukünftige Vorhersagen treffen. Voraussetzung hierfür ist, dass sich die Systemzustände durch ein mathematisches Modell beschreiben lassen [469].

Das KF adressiert das allgemeine Problem, einen Zustand $x \in \Re^n$ eines diskreten zeitgesteuerten Prozesses zu schätzen, der durch eine lineare stochastische Differenzgleichung (siehe Gleichung 3.21) gesteuert wird [469]. Die Menge aller Zustandsvektoren x_t bilden einen speziellen stochastischen Prozess, die Markov-Kette, ab [469], sodass der Zustand zu einem Zeitpunkt t nur vom unmittelbaren Vorgänger $t - 1$ abhängt [222]. Abbildung 3.15 zeigt die detaillierte Verarbeitungskette eines KF. Die Informationen fließen von links nach rechts und wieder zurück. Dabei werden sequentiell die beiden Arbeitsschritte *Vorhersage* und *Korrektur*, bzw. *Aktualisierung* durchgeführt.

Vorhersage

Die Übergangsfunktion F_t (in der Literatur häufig als Dynamik-, Vorhersage- und Übergangsmatrix oder Bewegungsmodell beschrieben) verwendet kinematisch Formeln (siehe Abschnitt 3.2.1), z. B. das Modell der konstanten Geschwindigkeit, um den nächsten Zustand x_t und dessen Unsicherheit P_t aus dem aktuellen Zustand x_{t-1} und dessen Unsicherheit P_{t-1} vorherzusagen. [375] Wobei x_t der (geschätzte) Zustand im Zeitschritt t , bspw. mit Position p und Geschwindigkeit v , ist. p und v sind zufällig und Gaußsch verteilt und korreliert, d. h. v beeinflusst p . Diese Korrelation bzw. die Unsicherheit von x_t wird in der Kovarianzmatrix P_t abgebildet. Optional (graue Pfeile und Text in Abbildung 2.7) beeinflussen und korrigieren exogene Informationen, z. B. die Umwelt, die Prognose, die eine Steuermatrix B und eine Steuergröße u beschreiben. Die Kovarianz P_t wird durch die Prozessrauschkovarianzmatrix Q_t vorhergesagt. x_{t-1} wird in eine neue Gaußsche Verteilung mit der Prozessunsicherheit Q_t verschoben, d. h. die Zustände, die nicht verfolgt werden, werden als Rauschen mit der Kovarianz Q_t beschrieben. Die neue optimale Schätzung x_t ist eine Vorhersage, die aus der vorherigen optimalen Schätzung x_{t-1} plus einer Korrektur aus F und (optional) B_t für bekannte äußere Einflüsse u_t getroffen wurde:

$$\boldsymbol{x}_t = \boldsymbol{F}_t \boldsymbol{x}_{t-1} + \boldsymbol{B}_t \boldsymbol{u}_t. \quad (3.21)$$

Die neue Unsicherheit P_t wird aus der aktuellen P_{t-1} mit F_t und mit zusätzlichen Unsicherheiten Q_t aus der Umgebung vorhergesagt:

$$\boldsymbol{P}_t = \boldsymbol{F}_t \boldsymbol{P}_{t-1} \boldsymbol{F}_t^T + \boldsymbol{Q}_t. \quad (3.22)$$

Aktualisierung

Die Sensoren werden durch die Messmatrix H_t , die Messungen der Sensoren durch z_t und die Messunsicherheit des Sensorrauschen durch die Messrauschkovarianzmatrix R_t , beschrieben. Im nächsten Schritt stimmt der Kalman Gewinn K (engl. Kalman gain) die Annahme über die geschätzten Sensorwerte (geschätzte Messung), die basierend auf dem vorhergesagten Zustand x_t geschätzt werden, mit der Schätzung basierend auf den tatsächlich beobachteten Sensorwerten z_t (tatsächliche Messung) ab, d. h. K korrigiert so, dass die Vorhersage des

Systemzustands zu den neuen Messwerten passt. K wird in jeder Iteration ebenfalls aktualisiert:

$$K = \frac{P_t H_t^T}{H_t P_t H_t^T + R}. \quad (3.23)$$

K wird kleiner, wenn die Messwerte dem vorhergesagten Systemzustand entsprechen. K aktualisiert den aktuellen Zustand x_t mit dem des Messmodells H_t und der Messung z_t und liefert den optimalen Zustand x'_t :

$$x'_t = x_t + K(z_t - H_t x_t), \quad (3.24)$$

wobei $z_t - H_t x_t$ die Innovation w_t des Zustands x_t ist, die aus den Messwerten z_t und der Messmatrix H_t berechnet wird. Der K aktualisiert auch die Unsicherheit P'_t des neuen Zustands:

$$P'_t = P_t - K H_t P_t. \quad (3.25)$$

Zusammen mit P'_t wird x'_t iterativ in einer neuen Runde von Vorhersagen oder Aktualisierungen zurückgemeldet, um zukünftige Zustände x_{t+i} und Unsicherheiten P_{t+i} abzuschätzen. [222] Das KF wird zum Zeitpunkt t_0 mit einer a priori Schätzung $x_0=0$ und Unsicherheit $P_0=0$ initialisiert und gibt für diese Schätzung anschließend eine erste Korrektur aus [469]. Anschließend wird der Prozess mit Hilfe von abwechselnder Vorhersage- und Korrektur-, bzw. Aktualisierungsphasen geschätzt, während dem KF neue Informationen anhand eines neuen, aktuellen Messwerts in der Korrekturphase zugeführt werden [469].

Üblicherweise werden die Parameter des KF manuell oder mithilfe von Algorithmen wie dem Erwartungsmaximierungsalgorithmus [375] optimiert. Im Fall der 2D-Trajektorienvorhersage ist ein optimaler Zustand $x_t = [p_x, p_y, v_x, v_y, a_x, a_y]$ [427]. Das KF modelliert jede Variable dieses Zustands durch eine Gaußsche Verteilung, die durch den Mittelwert μ und die Varianz σ_x gegeben ist.

3.5.2 Erweitertes Kalman-Filter

Häufig ist es in realen Umgebungen der Fall, dass der zu schätzende Prozess bzw. die Messbeziehung zum Prozess über die Zeit nicht konstant bleibt, d. h. nicht linear ist [469]. Ein KF, das solche nichtlinearen Modelle beschreiben kann, wird als EKF bezeichnet [469]. Das EKF linearisiert die aktuelle Schätzung, indem partielle Ableitungen des Prozesses und der Messfunktionen berechnet werden und die Verteilungen

(oder Dichten im kontinuierlichen Fall) der verschiedenen Zufallsvariablen nach ihren jeweiligen nichtlinearen Transformationen nicht mehr als normalverteilt angenommen werden [469]. Damit ist ein EKF ein Zustandsschätzer, der die Optimalität der Bayesschen Regel nur durch Linearisierung annähert [469].

Zu beachten ist, dass die Linearisierung explizit für die Kovarianzmatrizen erforderlich ist. Daher lassen sich EKF nur bedingt bei sehr ausgeprägten, nichtlinearen Bewegungen wie die eines Menschen und im Kontext von Sensoren, wie IMU, einsetzen [127]. RNN leisten vergleichsweise deutlich genauere, messbare Ergebnisse [128, 375]. Kapitel 7 bewertet das EKF und die datengetriebene Methode zur Posenschätzung.

3.6 Künstliche neuronale Netze

Dieser Abschnitt führt die Grundlagen datengetriebener Verfahren ein, insbesondere die künstlichen neuronalen Netze KNN (engl. Artifical Neural Network, ANN). Abschnitt 3.6.1 führt künstliche Neuronen ein. Abschnitt 3.6.2 erörtert einfache KNNs. Abschnitt 3.6.3 beschreibt die wesentlichen Schichten eines KNN. Abschnitt 3.6.4 erläutert renommierte Verlustfunktionen. Abschnitt 3.6.5 diskutiert Optimierungsstrategien. Abschnitt 3.6.6 beschreibt den Umgang mit Daten und Datensätzen und Abschnitt 3.6.7 erläutert die Rastersuche zum Entwerfen eines optimalen KNN.

Traditionelle Computerprogramme sind so konzipiert, dass sie zur Lösung arithmetischer und repetitiver Aufgaben sehr effizient eingesetzt werden können, während sie eine explizite Liste von Recheninstruktionen befolgen. Hingegen sind traditionelle Computerprogramme bei der Lösung von Problemen wie der Vorhersage der menschlichen Bewegung sehr eingeschränkt, zumal die kinematischen Grundlagen und Regularien schwierig zu kodifizieren sind, um individuelle, nichtlineare Bewegungen zu beschreiben oder vorherzusagen [56]. Die meisten modernen und erfolgreichen Algorithmen basieren auf KNN [374]. KNNs sind davon inspiriert, wie Neuronen im Gehirn zusammenarbeiten, um Wissen zu verarbeiten. Im Gegensatz zu handgefertigten Funktionen, die durch Analyse der verfügbaren Daten manuell optimiert werden müssen, lernen KNN eine parametrisierte Funktion, indem sie den (Abbildungs)Fehler zwischen geschätzten und erwarteten Ausgaben in Bezug auf die bereitgestellten

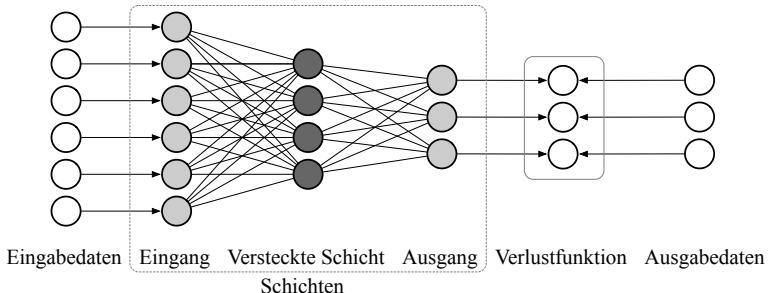


Abbildung 3.16: Darstellung eines einfachen neuronalen Netzes (Informationen fließen von links nach rechts).

Trainingsbeispiele minimieren und so die erforderlichen Funktionen automatisch extrahieren.

Im Allgemeinen besteht ein KNN aus drei wichtigen Komponenten: den Schichten, der Verlustfunktion² und der Optimierungsstrategie. Die Schichten eines KNN sind eine Abstraktion der Funktion, die die Eingabe auf die Ausgabe abbildet. Jede Schicht enthält eine Reihe von Parametern, üblicherweise als Gewichte bezeichnet, die unbekannt sind und auf der Grundlage der Trainingsdaten mit bestimmten Ein- und Ausgängen abgeleitet werden. Die Gewichte w sind Teil der zu optimierenden Parameter θ und werden durch eine Verlustfunktion optimiert, die angibt, wie gut die geschätzten Ausgänge des KNN mit den erwarteten Ausgängen der Trainingsdaten übereinstimmen. Schließlich aktualisiert ein Optimierer schrittweise alle Gewichte in Bezug auf die Minimierung einer gegebenen Verlustfunktion. Abbildung 3.16 zeigt eine Übersicht über ein einfaches KNN.

3.6.1 Neuronen

KNNs orientieren sich an der funktionalen Beschaffenheit des menschlichen Gehirns, das aus etwa 100 Milliarden Nervenzellen (Neuronen) und 100 Billionen synaptischen Verbindungen besteht, welche die Neuronen untereinander verbinden. Im Kern ist das biologische

²Die Verlustfunktion L berechnet den Fehler E eines einzelnen Datenpunktes, bzw. einer Iteration, wohingegen die Kostenfunktion J den Fehler des gesamten Trainingsdatensatzes, bzw. einer Epoche, berechnet.

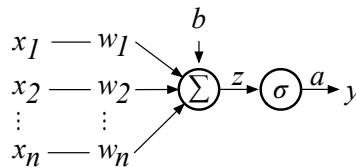


Abbildung 3.17: Schema für ein Neuron in einem künstlichen neuronalen Netz mit Eingangsinformationen x , Gewichten w , Bias b , Funktion z , Aktivierungsfunktion σ und Ausgaben a und y [56]).

Neuron optimiert, Informationen von anderen Neuronen via Dendriten zu empfangen, diese Informationen auf einzigartige Weise zu verarbeiten und das Ergebnis an andere neuronale Zellen weiter zu übertragen. Jede dieser eingehenden Verbindungen wird dynamisch durch spezielle Neurotransmitter gewichtet, d. h. je nachdem wie oft sie benutzt wird, wird die Information verstärkt oder abgeschwächt, so lernt das Gehirn. Diese Gewichtung entscheidet, ob die Eingangsinformationen, die in jedem Zellkörper aufsummiert werden, bei Überschreiten eines Schwellwertes ein neues Signal erzeugt, das sich entlang des Axons der Zelle ausbreitet und an andere Neuronen weitergeleitet wird. [56]

Wird dieses funktionale Verständnis zur neuronalen Informationsübertragung (Neurotransmission) im menschlichen Gehirn in ein künstliches Modell wie in Abbildung 3.17 übersetzt, so nimmt ein künstliches Neuron eine gewisse Anzahl von Eingangsinformationen (engl. inputs) x_1, x_2, \dots, x_n auf, von denen jede mit einem spezifischen Gewicht w_1, w_2, \dots, w_n multipliziert wird. Die Logit-Funktion z summiert diese gewichteten Inputs:

$$z = \sum_{i=0}^n w_i \cdot x_i. \quad (3.26)$$

Das Logit wird dann durch eine Aktivierungsfunktion f , bzw. σ in Abbildung 3.17 geleitet, um die Ausgabe (engl. output) $y = f(z)$ zu erzeugen. Je nachdem, welche Aktivierungsfunktion eingesetzt wird, liegt die Ausgabe in einem Intervall von $[0, 1]$ für die Sigmoid-Aktivierungsfunktion, $[0, +\infty[$ für die Rectifier-(engl. Rectified Linear Unit, ReLU)-Aktivierungsfunktion und $[-1, 1]$ für die Tangens Hyperbolicus (TanH)-Aktivierungsfunktion. Werden die Inputs als Vektor $x = [x_1, x_2, \dots, x_n]$ der Länge n und die Gewichte des Neurons eben-

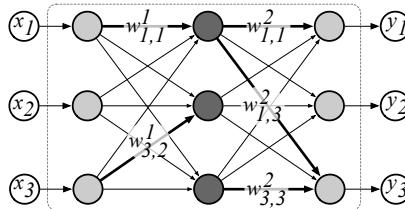


Abbildung 3.18: Einfaches vorwärtsgekoppeltes (engl. Feed-Forward, FF) KNN mit drei Schichten (Box mit gestrichelter Linie: Eingangs-, Verborgene- und Ausgangs-Schicht) und drei Neuronen je Schicht [56].

falls als Vektor $w = [w_1 w_2 \dots w_n]$ der Länge n betrachtet, dann wird die Ausgabe y des Neurons definiert:

$$y = f(x \cdot w + b), \quad (3.27)$$

wobei b der Bias-Term ist. Die Ausgabe y wird berechnet, indem das Skalarprodukt der Eingangs- x und Gewichtsvektoren w bestimmt und der Bias-Term b hinzugefügt wird, um das Logit zu erzeugen, und dann die Aktivierungsfunktion f darauf angewendet wird [56].

3.6.2 Vorwärtsgekoppelte neuronale Netze

Einzelne Neuronen sind nicht annähernd leistungsfähig genug, um komplexe Lernprobleme wie die Klassifikation von Bewegungszuständen oder die Regressions-basierte Schätzung und Vorhersage von menschlichen Posen zu lösen. Für ein einzelnes Neuron ist es unmöglich, verschiedene Bewegungszustände zu unterscheiden. Komplexe Aufgaben werden von vorwärtsgekoppelten KNNS bewältigt [56].

Die Neuronen im menschlichen Gehirn sind in Schichten organisiert. Tatsächlich besteht der menschliche Kortex die Struktur, die für den größten Teil der menschlichen Intelligenz verantwortlich ist aus sechs Schichten. Neuronale Informationen fließen über verschiedene Neuronen hinweg von einer Schicht zur nächsten, bis der sensorische Input in ein konzeptuelles Verständnis umgesetzt wird. Die unterste Schicht des visuellen Kortex bspw. erhält die visuellen Rohdaten der Augen. Diese Informationen werden von jeder Schicht verarbeitet und an die nächste weitergegeben, bis der Mensch in der letzten Schicht feststellt, ob ein Fahrzeug, eine Ampel oder eine Straßenkreuzung betrachtet wird. [56]

3 Grundlagen

Ein in Schichten organisiertes KNN, wie es in Abbildung 3.18 veranschaulicht wird, lässt sich erzeugen, indem zunächst mehrere künstliche Neuronen unterschiedlichen Schichten zugeordnet werden und anschließend sukzessive die Neuronenausgänge der einen Schicht vorwärts gerichtet mit dem Eingang eines oder mehrerer Neuronen der nächsten Schicht verbunden werden. Die mittlere(n) Schicht(en) werden als verborgene Schichten (engl. *hidden layers*) bezeichnet, während $w_{i,j}^k$ das Gewicht der Verbindung zwischen dem i^{ten} Neuron in der k^{ten} Schicht und dem j^{ten} Neuron in der $k+1^{ten}$ Schicht ist. Diese Gewichte w bilden den Parametervektor θ . Analog wie beim Perzepron hängt die Fähigkeit, Probleme mit KNNs zu lösen, davon ab, optimale Werte für die Gewichte w_n zu finden. Ebenso sind die Eingänge x_n , der Bias b sowie die Wahl der Aktivierungsfunktion f entscheidend über die Qualität der Endergebnisse des KNNs. Die Berechnung der Aktivierungen einer vollständigen Schicht ist definiert als [56]:

$$y = \sigma(Wx), \quad (3.28)$$

wobei y der Vektor der Ausgabeaktivierungen ist, x der Vektor der Eingabeaktivierungen ist und W die Gewichtsmatrix der Schicht ist.

Ein mehrschichtiges KNN wird aufgebaut, indem die Ausgangsaktivierungen der Neuronen einer Schicht als Eingang der nächsten verwendet werden. Die verborgenen Schichten ermöglichen es, dem KNN Zwischendarstellungen seiner Eingaben zu berechnen und Berechnungen an ihnen durchzuführen. Auf diese Weise kann das KNN die Berechnungen aller Neuronen effektiv zu einer einzigen, willkürlichen Funktion kombinieren. Hornik et al. [196] zeigen, dass KNNs mit einer einzelnen verborgenen Schicht universelle Funktionsapproximatoren sind, d. h. dass sie jede beliebige Funktion vom Eingang bis zum Ausgang modellieren. In einem vorwärtsgekoppelten KNN verlaufen die Verbindungen nur von einer unteren (linken) Schicht zu einer höheren (rechten) Schicht [56]. Es gibt also keine Verbindungen zwischen Neuronen in der gleichen Schicht und es gibt ebenso keine Verbindungen, die Daten von einer höheren Schicht zu einer niedrigeren Schicht zurückführen. Aus diesem Grund werden diese KNNs als vorwärtsgekoppelte neuronale Netze bezeichnet [56].

3.6.3 Schichten

KNNs sind typischerweise in Schichten organisiert. Die Schichten, die der Eingabe am nächsten liegen, werden als untere Schichten bezeichnet, während diejenigen, die näher an der Ausgabe liegen, als obere Schichten bezeichnet werden. Jede Schicht nimmt eine Reihe von Werten x der vorherigen Schicht und berechnet eine neue Menge transformierter Werte y , die an die nächste Schicht weitergegeben wird. In den folgenden Abschnitten werden diese Vektoren für 1D-Eingaben mit x_i , für 2D-Eingaben mit $x_{i,j}$ und für 3D-Eingaben mit $x_{i,j,k}$ indiziert und schließlich durch $x_{i,j,k,l}$ für 4D-Eingaben. Eine ähnliche Konvention wird für die Gewichte w , aus $w_{i,j,k}$, einer Schicht angewendet und liefert y , aus $y_{i,j,k}$.

Dichte Schichten (engl. dense layers)

Die dichte Schicht (oder vollständig verbundene Schicht) besteht aus einer Reihe von Knoten, die mit allen Knoten der vorherigen Schicht verbunden sind. Die grauen Knoten in Abbildung 3.16 stellen tief verbundene Schichten dar. Ihre Ausgabe ist definiert als:

$$y_i \leftarrow \sigma\left(\sum_{j=1}^n w_{i,j} \cdot x_j + b_i\right) \quad \text{oder kurz} \quad y = \sigma(Wx + b),$$

wobei die Ausgaben der vorherigen Schicht mit der Gewichtsmatrix W multipliziert und dann zur Bias b addiert werden, bevor die Aktivierungsfunktion σ angewendet wird. Die Aktivierungsfunktion kann verwendet werden, um Nichtlinearität einzuführen, da die Matrixvektormultiplikation selbst nur eine lineare Transformation ist. Der Bias kann wiederum verwendet werden, um diese Aktivierungsfunktion zu verschieben. Sowohl die Gewichte als auch der Bias sind Parameter des NNs, die gelernt werden müssen.

Die Aktivierungsfunktion σ ist eine Funktion, die eine skalare Eingabe x auf einen neuen skalaren Wert y abbildet. Typische Auswahlmöglichkeiten sind die logistische Sigmoidfunktion [152], die hyperbolische Tangentenfunktion [152] sowie die gleichgerichtete lineare Einheit [310]. In der Tabelle 3.1 sind die mathematischen Definitionen für diese Funktionen sowie ihre Ausgabebereiche aufgeführt. Eine lineare Aktivierung bedeutet, dass überhaupt keine Aktivierung angewendet wird. Die Anwendung der Aktivierungsfunktion kann auch als separate

3 Grundlagen

Tabelle 3.1: Beschreibung der allgemeinen Aktivierungsfunktionen.

Name und Abkürzung	Definition	Ausgabebereich
Linear	-	$\sigma(x) = x$ $(-\infty, \infty)$
Logistisches Sigmoid	Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$ $(0, 1)$
Hyperbolische Tangente	TanH	$\sigma(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ $(-1, 1)$
Gleichgerichtete Lineareinheit	ReLU	$\sigma(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \leq 0 \end{cases}$ $[0, \infty)$

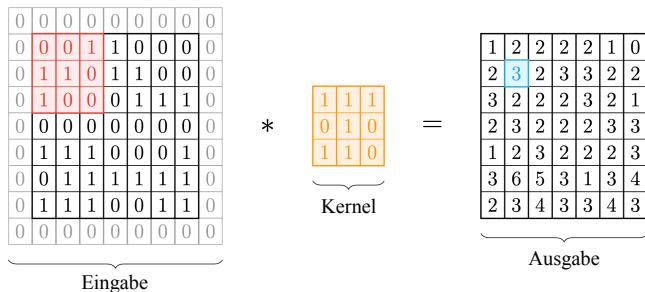


Abbildung 3.19: Darstellung einer zweidimensionalen Faltungsoperation mit Kernelgröße 3×3 und Null-Auffüllung.

Transformationszuordnung $x \mapsto \sigma(x)$ nach einer dichten Schichtoperation ohne Aktivierung betrachtet werden, weshalb sie in der Literatur häufig als zusätzlicher Schichttyp dargestellt wird.

Standardfaltungsschichten

Im Allgemeinen erhöht eine höherdimensionale Eingabe, wie z. B. Bilder, die Anzahl der Gewichte und die Bias dichter Schichten und führt somit schnell zu einem rechnerischen Engpass. Um mit diesen größeren Eingabedimensionen umzugehen, wurden sogenannte Faltungsschichten eingeführt, die vom visuellen Kortex des Menschen inspiriert sind [356]. Diese Schichten nutzen die beiden Konzepte der lokalen Konnektivität und der Gewichtsverteilung. Bei lokaler Konnektivität ist jeder Knoten nur mit einer lokalen Gruppe von Knoten der vorherigen Ausgabeschichten verbunden. Bei der Gewichtsverteilung teilt jede dieser Gruppen auch die gleichen Gewichte, um die Gesamtzahl der zu optimierenden Gewichte weiter zu verringern. Diese

Gewichte werden häufig auch als Kernel bezeichnet.

2D-Faltung. Abbildung 3.19 zeigt eine solche Faltungsoperation für zwei Dimensionen. Die Eingabezellen werden mit den jeweiligen Kernelzellen multipliziert und dann summiert, um den Wert der Ausgabeezelle zu erhalten. Auf die Eingabe kann keine Auffüllung angewendet werden, um dieselbe Ausgabeform beizubehalten. Andernfalls ist die Ausgabeform kleiner als die ursprüngliche Eingabeform. Der Eingang hat auch eine zusätzliche Dimension, den Kanal, mit der mehrere Werte für jede Zelle verarbeitet werden können. Das in Abbildung 3.19 dargestellte Beispiel zeigt der Einfachheit halber nur einen Kanal. Für ein Eingabebild gibt es üblicherweise drei Kanäle, z. B. für rote, grüne und blaue Farbkomponenten. Die Faltung wird dann für jeden dieser Eingangskanäle mit unterschiedlichen Kerneln durchgeführt, und diese Ausgänge werden dann erneut summiert, um den endgültigen Ausgang zu erhalten.

Schritt. Ein weiteres wichtiges Konzept für Faltungen ist der Schritt, der den Abstand zwischen den diskreten Faltungsanwendungen zur Eingabe bestimmt, wie in Abbildung 3.21 dargestellt. Ein Schritt größer als eins verringert die Ausgabegröße, da die Indizes um mehr als eins vorgerückt werden, wenn die Eingabedimension mit dem Faltungskern der neuen Ausgabedimension zugeordnet wird. Dies verringert den Rechenaufwand, beachtet jedoch (wichtige) Teile der Eingabe nicht.

Eingabedimensionen. Diese Faltungsoperationen können auch auf verschiedene Eingabedimensionen erweitert werden. Für die 1D- und 2D-Faltung wird die Ausgabe für einen Knoten wie folgt berechnet:

$$\begin{aligned} \text{1D: } y_i &\leftarrow \sigma \left(\sum_{w=1}^{n_c} \sum_{v=1}^{n_1} w_{v,w} \cdot x_{s_1(i-1)+v,w} \right), \\ \text{2D: } y_{i,j} &\leftarrow \sigma \left(\sum_{w=1}^{n_c} \sum_{u=1}^{n_1} \sum_{v=1}^{n_2} w_{u,v,w} \cdot x_{s_1(i-1)+u, s_2(j-1)+v,w} \right), \end{aligned}$$

wobei σ die Aktivierungsfunktion ist, n_1 , n_2 und n_3 die Kernelgrößen, n_i die Anzahl der Eingangskanäle und s_1 , s_2 und s_3 die verschiedenen Schritte. Die meisten renommierten Architekturen verwenden eine Kernelgröße von 3 in allen Dimensionen, da dies erfahrungsgemäß zu den besten Ergebnissen führt [356].

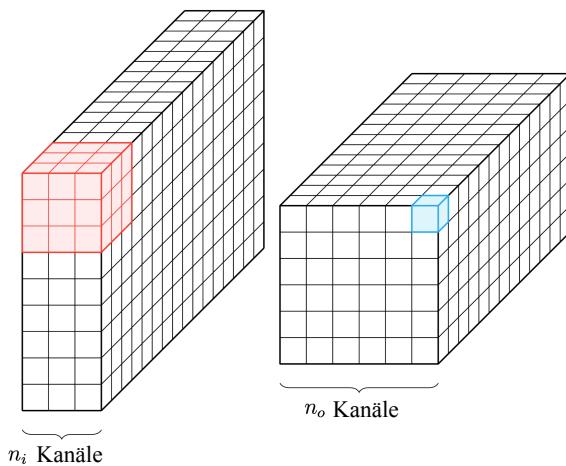


Abbildung 3.20: Darstellung einer zweidimensionalen Standardfaltung für $n_i = 3$ Eingangskanäle und $n_o = 6$ Ausgangskanäle.

Kernel. Jede Faltungsschicht besteht aus n_o Kernen derselben Größe, und die Ausgaben der verschiedenen Kernel sind entlang der neuen Kanaldimension ausgerichtet. Dies ist für eine zweidimensionale Faltung in Abbildung 3.20 mit $n_i = 3$ Eingangskanälen und $n_o = 6$ Ausgangskanälen dargestellt, die der Anzahl der Kernel dieser Faltungsschicht entsprechen. Jeder Kernel hat eine Größe von 3×3 für jeden der 3 Eingangskanäle, was zu insgesamt 27 Gewichten pro Kernel führt. Diese 27 Gewichte werden elementweise mit dem rot hervorgehobenen lokalen Eingabebereich multipliziert und dann summiert, was zu dem in Abbildung 3.20 gezeigten blauen Ausgabewert führt.

Vereinigungs-Schichten (engl. pooling layers)

Um eine Invarianz gegenüber kleinen Verzerrungen innerhalb von Bildern, bzw. Matrizen, zu erreichen und die Anzahl der Gewichte für tiefere Schichten zu verringern, werden Vereinigungs-Schichten verwendet [209]. Diese können als Faltungsschichten interpretiert werden, in denen die Kernel durch Vereinigungs-Regeln ersetzt werden. Um eine sinnvolle Größenreduzierung zu erreichen, muss ein Schritt größer als eins verwendet werden. Anders als bei den Faltungsschichten

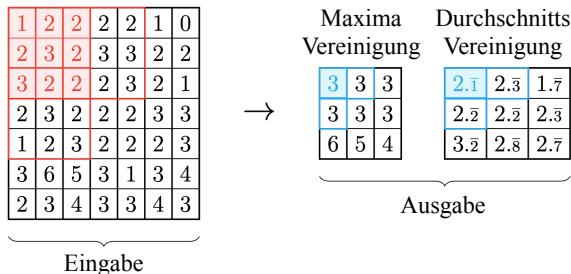


Abbildung 3.21: Darstellung von Maximale- und Durchschnitts-Vereinigungs-Operationen mit Kernelgröße 3×3 und Schritt 2.

wird diese Operation jedoch einmal pro Kanal ausgeführt, sodass die Anzahl der Eingangskanäle der Anzahl der Ausgangskanäle entspricht.

Abbildung 3.21 zeigt sowohl Maxima- als auch Durchschnitts-Vereinigungs-Operationen mit einer Kernelgröße von 3×3 und einem Schritt von 2. Für die Maxima-Vereinigung wird der Maximalwert des lokal verbundenen Bereichs der Eingabe verwendet, während für die Durchschnitts-Vereinigung der Durchschnittswert berechnet wird [51]. Wenn die Dimension des Vereinigungs-Bereichs der Eingabedimension entspricht, ohne Berücksichtigung der Kanäle, wird die Operation normalerweise als globale Maxima- oder Durchschnitts-Vereinigung bezeichnet, da das globale Maximum oder der Durchschnitt jedes Eingangskanals berechnet wird.

Stapel-Normalisierungsschicht

Während der Trainingsphase verschieben sich die Verteilungen der Aktivierungen innerhalb der internen Schichten des KNNs aufgrund von Parameteraktualisierungen. Dies wird als interne Kovariatenverschiebung bezeichnet und verlangsamt das Training, da die Gewichte kontinuierlich an die sich ändernden Verteilungen angepasst werden müssen. Um diese Verschiebung zu verringern und das Training zu beschleunigen, wird eine Normalisierung in Richtung einer Verteilung mit einem Mittelwert von Null ($\mu=0$) und einer Varianz von Eins ($\sigma=1$) auf die Aktivierungen x_i jeder Schicht angewendet, die sogenannte Stapel-Normalisierungsschicht (engl. Batch Normalization layer, BN) [201]. Die Stapel-Normalisierung wird wie das Abstiegsverfahren des Mini-

3 Grundlagen

Stapel-Gradienten (engl. mini batch gradient), der im Abschnitt 3.6.5 erläutert wird für jeden Mini-Stapel durchgeführt:

$$\begin{aligned}
 \mu_{\mathcal{B}} &\leftarrow \frac{1}{m} \sum_{i=1}^m x_i && \text{Mittelwert des Mini-Stapels} \\
 \sigma_{\mathcal{B}}^2 &\leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 && \text{Varianz des Mini-Stapels} \\
 \hat{x}_i &\leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} && \text{Normalisierung} \\
 y_i &\leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) && \text{Skalieren und Verschieben}
 \end{aligned}$$

Die Transformation $\text{BN}_{\gamma, \beta} : x \mapsto y$ ordnet die Aktivierungen $x_{1, \dots, m} \in x$ den Stapel-normalisierten Aktivierungen $y_{1, \dots, m} \in y$ zu. Mit den Parametern γ und β kann das KNN den Maßstab und die Verschiebung dieser Normalisierung anpassen. Dabei erfolgt die Stapel-Normalisierung entlang der Kanaldimension. Ioffe et al. [201] zeigen, dass die Stapel-Normalisierung die Gesamtgenauigkeit von faltenden neuronalen Netzen (engl. Convolutional Neural Networks, CNN) effektiv verbessert und die Anzahl der Trainingsschritte verringert. Santurkar et al. [382] vermuten, dass dies auf die rauscharme Verlustfunktion von Stapel-normalisierten Netzen zurückzuführen ist.

3.6.4 Verlustfunktionen

Die Verlustfunktion ist ein Maß, das bestimmt, wie weit ein aktueller Satz von Vorhersagen \hat{y} von der erwarteten Ausgabe y entfernt ist. Die Abweichung der vorhergesagten Werte von den erwarteten Werten ist umso größer, je größer der Wert der Verlustfunktion ist. Wenn beide übereinstimmen, ist der Verlustwert = 0. Methoden zur Klassifizierung verwendet als Funktion die kategoriale Kreuzentropie, die für n Klassen wie folgt berechnet wird:

$$L(y, \hat{y}) = - \sum_{i=1}^n \hat{y}_i \log(y_i).$$

Da die Ausgaben eine kategoriale Verteilung darstellen, d. h. Wahrscheinlichkeiten für jede Klasse, die gemeinsam eins ergeben, wird eine normalisierende Softmax-Transformation auf die Vorhersagen y ange-

wendet, wobei die transformierte Ausgabe für die Klasse i folgendermaßen berechnet wird:

$$\sigma(y)_i = \frac{e^{y_i}}{\sum_{j=1}^n e^{y_j}}.$$

Typische Fehlerfunktionen

Angenommen, $X = \text{Abtastwerte} \times \text{Zeitschritte} \times \text{Achsen}$ ist das Kreuzprodukt aus Abtastwerten x , Zeitschritten t und Achsen a , dann werden die typischen Fehlerfunktionen zwischen Referenz \hat{y} und Schätzung y wie folgt berechnet:

Mittlerer quadratischer Fehler (engl. Mean Squared Error, MSE):

$$\text{MSE}(y, \hat{y}) = \frac{\sum_{x \in X} (y_{x,t,a} - \hat{y}_{x,t,a})^2}{|X|}, \quad (3.29)$$

mittleres Fehlerquadrat (engl. Root Mean Square Error, RMSE):

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{\sum_{x \in X} (y_{x,t,a} - \hat{y}_{x,t,a})^2}{|X|}}, \quad (3.30)$$

mittlerer absoluter Fehler (engl. Mean Absolute Error, MAE):

$$\text{MAE}(y, \hat{y}) = \frac{\sqrt{\sum_{x \in X} |y_{x,t,a} - \hat{y}_{x,t,a}|}}{|X|}. \quad (3.31)$$

Die Fehlerfunktionen MSE und RMSE betonen Ausreißer stärker als MAE, da jeder einzelne Fehler vor der Summierung quadriert wird.

3.6.5 Optimierungsstrategien

KNNs modellieren eine gewünschte Funktion indem sie Gewichte aus einer Reihe von Beispielen, den Trainingsdaten, ableiten. Ein Optimierer passt die Modelle an die Eigenschaften der Daten an, ohne dass manuelle Anpassungen der Parameter erforderlich sind. Er aktualisiert die Gewichte des KNNs, indem eine Verlustfunktion minimiert wird. Renommierte Algorithmen basieren auf dem Gradientenabstiegsverfahren (engl. Gradient Descent, GD), wobei ein Satz von Parametern θ einer (Ziel)Kostenfunktion $J(\theta)$ mit der Schrittgröße η in die entgegengesetzte Richtung des Gradienten iterativ verarbeitet wird: $\nabla_{\theta} J(\theta)$. Die

3 Grundlagen

Optimierung läuft quasi in Richtung des steilsten Abstiegs, bis ein lokales Minimum erreicht ist. Allerdings garantiert dieses Verfahren nicht, dass ein globales, wirklich optimales Minimum erreicht wird.

Der Gradientenabstieg

Während der Trainingsphase werden einem KNN eine große Anzahl von Trainingsbeispielen, sogenannte Sampels, als Input zugeführt und die Gewichte w iterativ modifiziert, bis die Verlustfunktionen E zwischen der Ausgabe y und dem Trainingsziel \hat{y} (engl. target) zum i^{ten} Trainingsbeispiel minimal ist. Zu den am meisten eingesetzten Verlustfunktionen zählt der quadratische Fehler $E = \frac{1}{2} \sum_i (t^{(i)} - y^{(i)})^2$. [56]

Das Trainingsziel ist, den Parametervektor θ , die Werte für alle Gewichte w in einem KNN, so zu wählen, dass $E = 0$ gilt. Der Fehler ist $E = 0$, wenn das KNN eine vollkommen korrekte Vorhersage macht. Zur Minimierung des Fehlers E zwischen der Ausgabe y und dem Trainingsziel \hat{y} , wird der GD Algorithmus zusammen mit der Methode zur Fehlerrückführung (engl. Backpropagation, BP) eingesetzt.

Die Gewichte eines KNNs werden zu Beginn des Trainings typischerweise zufällig initialisiert und zusammen mit den Trainingssampels und dem Bias b durch das KNN vorwärts propagiert. Am Ende der letzten Schicht des KNNs wird anhand der Verlustfunktionen E der Unterschied zwischen den Trainingsausgaben y und dem Trainingsziel \hat{y} ausgewertet. E wird anschließend zur ersten Schicht des Netzes zurück propagiert (engl. backpropagation) und dort eine Aktualisierung der Gewichte w zur Minimierung des Fehlerwertes E durchgeführt. Anschließend werden die aktualisierten Gewichte zusammen mit den Trainingssampels wieder durch das KNN zur letzten Schicht vorwärts propagiert und dort erneut die Verlustfunktionen, der Fehler E , ausgewertet. Dieser Prozess wird so lange wiederholt, bis die Verlustfunktionen E im Optimalfall den Wert 0 erreicht oder annähert. [56]

Lernrate und -planer

Zusätzlich zu den Gewichtsparametern w , die in einem KNN definiert sind, benötigen die Lernalgorithmen noch eine Reihe weiterer Hyperparameter, um den Trainingsprozess durchzuführen. Einer dieser so genannten Hyperparameter ist die Lernrate η (engl. learning rate). Die Lernrate η gibt an, wie groß die Schritte in Richtung $E = 0$ sind, d. h. wie stark die Gewichte verändert werden.

Mit größeren Lernraten, d. h. größeren Schritten in Richtung $E = 0$, lässt sich der Trainingsprozess beschleunigen, jedoch führt eine große Lernrate η , kurz vor Erreichen des absoluten Fehlerminimums $E = 0$, zu divergierenden Effekten, d. h. dass sich das Fehlerminimum aufgrund zu groß gewählter Lernraten η nie erreichen lässt. Wird hingegen eine zu kleine Lernrate η gewählt, so dauert der Trainingsprozess unendlich lange. Jedoch ist mit einer kleineren Lernrate η die Wahrscheinlichkeit sehr hoch, das Fehlerminimum $E = 0$ zu erreichen.

Mit jeder Gewichtskonfiguration w und Neuberechnung des momentanen Fehlers E verändert sich der Gradientenverlauf im 3D-Raum. Die maximale Entfernung zum Fehlerminimum $E = 0$ stellt eine sehr steile Kontur und die minimale Entfernung zum Fehlerminimum $E = 0$ eine sehr flache Kontur dar. So wird der Gradient mit jeder Gewichtsveränderung w neu bewertet. Der Gradient wird durch die partielle Ableitung der Fehlerfunktion E in Bezug auf jedes der Gewichte w dargestellt [56]:

$$\Delta w_k = -\eta \frac{\partial E}{\partial w_k} \quad (3.32)$$

$$= -\eta \frac{\partial E}{\partial w_k} \left(\frac{1}{2} \sum_i (t^{(i)} - y^{(i)})^2 \right) \quad (3.33)$$

$$= \sum_i \eta (t^{(i)} - y^{(i)}) \frac{\partial y_i}{\partial w_k} \quad (3.34)$$

$$= \sum_i \eta x_k^{(i)} (t^{(i)} - y^{(i)}). \quad (3.35)$$

Wird diese Methode bei jeder Iteration, d. h. mit jeder Veränderung der Gewichte w , angewendet und die Lernrate η nur statisch angepasst, d. h. einmalig zu Beginn des Trainings festgelegt, so ist der Trainingsprozess entweder sehr ineffizient, bei sehr klein gewähltem η , oder sehr ineffektiv, bei hinreichend groß gewähltem η [56].

Um den Trainingsprozess gleichermaßen effizient und effektiv zu gestalten, wird die Lernrate η dynamisch angepasst, d. h. mit Beginn des Trainings werden größere Anpassungen der Lernrate η und im späteren Verlauf kleinere Wertanpassungen der Lernrate η vorgenommen, um das Fehlerminimum $E = 0$ effizient und effektiv zu erreichen. Alternative Optimierungsalgorithmen, darunter der Adam Algorithmus, ermöglichen eine dynamische Anpassung der Lernrate η zur Trainingszeit. Zusammen mit sogenannten Lernratenplanern wird die Lernrate η

3 Grundlagen

zu Beginn des Trainings groß gewählt und im Trainingsverlauf iterativ reduziert, um das Fehlerminimum $E = 0$ anzunähern. [235]

Der stochastische Gradientenabstieg

Der stochastische Gradientenabstieg (SGD) wurde entwickelt, um die Gradientenabstiegmethode für große Datenmengen mit kurzen Trainingszeiten zu ermöglichen. Der SGD-Lernprozess optimiert die Gewichte wie GD basierend auf einer Reihe von Eingaben und entsprechenden gewünschten Ausgaben. Um dies zu ermöglichen, verarbeitet SGD im Gegensatz zu GD nicht alle Trainingsdaten in einer Epoche, sondern nur eine Stichprobe der Trainingsdaten. Für eine Eingabe x berechnet SGD die Ausgabe $y = f(x)$, wobei f die vom Modell beschriebene Funktion ist. Die Ausgabe y wird mit der gewünschten Ausgabe \hat{y} , dem Ziel, verglichen und der Fehler oder Verlust als $L = \text{loss}(y, \hat{y})$ berechnet. Die Verlustfunktion L wird passend zum Ziel gewählt und durch BP [372] die partielle Ableitung von L in Bezug auf ein bestimmtes Gewicht, $\frac{\partial L}{\partial w}$, angenähert und entlang der Lernrate η aktualisiert:

$$w_{\text{neu}} = w_{\text{alt}} - \eta \frac{\partial L}{\partial w} \quad (3.36)$$

Ausgehend von der letzten Schicht berechnet BP anhand der Kettenregel $\frac{\partial L}{\partial w}$ für jedes Gewicht. Dann wird $\frac{\partial L}{\partial x_l}$, mit der Eingabe der Schicht x_l , berechnet. Dieser Wert wird dann iterativ durch die Schichten zurückgegeben, bis für alle Gewichte $\frac{\partial L}{\partial w}$ gefunden wird.

Der Mini-Stapel stochastische Gradientenabstieg

Beim Training mit GD werden normalerweise alle Trainingsmuster in die Berechnung der Fehlerfunktion E einbezogen. Insbesondere bei großen Datenmengen führt dies zu einer langsamen und langwierigen Aktualisierung der Gewichte w und einer Neuberechnung des Gradienten zur Bewertung der Fehlerfunktion E . Im Gegensatz dazu verwendet SGD nur einzelne Trainingsmuster und konvergiert mit einer sehr hohen Varianz zwischen den Epochen. Um die Zielfunktion für das KNN zu minimieren, muss der Verlust für die aktuellen Gewichte über den gesamten Trainingssatz berechnet werden. Da dies viele Berechnungen für eine einzelne Aktualisierung durch SGD erfordert wird der Mini-Stapel-Gradientenabstieg verwendet. Hier wird der Gradient nur für einen kleinen Mini-Stapel von Beispielen berechnet und

Algorithmus 1: Trainingsprozess für neuronale Netze.

Parameter: Trainingsdaten $d = \{(x_i, \hat{y}_i) \mid i \in \{1, \dots, s_d\}\}$,
 Stapelgröße (engl. batch size) s_b , Anzahl der
 Epochen n_e

```

  Stapel  $\leftarrow \{(x_{j\dots j+s_b}, y_{j\dots j+s_b}) \mid j \in \{1, s_b, 2s_b, \dots, s_d\}\}$ 
  for Epoche  $\leftarrow 0$  to  $n_e$  do
    Stapel  $\leftarrow$  Mische (engl. shuffle) Stapel
    foreach  $(x, y)$  in Stapel do
       $y \leftarrow$  Ausgabe berechnen für  $x$ 
       $J(\theta_{t-1}) \leftarrow L(y, \hat{y})$ 
       $\theta_t \leftarrow$  Parameteraktualisierung mit  $\theta_{t-1}$  und  $J(\theta_{t-1})$ 
    end
  end

```

dann zur Durchführung einer Aktualisierung der Parameter verwendet.

Algorithmus 1 kodiert den Trainingsprozess und führt die Konzepte von Epochen und Stapeln (engl. batches) ein. Um den Nachteilen von GD und SGD entgegenzuwirken, wird der gesamte Trainingsdatensatz in kleinere Einheiten, sogenannte Mini-Batches, aufgeteilt und jeder Mini-Batch in Epochen trainiert. Die Gewichte w werden nach jeder Epoche aktualisiert und das Training effektiver gestaltet, um den minimalen Fehler $E = 0$ effizienter zu erreichen. [369]

Die Trainingsdaten werden zunächst in Stapel mit der Größe s_b aufgeteilt. Während einer Epoche führt das KNN eine Aktualisierung für jede Probe in den Trainingsdaten durch. Dies wird n_e Mal durchgeführt, obwohl ein vorzeitiges Stoppen implementiert werden kann, wodurch die Schleife unterbrochen wird, wenn für eine festgelegte Anzahl von Epochen keine Verbesserung der Validierungsgenauigkeit erkennbar ist. Zu Beginn jeder Epoche werden die Stapel (optional) zufällig gemischt, um den Auswirkungen der stochastischen Verlustnäherung entgegenzuwirken. Schließlich wird für jeden Stapel die Verlustfunktion berechnet und ein einzelner Gradientenabstiegschritt durchgeführt, um die Parameter θ zu aktualisieren.

Mini-Stapel-SGD berechnet L und $\frac{\partial L}{\partial W}$ über eine Teilmenge der Trainingsdaten für jede Aktualisierung, da das Aktualisieren des Verlaufs über den gesamten Trainingsdatensatz durch GD häufig zu rechenauf-

wendig ist und die Gewichte durch SGD dazu neigen, in einem lokalen Minimum zu konvergieren. Mini-Stapel-SGD Verarbeitung aktualisiert die Gewichte häufiger und nutzt kleine Datensätze, um lokale Minima zu vermeiden [369]. Um lokalen Minima vorzubeugen, verwenden moderne Methoden wie Adam [235], das Konzept des Impulses, das die Gewichte beständig anpasst [422]. Das Konzept wird häufig mit Lernratenplanern (siehe Abschnitt 3.6.5) kombiniert, die die Lernrate η während des Trainings ändern. Dies ermöglicht vergleichsweise große Aktualisierungen zu Beginn des Trainings und eine spätere Feinabstimmung, um den Verlust zu optimieren.

3.6.6 Datensatz

Das Ziel des Trainings von KNNs besteht darin, einen Satz von Gewichten w für den zugrunde liegenden Trainingsdatensatz zu finden, um die Ausgabe y an das Trainingsziel \hat{y} anzunähern, d. h. den Fehler E zu minimieren. Um einer Überanpassung (engl. overfitting)³ vorzubeugen, wird der zugrunde liegende Datensatz in Trainings-, Validierungs- und Testdatensätze aufgeteilt.

Die Validierungs- und Testdatensätze enthalten typischerweise vollständig unbekannte und vom Trainingsdatensatz unabhängige Datenpunkte, bspw. durch Daten von Testpersonen, die im Trainingsprozess zu keinem Zeitpunkt verwendet werden. Der Trainingsprozess ist in Epochen organisiert. Eine Epoche ist eine einzelne Iteration über den gesamten Trainingsdatensatz. Wenn ein Trainingsdatensatz der Größe d vorliegt und in Mini-Stapel aufgeteilt wird, dann entspricht eine Epoche $\frac{d}{b}$ Modellaktualisierungen. Am Ende jeder Epoche wird anhand des Validierungsdatensatzes und der Fehlerfunktion gemessen, wie gut das Modell auf neue, noch ungesehene (Validierungs-)Daten verallgemeinert. Wenn die Genauigkeit des Trainingsdatensatzes weiter zunimmt, während die Genauigkeit des Validierungsdatensatzes gleich bleibt (oder abnimmt), wird der Trainingsprozess beendet, da das Modell beginnt sich übermäßig auf die Trainingsdaten anzupassen.

Nach Beendigung des Trainings wird mit dem Testdatensatz evaluiert, wie gut die beste Modellkonfiguration auf diesen unbekannten, abstrahierten Testdaten generalisiert. Die Genauigkeit des KKNs auf

³Überanpassung eines KNN führt u. a. dazu, dass das KNN mit bekannten Daten, wie dem Trainingsdatensatz eine sehr gute Leistung erzielt, jedoch eine sehr schlechte mit unbekannten Daten [173].

den Testdaten ist dann ein Indikator für die Fähigkeit und Qualität der Verallgemeinerung des KNNs auf neue, unbekannte Datenpunkte. [56]

3.6.7 Rastersuche

Um die optimale Architektur und die optimalen Parameter eines KNN für einen Datensatz zu finden, wird die Hyperparameter-Optimierung oder Rastersuche (engl. grid search) durchgeführt. Der Modellfehler auf dem Validierungsdatensatz ist das Maß für die Genauigkeit der Rastersuche. Die Rastersuche findet die optimale Konfiguration von Hyperparametern, z. B. die optimale Lernrate, Mini-Stapel-Größe und Sequenzlänge. Für jeden dieser Hyperparameter wird ein Wert aus einer endlichen Menge von Optionen ausgewählt, z. B. $\eta \in \{0.001, 0.01, 0.1\}$ und $batch\ size \in \{16, 32, 64, 128\}$. Das KNN wird mit jeder möglichen Permutation der vorgegebenen Hyperparameter-Auswahl trainiert und die optimale Modellkonfiguration mit der höchsten Genauigkeit und Zuverlässigkeit ausgewählt. [56]

3.7 Sequenzbasierte Lernverfahren

In diesem Abschnitt werden Grundlagen der sequenzbasierten Lernverfahren eingeführt. Abschnitt 3.7.1 erläutert die Herausforderungen der Zeitreihenanalyse und motiviert den Einsatz datengetriebener Verfahren. Abschnitt 3.7.2 führt die RNNs ein. Abschnitt 3.7.3 beschreibt die spezielle Optimierungsstrategie für RNN. In Abschnitt 3.7.4 werden die renommiertesten Zellkonzepte der RNN-Familie erläutert. Und in Abschnitt 3.7.5 werden verschiedene bekannte Architekturkonzepte erläutert, die die Zellen unterschiedlich verbinden.

Die Zeitserienanalyse, als Spezialform der Regressionsanalyse, umfasst Methoden zur inferenzstatistischen Analyse von Zeitreihendaten, die von Sensoren oder anderen Quellen erzeugt wurden, um statistische Merkmale und weitere abstrahierte Charakteristiken wie zukünftige Entwicklungen vorherzusagen. Die Annahme ist, dass die aktuellen Werte einer oder mehrerer *abhängiger* Zeitreihen den aktuellen Wert einer anderen Zeitreihe beeinflussen [404]. Da Zeitreihendaten eine natürliche, zeitliche Ordnung haben, gehen stochastische Regressionsmodelle davon aus, dass zeitlich beieinanderliegende Beobachtungen enger miteinander verbunden sind als weiter auseinanderliegende Beobachtungen. Außerdem gehen stochastische

Regressionsmodelle davon aus, dass Beobachtungen eines bestimmten Zeitraums von vergangenen abhängen und sich zukünftige Werte nur anhand vergangener Beobachtungen vorhersagen lassen [404].

Während klassische Laufzeit-basierte Verfahren, wie z. B. die Koppeleingabe [223], das KF [34] und Regressionsmodelle einsetzen, um bspw. die Objektpose während der Laufzeit zu schätzen, bilden vorwärtsgekoppelte neuronale Netze (engl. Feed Forward Neural Networks, FFNNs) die Eingabedaten auf kategorische Ausgabedaten ab, um weitere Merkmale aus den Eingabedaten zu extrahieren bzw. die Eingabedaten auf kategorische Ausgabedaten abzubilden [507]. RNN bilden intern ebenfalls ein Regressionsmodell ab und können zusammen mit FFNN eingesetzt werden, um die zeitgleiche Glättung von mit Sensorsrauschen behafteten Eingabedaten zu korrigieren, sowie zukünftige Ereignisse vorherzusagen [523].

3.7.1 Herausforderungen der Zeitreihenanalyse

FFNN können Zeitreihendaten verarbeiten, wenn die Zeitdimension abgeflacht wird und eine Sequenz die vollständige Zeitreihe darstellt. Dies hat jedoch erhebliche Nachteile: FFNN ist für Sequenzen beliebiger Länge nicht anpassbar [109]. Da das Modell nur mit vollständigen Sequenzen arbeitet, muss immer der gesamte relevante Verlauf gespeichert werden. Außerdem sind FFNN nicht recheneffizient, wenn die Sequenz ein rollendes Fenster ist oder wenn zu jedem Zeitschritt eine Vorhersage erforderlich ist, da das Netz dann die gesamte Eingabesequenz zu jedem Zeitschritt verarbeiten muss [109]. Da KNN kein inhärentes Zeitkonzept haben, liefern zeitversetzte Eingabesequenzen völlig unterschiedliche Ergebnisse [109]. Eine Klassifizierung von Objekten in Filmsequenzen mit KNNs unmöglich, da eine um ein einzelnes Bild verschobene Videosequenz nicht mit derselben Ausgabe wie das Original klassifiziert werden kann. Eine Vielzahl von KNN adressieren diese Probleme. Dazu gehören zeitverzögerte neuronale Netze (engl. Time Delay Neural Networks, TDNNs) [450], zeitliche faltende neuronale Netze (engl. Temporal Convolutional Networks, TCNs) [396], aufmerksamkeitsbasierte Methoden wie die Transformer-Architektur [443] und die Familie der rekurrenten neuronalen Netze (engl. Recurrent Neural Networks, RNNs) [109, 191].

Während heute, 2021, Transfomer-basierte Netze die genauesten Ergebnisse bei der Sprachverarbeitung erzielen [443], benötigen sie eine spezielle Codierung der globalen Ortsinformation jeder einzelnen eindeutigen Information in allen Eingabesequenzen, bspw. wo ein beliebiges Wort in einem Satz in einem Buch angeordnet ist. Diese komplexe Struktur ist nicht intuitiv auf das Problem der Trajektorienvorhersage anwendbar. Im Gegensatz dazu lösen TDNNs, vorgestellt von Waibel et al. [450], die Probleme regulärer KNNs, indem sie 1D-Faltungen über die Zeitdimension verwenden. TCNs formulieren die Netzstruktur der TDNNs neu [396]. Beide verwenden Neuronen, die zu einem aktuellen Zeitschritt und mit einer Reihe von Verzögerungen Verbindungen zu Eingangsvariablen haben. Diese Verbindungen werden mit den gleichen Gewichten für jeden Zeitschritt berechnet. Ähnlich wie klassische CNN verwenden sie Kernel, um Abstraktionen für die Eingabezeitreihen zu berechnen. Diese bilden eine hierarchische Darstellung der Zeitreihen auf mehreren Ebenen. TCNs können somit parallel optimiert werden, sind wesentlich ressourceneffizienter als RNNs und ermöglichen ein effizientes Training für sehr große Datenmengen, wenn die Zeitreihen kurz sind und eine feste, vordefinierte Länge haben. Während der Inferenz erfordert die Vorrückkopplungsstruktur der TCNs jedoch teure Berechnungen des gesamten Verlaufs der Zeitreihen. Mit zunehmender Sequenzlänge nimmt die Breite und Tiefe der Architektur zu. Die Sequenzlänge kann nicht geändert werden. Das macht es unmöglich, die Berechnungen bspw. auf eingebetteter Hardware auszuführen, wenn kurze Sequenzlängen von der Hardware vordefiniert werden. Stattdessen komprimiert der von RNNs verfolgte Ansatz den Verlauf in eine Darstellung mit einer festen Größe, die bei jedem Zeitschritt nur eine minimale Berechnung erfordert. Daher werden in dieser Dissertation TCNs evaluiert, allerdings liegt der Schwerpunkt auf der Anpassung und Entwicklung von Methoden der RNN-Familie.

3.7.2 Rekurrente neuronale Netze

Rekurrente neuronale Netze (RNN) wurden 1990 von Elman et al. [109] vorgeschlagen und unterscheiden sich von den in Kapitel 3.6.2 beschriebenen FFNN, da sie eine spezielle Art von neuronaler Schicht, die als rekurrente Schicht bezeichnet wird, nutzen, die es dem Netz ermöglicht, den Zustand (engl. state) zwischen den einzelnen Schichten des Netzes aufrechtzuerhalten. Diese rekurrente Schicht ist ein verborgener Zustandsvektor (engl. hidden state vector) und

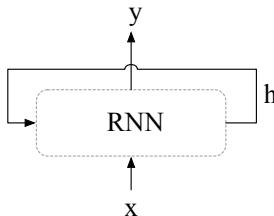


Abbildung 3.22: Vereinfachte Darstellung der Grundstruktur eines rekurrenten neuronalen Netzes (RNN). RNNs sind eine angepasste Form von vorwärtsgekoppelten NNs, die einen verborgenen Zustandsvektor h , eine Art Gedächtnis, annehmen, um Informationen x über den Verlauf und den aktuellen Kontext zu speichern. h wird in jedem Zeitschritt berechnet und dann als Eingabe für den nächsten Zeitschritt weitergeleitet. Das Netz verwendet h , um relevante Merkmale aus dem Eingabeverlauf zu komprimieren. Dies ermöglicht eine effiziente Berechnung der Ausgabe y , die nur durch die Größe von h begrenzt ist. Die Box mit der grauen gestrichelten Linie ist durch eine oder viele Zellen (vgl. Box mit grauer gestrichelter Linie in der Abbildung 3.25) zu ersetzen.

ein Gedächtnis, um Informationen über die Vergangenheit und den aktuellen Kontext zu speichern. Dieser verborgene Zustand wird vom Netz in jedem Zeitschritt berechnet und dann als Input für den nächsten Zeitschritt zurückgeführt. Das Netz nutzt den verborgenen Zustand, um die relevanten Merkmale aus der Eingabehistorie in eine kompaktere Darstellung zu überführen. Dies ermöglicht eine effiziente Berechnung der Ausgabe in jedem Zeitschritt t und eine Darstellung, die nur durch die Größe des verborgenen Zustandsvektors begrenzt ist. RNN sind natürliche Modelle für Zeitreihen. Die Annahme ist, dass das Modell eine Zustands- und Übergangsfunktion hat, die aus ihrem Vorgänger und einem beliebigen Modelleingang nachfolgende Zustände berechnet. Deshalb eignen sich RNNs für die Verarbeitung von Zeitreihendaten, die bspw. von unterschiedlichen Sensormodalitäten erfasst wurden (siehe Abschnitt 3.4.1). [56]

In einem RNN haben alle Neuronen sowohl eingehende Verbindungen, die von allen Neuronen der vorhergehenden Schicht ausgehen, als auch ausgehende Verbindungen, die zu allen Neuronen der nachfolgenden Schicht führen. Im Gegensatz zu einer FF Schicht gibt es bei RNN auch in der rekurrenten Schicht wiederkehrende Verbindungen, die Informationen zwischen den Neuronen derselben Schicht, einschließlich

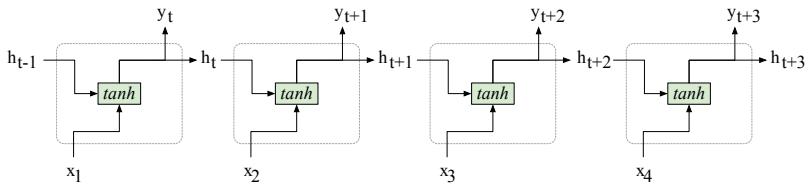


Abbildung 3.23: Vereinfachte Darstellung des Informationsflusses durch ein abgerolltes RNN, gleicht einem vorwärtsgekoppelten Netz. Vorwärtsdurchlauf: Verarbeitung einer Reihe von Eingaben x_1 bis x_4 . Eine vereinfachte Version der Zellen wird für eine bessere Übersicht gezeigt. Abbildung 3.25 zeigt eine vollständig Version derselben Elman-Zelle (vgl. Box mit grauer gestrichelter Linie).

dieselben Neurons einer Schicht, weitergeben. Eine rekurrente Schicht mit r Neuronen hat also insgesamt r^2 rekurrente Verbindungen [56].

Bei RNN lässt sich die Lebensdauer der Netzinstanz in diskrete Zeitschritte aufteilen. Bei jedem Zeitschritt wird das Modell mit dem nächsten Element der Eingabe versorgt. Die vorwärts gekoppelten Verbindungen in einem RNN repräsentieren den Informationsfluss von einem Neuron zum nächsten, wobei die übertragenen Daten die berechnete neuronale Aktivierung des aktuellen Zeitschritts darstellen. Abbildung 3.23 stellt den Informationsfluss rekurrenter Verbindungen dar, bei dem die Daten die gespeicherte neuronale Aktivierung aus dem vorherigen Zeitschritt, d. h. von links und von unten beeinflussen. Somit stellen die Aktivierungen der Neuronen in einem RNN den akkumulierenden Zustand der Netzinstanz dar. Die anfänglichen Aktivierungen der Neuronen in der rekurrenten Schicht sind Parameter des Modells. Beim Training von RNN werden die optimalen Gewichte w gesucht, um die Netzausgabe y an das Trainingsziel \hat{y} anzunähern, d. h. den Fehler E zu minimieren. Bei einer festen Lebensdauer, wie bspw. t Zeitschritte, lässt sich eine RNN-Instanz als ein *abgerolltes* unregelmäßig strukturiertes FFNN darstellen. [56]

3.7.3 Optimierungsstrategien

Beim Training von RNN wird der Gradient auf Basis der abgerollten Variante berechnet. Somit kann prinzipiell die Backpropagation-Technik für FFNN auch für das Training von RNN angewendet werden. Problematisch ist allerdings, dass Standard-RNN Informationen aus früheren Zeitpunkten als Input für die nächsten Zeitpunkte verwenden

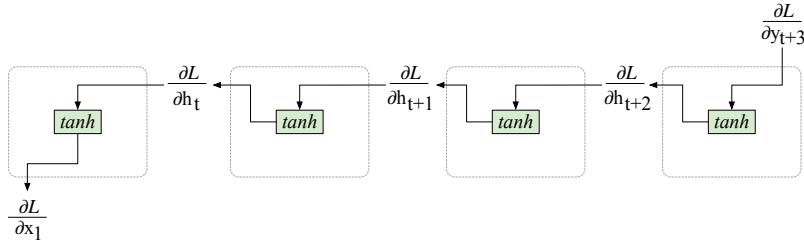


Abbildung 3.24: Vereinfachte Darstellung des Gradientenflusses während der Fehlerrückführung durch die Zeit durch ein abgerolltes RNN (ähnelt einem vorwärts gekoppelten Netz). Eine vereinfachte Version der Zellen wird für eine einfache Übersicht gezeigt. Abbildung 3.25 zeigt eine vollständig Version derselben Elman-Zelle (Box mit grauer gestrichelter Linie).

den und dass die Aktualisierung der rekurrenten Gewichte fordert, dass die Fehlerinformation E durch alle vergangenen und involvierten Zeitschritte zurück propagiert wird. Zur Gradientenberechnung sind deshalb sehr lange Ketten von Gewichten, die miteinander multipliziert werden, involviert, das aufgrund der Multiplikation von mehrfach sehr kleinen bzw. großen Werten schnell zu verschwindenden (engl. vanishing gradient) oder explodierenden (engl. exploding gradient) Gradienten führt (Abschnitt 3.7.3). Der Zeitaufwand zur Aktualisierung der Netzgewichte wächst und resultiert in einer deutlich längeren Trainingsdauer. Unterschiedliche Aktivierungsfunktionen, wie ReLU, TanH oder Sigmoid skalieren den Gradienten und beugen dessen Unter- oder Überschreiten vor. [56]

Um die Gradienten für die Gewichtsoptimierung zu bestimmen, wird die Standard-Backpropagation um Zeitachse erweitert. Dieses Verfahren wird als Fehlerrückführung durch die Zeit (engl. Backpropagation Through Time, BPTT) bezeichnet. Das Netz wird abgewickelt (siehe Abbildung 3.23) und vom letzten Zeitschritt ausgehend werden sowohl $\frac{\partial L}{\partial W}$ als auch $\frac{\partial L}{\partial h_{t-1}}$ mit der Backpropagation abgeleitet. Dieser Wert wird dann durch die Zeit weitergegeben. Abbildung 3.24 veranschaulicht das BPTT Verfahren.

Verschwindende und explodierende Gradienten

Das Anwenden von BPTT auf das Konzept der Elman Zelle von t_0 bis t_n hat jedoch eine gravierende Schwachstelle: die Gradientenaus-

breitung durch die Zelle im Rückwärtsdurchlauf. BPTT berechnet den Gradient der Zielfunktion bezüglich des letzten verborgenen Zustands:

$$\frac{\partial L}{\partial h_{t-1}} = W^T \frac{\partial L}{\partial y_t}. \quad (3.37)$$

Um den Gradienten für eine längerfristige Abhängigkeit zu bestimmen wird BPTT mehrmals durchgeführt. Der Gradient entwickelt sich:

$$\frac{\partial L}{\partial x_0} \approx W^T (W^T (\cdots W^T \frac{\partial L}{\partial y_n})) \approx W^{T^n}. \quad (3.38)$$

Das führt für große n , für lange Abhängigkeiten, entweder zu explodierenden ($w > 1$) oder verschwindenden ($w < 1$) Gradienten. Diese werden als explodierendes oder verschwindendes Gradientenproblem bezeichnet [36]. Beide Probleme verhindern das Lernen von Gewichtsmatrizen für langfristige Abhängigkeiten, da die Gewichtsaktualisierungen auf diesen Gradienteninformationen basieren. Es werden verschiedene Methoden vorgeschlagen, um diese Probleme zu bewältigen [37, 191, 471]. Das Abschneiden von Gradienten [37] begrenzt bspw. die Gradientengröße, vermeidet aber nur numerische Probleme, die durch explodierende Gradienten verursacht werden, lindert jedoch nicht das Problem des verschwindenden Gradienten.

Das 1997 vorgeschlagene Lang-Kurzzeitgedächtnis (engl. Long Short-Term Memory, LSTM) von Hochreiter und Schmidhuber [191] verwendet ein anderes Zellendesign, das zur Berücksichtigung von Langzeit-abhängigkeiten entwickelt wurde und die Probleme der verschwindenden und explodierenden Gradienten der Elman Zelle löst. Wie in Abbildung 3.27 gezeigt, verwendet die LSTM-Zelle Strukturen, sogenannte Gatter (engl. gates), um den Fluss des verborgenen Zustands zu manipulieren und zu steuern. Es werden zwei verborgene Zustände verwendet: ein verborgener Zustand h aus dem Standard RNN und ein Zellenzustand c , der nur durch die Gatter modifiziert wird. Abschnitt 3.7.4 beschreibt LSTM detailliert.

3.7.4 Architekturen rekurrenter Zellen

Die Elman RNN Zelle und die LSTM Zelle sind die Basis aller RNN-Konzepte. RNNs werden in Anwendungen wie der Verarbeitung natürlicher Sprache [423], der Musikmodellierung [42] und der

Trajektorienvorhersage [11, 15] verwendet. Die komplexe Struktur der LSTM Zelle mit vier Gattern und acht unterschiedliche Gewichtsmatrizen löst die Probleme der RNN Zelle führt jedoch zu schwierigen Trainingsverläufen. Deshalb wurden einfachere Zellen erforscht, die mit verschwindenden Gradienten umgehen können [76, 155, 217, 254] und eine kleinere Darstellung mit weniger Parametern verwenden. Die bekanntesten und erfolgreichsten dieser Zelltypen sind die gegatterte rekurrente Einheit (engl. Gated Recurrent Units, GRU) von Cho et al. [76] und das Identitäts rekurrente neuronale Netz (engl. Identity Recurrent Neural Network, IRNN) von Le et al. [254].

Chung et al. [81] vergleichen die RNN-, GRU- und LSTM-Zelldesigns mit der Erzeugung von Musik und Sprache. Sie finden heraus, dass die Gatter der GRU und LSTM Zellen das grundlegende RNN-Zellkonzept übertreffen. Jozefowicz et al. [217] verwenden einen genetischen Algorithmus, um optimale rekurrente Zellkonzepte zu finden. Sie optimieren ihre Modelle für Aufgaben in den Bereichen Arithmetik und Textvorhersage. Sie finden eine Architektur, die sowohl GRU als auch LSTM bei allen Aufgaben außer der Textvorhersage übertrifft. Ihre Struktur lässt sich nicht auf die Lokalisierung und Sensorfusion übertragen. Greff et al. [155] bewerten verschiedene Variationen des LSTM-Zellkonzepts, indem sie unterschiedliche Gatter entfernen. Einige ihrer Zellkonzepte verbessern die Inferenzgeschwindigkeit und liefern akzeptable Ergebnisse. Allerdings beeinflussen die Lernrate, die Netzgröße und die Anzahl der Parameter die Leistung am meisten.

Da keine Variation das LSTM-Zellkonzept übertrifft und es keine öffentlich bekannten verwandten Arbeiten gibt, die verschiedene Zelldesigns der RNN Familie zur Lokalisierung untersuchen, werden im Rahmen dieser Dissertation nur die renommiertesten Zellkonzepte, das Elman RNN-, IRNN-, LSTM- und GRU-Zellkonzept untersucht.

Standard (Vanilla) rekurrentes neuronales Netz

Abbildung 3.25 zeigt das Zellkonzept des Standard-(in der Literatur auch Vanilla-)RNN, das von Elman [109] vorgestellt wurde. In dieser Zelle werden die Eingabe x_t und der verborgene Zustand h mit jeweils einer Gewichtsmatrix W multipliziert, addiert und einer tanh-Aktivierungsfunktion übergeben:

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h). \quad (3.39)$$

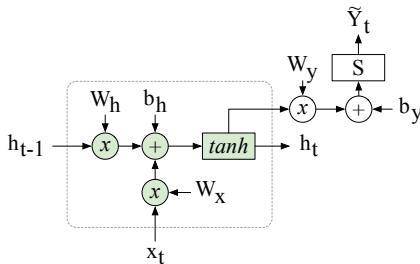


Abbildung 3.25: Detaillierte Ansicht einer Elman-Zelle [109] des RNN während der Vorhersage von \tilde{y}_t aus x_t und h_{t-1} und h_t . Die Softmax Funktion S außerhalb der Box mit der gestrichelten grauen Linie ist nicht Teil der Zelle und kann durch beliebig andere Funktionen ersetzt werden, bspw. durch eine weitere gestapelte Elman-Zellen rechts und oberhalb dieser Zelle, siehe Abschnitt 3.7.5. Durch die Backpropagation des Gradienten durch die Zeit werden die Gewichte W und Bias b optimiert.

Die Ausgabe y_t wird folgendermaßen berechnet:

$$y_t = W_{hy} h_t + b_y. \quad (3.40)$$

Die Eingaben x_t einer Sequenz werden bei jedem Zeitschritt t in die Elman-Zelle eingefügt. Und der verborgene Zustand h_t und die Ausgabe y_t werden berechnet und für den nächsten Schritt $t + 1$ gespeichert. Diese naive Implementierung leidet jedoch unter den Problemen des Verschwindens oder Explodierens des Gradienten [36, 329].

Lang-Kurzzeitgedächtniszelle, LSTM

Um das Verschwinden oder Explodieren des Gradienten der Elman-Zelle zu beheben, wurde 1997 von Hochreiter et al. [192] die Lang-Kurzzeitgedächtniszelle (engl. Long Short-Term Memory, LSTM) vorgeschlagenen. Um sowohl lang- als auch kurzzeitbezogene Abhängigkeiten innerhalb von Zeitreihendaten zu erfassen, enthalten einfache rekurrente Schichten Rückkopplungsschleifen, mit denen zuvor gesehene Eingaben die aktuelle Ausgabe beeinflussen können. Die LSTM-Zellen implementieren komplexere Aktivierungsfunktionen, die sogenannte Gatter (engl. gates [191]). Die Gatter ermöglichen es einem internen Zustand Informationen aus den Daten zu speichern. Jozefowicz et al. [217] formalisieren LSTM folgendermaßen, ohne Gucklochverbindungen und \odot bezeichnet das elementweise Hadamard-Produkt:

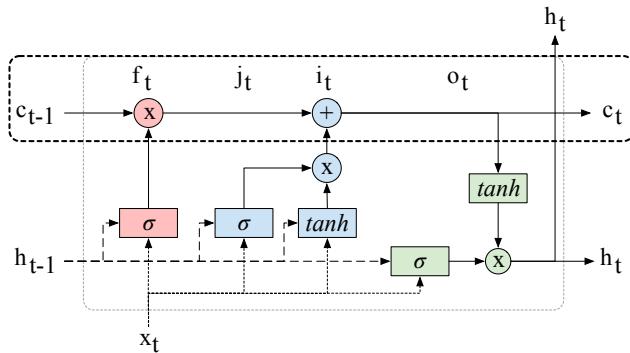


Abbildung 3.26: Eine Lang-Kurzzeitgedächtniszelle (LSTM), vorgeschlagen von Hochreiter und Schmidhuber [191]. LSTM verwendet die Gatter f_t , i_t , j_t und o_t , um den verborgenen h_t und den Zellzustand, das Gedächtnis, c_t selektiv zu ändern. Die Gatter und die Zellenstruktur ermöglichen einen optimierten Gradientenfluss, der es dem Netz ermöglicht, lang- und kurzfristige Abhängigkeiten zu lernen. Gewichte W und Bias b werden nicht angezeigt.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad \text{Vergessensgatter, (3.41)}$$

$$j_t = \sigma(W_{xj}x_t + W_{hj}h_{t-1} + b_j) \quad \text{Aktualisierungsgatter, (3.42)}$$

$$i_t = \tanh(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad \text{Eingangsgatter, (3.43)}$$

$$o_t = \tanh(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad \text{Ausgabegatter, (3.44)}$$

$$c_t = c_{t-1} \odot f_t + i_t \odot j_t \quad \text{Zellzustand, (3.45)}$$

$$h_t = \tanh(c_t) \odot o_t \quad \text{Verborgener Zustand. (3.46)}$$

Zellzustand bzw. Gedächtniszelle

Eine der Kernkomponenten der LSTM-Architektur ist die Gedächtniszelle (Zellzustand bzw. Kontextvektor), ein Tensor, der durch die schwarz umrahmte Box in Abbildung 3.26 dargestellt wird. Die Gedächtniszelle enthält kritische Informationen, die sie im Laufe der Zeit gelernt hat. Die LSTM-Zelle ist so ausgelegt, dass nützliche Informationen in der Gedächtniszelle über viele Zeitschritte hinweg effektiv erhalten bleiben [56]. Bei jedem Zeitschritt modifiziert die LSTM-Zelle die Gedächtniszelle mit neuen Informationen in drei

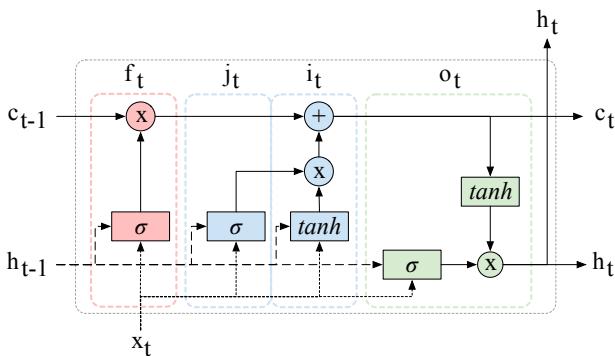


Abbildung 3.27: Detaillierte Darstellung der Architektur einer Lang-Kurzzeitgedächtniszelle (LSTM). Gewichte W und Bias b werden nicht angezeigt.

verschiedenen Phasen, die als rote, blaue und grüne gestrichelte Boxen in Abbildung 3.27 dargestellt sind. Die LSTM-Zellstruktur ermöglicht im Gegensatz zur Elman-Zelle eine Trennung zwischen diesem Zell- und Ausgangszustand. Diese Trennung und die Tatsache, dass der Zellzustand nicht direkt durch eine nichtlineare Aktivierungsfunktion beeinflusst wird, reduzieren die verschwindenden Gradienten.

Abbildung 3.27 stellt den Informationsfluss (von unten links nach oben rechts und von links nach rechts) in einer LSTM-Zelle dar. Es ist ersichtlich, dass sich die LSTM-Zelle (Box mit grauer gestrichelter Linie) aus mehreren Schlüsselkomponenten zusammensetzt und drei Ein- (c_{t-1} , h_{t-1} und x_t) und drei Ausgänge (c_t , h_t oben und h_t rechts) hat [56]. Die LSTM-Zelle besteht aus drei Gattern: das Vergessensgatter f_t (rote gestrichelte Box), das Eingangs- i_t und Aktualisierungsgatter j_t (blaue gestrichelte Box) und das Ausgabegatter o_t (grüne gestrichelte Box). Das Vergessensgatter entfernt bestimmte Informationen aus dem Gedächtnis, dem Zellzustand c_t . Die Eingangs- und Aktualisierungsgatter fügen bestimmte Informationen zum Zellzustand hinzu oder aktualisieren diesen. Und das Ausgabegatter erzeugt eine Ausgangsvorhersage basierend auf der Eingabe und dem aktualisierten Zellzustand (in der Literatur auch Kontextvektor oder Gedächtniszelle genannt). Alle Gatter arbeiten mit der aktuellen Eingabe x_t sowie dem vorherigen verbor-

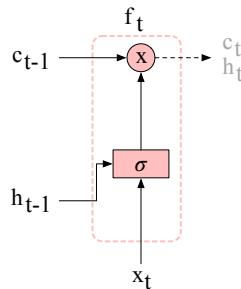


Abbildung 3.28: Architektur des Vergessensgatter einer LSTM-Zelle (die grauen Symbole c_t und h_t und der gestrichelte Pfeil zeigen an, dass diese keine direkte Verbindung mit dem Gatter haben).

genen Zustand \mathbf{h}_{t-1} . Die Matrizen \mathbf{W} und die Bias \mathbf{b} sind die Gewichte der Zelle, die optimiert werden:

$$c_t = c_{t-1} \odot f_t + i_t \odot j_t \quad \text{Zellzustand,} \quad (3.47)$$

$$h_t = \tanh(c_t) \odot o_t \quad \text{Verborgener Zustand.} \quad (3.48)$$

Vergessensgatter

Zuerst muss die LSTM-Zelle bestimmen, wie viel des bisherigen Speichers beibehalten werden soll. Dies wird durch das Vergessensgatter f_t bestimmt (rote gestrichelte Box in Abbildung 3.28).

Die Grundidee des Vergessensgatter f_t ist: Der Zellzustand c_{t-1} aus dem vorherigen Zeitschritt ist reich an Informationen, jedoch sind einige dieser Informationen möglicherweise veraltet und müssen daher gelöscht werden. Um herauszufinden, welche Elemente im Zellzustandstensor c_{t-1} noch relevant sind und welche Elemente für c_t irrelevant sind, wird in f_t versucht einen Bit-Tensor, Tensor bestehend aus Nullen und Einsen, zu berechnen, der mit dem vorherigen Zustand c_t multipliziert wird. Wenn eine bestimmte Stelle im Bit-Tensor eine 1 enthält, bedeutet das, dass die Stelle in der Speicherzelle immer noch relevant ist und beibehalten wird. Wenn diese bestimmte Stelle stattdessen eine 0 enthält, bedeutet das, dass die Stelle in der Speicherzelle nicht mehr relevant für c_t ist und entfernt wird. Dieser Bit-Tensor wird approximiert, indem der Eingang dieses Zeitschritts

x_t und der Ausgang h_{t-1} der LSTM-Zelle des vorherigen Zeitschritts verknüpft werden und eine Sigmoidschicht auf den resultierenden Tensor angewendet wird. Ein sigmoidales Neuron gibt einen Wert aus, der entweder sehr nahe bei 0 oder sehr nahe bei 1 liegt.

Formalisierung: Die Vergessensgatter-Aktivierung f_t wird aus dem alten verborgenen Zustand h_{t-1} und der Eingabe x_t berechnet (Gleichung 3.41). Das Ergebnis wird verwendet, um Elemente des Zellzustandsvektors c_{t-1} selektiv zu löschen, auf Null zu setzen, (Gleichung 3.45). Gers et al. [148] empfehlen für lange Sequenzen das Bias b_f explizit >1 zu setzen, um lange Sequenzen zu ermöglichen. Das Vergessensgatter (engl. forget gate) gibt den Bruchteil des vorherigen Zellzustands c_{t-1} an, der für den neuen Zellzustand beibehalten wird (die Box f_t mit roter gestrichelter Linie in Abbildung 3.27 hebt den Mechanismus des Vergessensgatters hervor).

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad \text{Vergessensgatter.} \quad (3.49)$$

Eingangs- und Aktualisierungsgatter

Sobald herausgefunden wurde, welche Informationen im alten Zustand c_{t-1} beibehalten werden und welche gelöscht werden, lässt sich festlegen, welche Informationen in den Zellzustand c_t geschrieben werden. Dieser Teil der LSTM-Zelle wird als Eingangs- j_t bzw. Aktualisierungsgatter i_t bezeichnet und ist in Abbildung 3.29 in den blauen gestrichelten Boxen dargestellt.

Das Eingangsgatter (engl. input gate) ist in zwei Teile untergliedert: Die erste Komponente i_t beinhaltet die Festlegung, welche Informationen in den Zustand geschrieben werden. Diese wird von der TanH-Schicht berechnet, um einen Zwischentensor zu erzeugen. Die zweite Komponente j_t besteht darin, herauszufinden, welche Komponenten dieses berechneten Tensors tatsächlich in dem neuen Zustand aufgenommen und welche vor dem Schreiben verworfen werden. Dies wird erreicht, indem ein Bitvektor aus 0 und 1 approximiert wird und dabei die gleiche Strategie (eine sigmoidale Schicht) wie beim Vergessensgatter verwendet wird. Der Bitvektor wird mit dem Zwischentensor multipliziert und anschließend das Ergebnis addiert, um den neuen Zellzustand c_t für die LSTM-Zelle zu erzeugen.

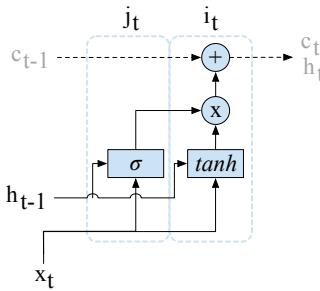


Abbildung 3.29: Architektur der Eingangs- und Aktualisierungsgatter einer LSTM-Zelle (die grauen Symbole c_{t-1} und c_t und h_t und die gestrichelten Pfeile zeigen an, dass diese keine direkte Verbindung mit dem Gatter haben).

Formalisierung: Das Eingangsgatter (engl. input gate) steuert den Bruchteil des vorherigen Zellzustands c_{t-1} über i_t , der mit der neuen Eingabe x_t über j_t und i_t aktualisiert wurde. Das Ergebnis wird zu c_{t-1} über i_t hinzugefügt, um den endgültigen Zellenstatus c_t zu erhalten (die Boxen j_t und i_t mit blauer gestrichelter Linie in Abbildung 3.27). Das Eingangsgatter fügt nicht nur neue Eingaben hinzu, sondern aktualisiert diese auch, deshalb wird es auch das *Aktualisierungsgatter* (engl. update gate) genannt. Es berechnet einen Vorschlag zur Aktualisierung des verborgenen Zustands über i_t und j_t (Gleichungen 3.43 und 3.42). Für den neuen Zellenstatus c_t wird diese Aktualisierung mit dem Zurücksetzen des Zellenstatus f_t in c_t kombiniert (Gleichung 3.45; die Boxen j_t und i_t mit blauer gestrichelter Linie in Abbildung 3.27 heben den Mechanismus des Aktualisierungsgatters hervor):

$$j_t = \sigma(W_{xj}x_t + W_{hj}h_{t-1} + b_j) \quad \text{Aktualisierungsgatter,} \quad (3.50)$$

$$i_t = \tanh(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad \text{Eingangsgatter,} \quad (3.51)$$

Ausgabegatter

Schließlich liefert die LSTM-Zelle in jedem Zeitschritt eine Ausgabe h_t . Während der verborgene Zustandsvektor h_t direkt ausgegeben wird kommuniziert jede LSTM-Zelle auch ihren Zellzustand c_t an die nächsten (nach nächst höhere und rechte) LSTM-Zellen weiter. Die Architektur des Ausgabegatters (engl. output gate) ist in der Box mit

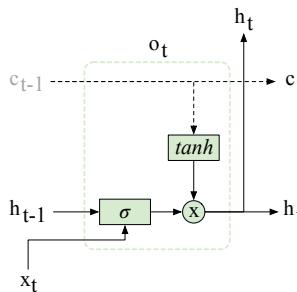


Abbildung 3.30: Architektur des Ausgabegatters einer LSTM-Zelle (das graue Symbole c_{t-1} und die gestrichelten Pfeile zeigen an, dass diese keine direkte Verbindung mit dem Gatter haben).

der grünen gestrichelten Linie in Abbildung 3.30 dargestellt.

Die Struktur ist identisch zum Eingabe- und Ausgabegatter. Allerdings fließen die Informationen diesmal vom neuen Zellzustand c_t in den verborgenen Zustand h_t (von oben nach unten). Dabei erzeugt die TanH-Schicht einen Zwischentensor aus dem Zustandsvektor c_t . Die Sigmoid-Schicht erzeugt eine Bit-Tensor-Maske unter Verwendung des aktuellen Eingangs x_t und des vorherigen Ausgangs h_{t-1} . Der Zwischentensor wird mit dem Bit-Tensor multipliziert, um die endgültige Ausgabe h_t zu erzeugen. Der endgültige verborgene Zustand h_t ist der Ausgabevektor für eine einzelne Ausgabe nach allen Zeitschritten, wenn eine Sequenz auf einen Wert abgebildet wird. In Situationen, in denen jeder Zeitschritt eine Ausgabe erfordert, wenn eine Sequenz auf eine Sequenz abgebildet wird, wird der verborgene Zustand jedes Zeitschritts als Ausgabe betrachtet. In LSTM wird das Langzeitgedächtnis durch den Kontextvektor c_t erreicht, während das Kurzzeitgedächtnis durch den verborgenen Zustand h_t erreicht wird [380].

Formalisierung: Das Ausgabegatter o_t steuert den Bruchteil des internen Zellenzustands, der dann für den verborgenen Zustand h_t verwendet wird. Dazu berechnet das Ausgabegatter o_t die Ausgabe (und den neuen verborgenen Zustand h_t) der Zelle aus dem Zellenzustand

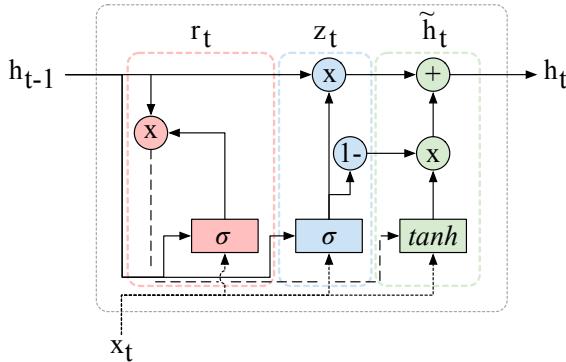


Abbildung 3.31: Gegatterte rekurrente Zelle (GRU), vorgeschlagen von Cho et al. [76]. GRU verwendet die Gatter r_t , z_t und \tilde{h}_t , um den verborgenen Zustand h_t , das Gedächtnis, selektiv zu ändern. Trotz der reduzierten Parameteranzahl lernt die Zelle langfristige und kurzfristige Abhängigkeiten. Gewichte W und Bias b werden nicht angezeigt.

c_t (Gleichung 3.44; die Box o_t mit grüner gestrichelten Linie in Abbildung 3.27 hebt den Mechanismus des Ausgabegatters hervor):

$$o_t = \tanh(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad \text{Ausgabegatter.} \quad (3.52)$$

Gegatterte rekurrente Zelle, GRU

Da LSTM-Zellen sehr viele Parameter umfassen, topologisch sehr komplex sind und einen sehr hohen Rechenaufwand und Speicherbedarf haben, entwickeln Cho et al. [76] die gegatterte rekurrente Zelle (engl. Gated Recurrent Unit, GRU) als vereinfachte Version der LSTM-Zelle die zwei Gatter zusammenfasst. Dazu kombiniert die GRU-Zelle die Vergessens- und Eingabegatter der LSTM-Zelle zu einem neuen Aktualisierungsgatter (engl. update gate) z_t und führt den Zellzustand c_t und den verborgenen Zustand h_t zusammen [14]. Da die Anzahl der Gatter auf drei reduziert wurde, hat die GRU-Zelle im Vergleich zur LSTM-Zelle weniger Parameter und ist daher vergleichsweise schneller. Auf der anderen Seite bietet die LSTM-Zelle einen höheren Wirkungsgrad, wenn genügend Daten und Rechenleistung zur Verfügung stehen [14].

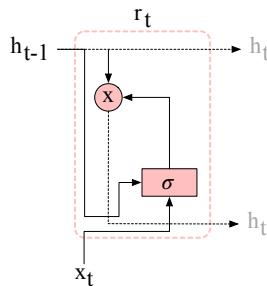


Abbildung 3.32: Architektur des Rücksetzgatters einer GRU-Zelle (die grauen Symbole h_t und die gestrichelten Pfeile zeigen an, dass diese keine direkte Verbindung mit dem Gatter haben).

Abbildung 3.31 zeigt die Struktur und den Informationsfluss der GRU-Zelle. Jozefowicz et al. [217] formalisieren die GRU-Zelle folgendermaßen, \odot bezeichnet das elementweise Hadamard-Produkt:

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \quad \text{Rücksetzgatter,} \quad (3.53)$$

$$\tilde{h}_t = \tanh(W_{x\tilde{h}_t}x_t + W_{r\tilde{h}_t}(r_t \odot h_{t-1}) + b_h) \quad \text{Verborgenes Zustandsgatter,} \quad (3.54)$$

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad \text{Aktualisierungsgatter,} \quad (3.55)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \quad \text{Verborgener Zustand.} \quad (3.56)$$

Alle Gatter arbeiten mit x_t und h_{t-1} , und die zu optimierenden Gewichte sind \mathbf{W} und \mathbf{b} .

Rücksetzgatter

Das Rücksetzgatter (engl. *Reset-Gate*) r_t wird verwendet, um selektiv Teile des verborgenen Zustands h_{t-1} zu löschen, siehe Gleichung 3.53. Dies ähnelt dem Vergessensgatter f_t der LSTM-Architektur und dient als expliziter Vergessensmechanismus [76]. Außerdem können mit dem

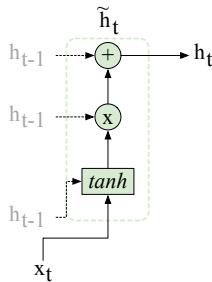


Abbildung 3.33: Architektur des Verborgenen Zustandsgatters einer GRU-Zelle (das grauen Symbole h_t und die gestrichelten Pfeile zeigen an, dass diese keine direkte Verbindung mit dem Gatter haben).

Rücksetzgatter bestimmte Teile des vorherigen verborgenen Zustands \mathbf{h}_{t-1} gelöscht werden, bevor die neue Eingabe \mathbf{x}_t angewendet wird.

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \quad \text{Rücksetzgatter.} \quad (3.57)$$

Verborgenes Zustandsgatter

Ein Kandidat des verborgenen Zustands \tilde{h}_t wird aus r_t und der Eingabe x_t berechnet, siehe Gleichung 3.54:

$$\tilde{h}_t = \tanh(W_{x\tilde{h}_t}x_t + W_{r\tilde{h}_t}(r_t \odot h_{t-1}) + b_h) \text{ Verborgenes Zustandsgatter.}$$

(3.58)

Aktualisierungsgatter

Das Aktualisierungsgatter z_t wird verwendet, um zwischen dem vorherigen verborgenen Zustand h_{t-1} und dem verborgenen Kandidatenzustand \tilde{h}_t zu interpolieren, um den neuen verborgenen Zustand h_t zu berechnen, der als Ausgabe y verwendet wird, siehe Gleichung 3.56. Das Aktualisierungsgatter bestimmt also den Interpolationsfaktor zwischen dem vorherigen verborgenen Zustand h_{t-1} und einem verbesserten verborgenen Zustand \tilde{h}_t , der mit der neuen Eingabe x_t aktualisiert wurde.

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + \mathbf{b}_z) \quad \text{Aktualisierungsgatter.} \quad (3.59)$$

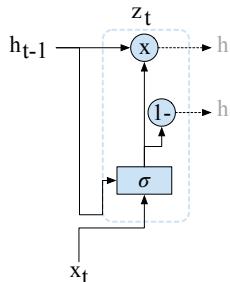


Abbildung 3.34: Architektur des Aktualisierungsgatter einer GRU-Zelle (die grauen Symbole h_{t-1} und die gestrichelten Pfeile zeigen an, dass diese keine direkte Verbindung mit dem Gatter haben).

Identitäts rekurrentes neuronales Netz, IRNN

Identitäts rekurrente neuronale Netze wurden 2018 von Le et al. [254] vorgeschlagen. IRNN-Zellen sind genauso wie die Vanilla RNN-Zelle strukturiert (siehe Abbildung 3.25). Allerdings wird die Aktivierungsfunktion der Vanilla RNN-Zelle von tanh Funktion durch die gleichgerichtete Lineareinheit $\text{ReLU}(x) = \min(0, x)$ ersetzt. Der Hauptunterschied ist, dass die Übergangsmatrix W_{hh} , siehe Gleichung 3.60, zu Beginn des Trainingsprozesses als Identitätsmatrix I initialisiert wird. Le et al. [254] zeigen, dass diese minimalen Änderungen zu einem signifikant besseren Gradientenfluss während der Fehlerrückführung führt.

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h), \text{ mit } W_{hh} = I. \quad (3.60)$$

3.7.5 Architekturen rekurrenter Netze

Die oben eingeführten Zellen lösen eine Vielzahl von Problemen, wie z. B. künstliche Additionsaufgaben oder Musikgenerierung [191, 217, 254] mit mäßigem Erfolg. Daher werden viele verschiedene RNN Architekturen vorgeschlagen, die diese Zellen verwenden, um komplexere Netzarchitekturen zu entwerfen. Die prominentesten Architekturen sind verschiedenen Variationen von (1) gestapelten oder (2) tiefen RNNs [180, 330], spezialisierteren Architekturen wie (3) bidirektionalen RNNs (engl. bidirectional) [392] (siehe Abbildung 3.37), (4) Enkodier-Dekodier-Netze (engl. encoder-decoder networks) [76, 423]

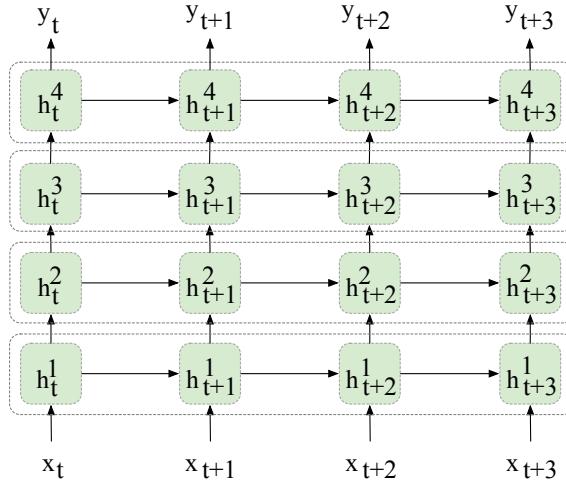


Abbildung 3.35: Die gestapelte RNN-Architektur strukturiert mehrere RNN-Zellen übereinander und nebeneinander und leitet die Ausgabe einer unteren und vorderen (linken) Zelle (h_t^l) bei jedem Schritt an die nächste obere (h_t^{l+1}) und hintere (rechte, h_{t+1}^{l+1}) Zelle weiter. Horizontale Schichten l je Zelle h_t^l in jedem Zeitschritt t , Boxen mit grauen gestrichelten Linien umschließen die Zellen einer Schicht, entstehen implizit entlang der Spalten. Eine vereinfachte Version der Zellen wird für eine einfache Übersicht gezeigt. Abbildung 3.25 zeigt eine vollständig Version derselben Elman-Zelle (Box mit grauer gestrichelter Linie).

und (5) Enkodier-Dekodier Netze mit Aufmerksamkeitsmechanismus (engl. encoder-decoder network with attention mechanism) [23].

Mit Ausnahme der Architekturen (1) bis einschließlich (3) können diese Architekturen jedoch nur verwendet werden, wenn jederzeit eine vollständige Eingabesequenz verfügbar ist. Mit zunehmender Sequenzlänge können lückenlose Datenströme in einer realen Anwendung kaum garantiert werden. Eine mögliche Lösung besteht darin, die Datenströme zu steuern, die Lücken zu füllen und Fehler zu korrigieren. Die Netze verarbeiten dann jedoch Daten aus der Vergangenheit und treffen veraltete Vorhersagen. Daher werden im Folgenden nur die Architekturen (1) bis einschließlich (3) diskutiert.

Gestapelte RNN-Architektur

Tiefe, mehrschichtige, FFNN Architekturen werden in vielen Anwendungen mit großem Erfolg eingesetzt [255]. Es ist offensichtlich, RNN-Zellen auch in einer mehrschichtigen Architektur zu strukturieren. Insbesondere für Aufgaben, die eine umfangreiche Verarbeitung der Eingabedaten erfordern, übertreffen gestapelte-RNN Architekturen (engl. stacked) die einschichtige RNN Architektur deutlich [127, 423].

Abbildung 3.35 veranschaulicht, dass die *gestapelte* RNN Architektur aus mehreren Schichten von RNN-Zellen besteht, wobei die Ausgabe jeder Schicht als Eingabe für die nächste verwendet wird [180]. Dies ermöglicht es dem Netz, kompliziertere Berechnungen in einem einzigen Zeitschritt durchzuführen [196]. Gestapelte RNNs bilden die Basis vieler Sprachmodelle [423]. Die gestapelte RNN Architektur wird mit unterschiedlichen Zellkonzepten und -tiefen in Kapitel 7 umfangreich ausgewertet. Kapitel 4 evaluiert ebenfalls die Auswirkungen der Struktur gestapelter LSTM-Zellen mit unterschiedlicher Breite und Tiefe. Die Kapitel 4 und 7 verwenden unterschiedliche gestapelte Architekturen und beschreiben diese und deren Funktionsweise detailliert.

Dichte-RNN Architektur

Die dichte-RNN-Architektur ist eine Variation der gestapelten-RNN Architektur und basiert auf der Idee von Long et al. [282]. Die dichte-RNN-Architektur verwendet Sprungverbindungen (engl. skip connections), um jeder Schicht zu ermöglichen, die Informationen aller unteren Schichten, inklusive der Eingabesequenz, zu nutzen.

Abbildung 3.36 veranschaulicht, dass die dichte-RNN Architektur der gestapelten-RNN-Architektur ähnelt, es gibt mehrere Schichten l für jedes h_t^l in jedem Zeitschritt t , und fügt Sprungverbindungen (rote gestrichelte Linien in Abbildung 3.36) hinzu, sodass jeder Schicht die Informationen der unteren Schicht als Eingabe zur Verfügung stehen. Pascanu et al. [331] zeigen, dass die Kombination aus tiefer, gestapelter und dichter-RNN Architektur zwischen den Zeitschritten mehr ineinandergreifende Berechnungen durchführen kann, um die Genauigkeit bei einigen Anwendungsbeispielen zu verbessern [331]. Allerdings führen insbesondere die tiefen Sprungverbindungen der dichten-RNN Architektur zu schwankenden und explodierenden Gradienten und zu schwer zu trainierenden Modellen [331].

3 Grundlagen

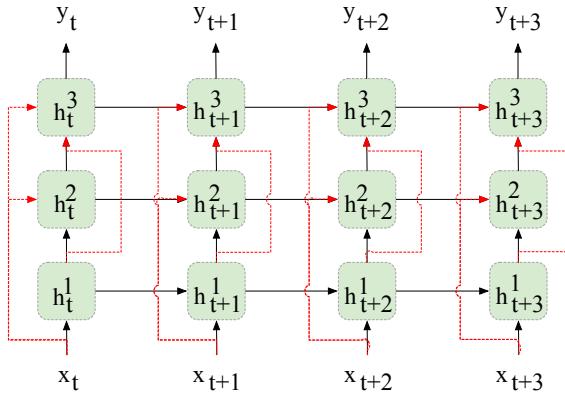


Abbildung 3.36: Das dichte RNN ähnelt der gestapelten Architektur und fügt Sprungverbindungen [282] (rote Linien) hinzu, sodass für jede Schicht jede untere Schicht als Eingabe verfügbar ist. (Rechteck: grundlegende RNN-Zelle).

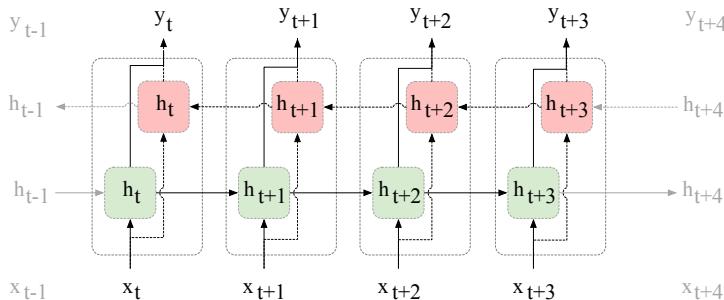


Abbildung 3.37: Bidirektionales rekurrentes neuronales Zeitreihenmodell, das auf Elman-Zellen aufbaut. Für eine einfache Übersicht wird jede Zelle vereinfacht durch eine Box mit grauen gestrichelten Linien und durch den jeweiligen verborgenen Gedächtniszustand h dargestellt. Die grünen Zellen verarbeiten Informationen x und h aus der Vergangenheit (von Zeitschritt t bis $t+3$). Die roten Zellen verarbeiten Informationen aus der Zukunft (von $t+3$ bis t). Die grauen Symbole und gestrichelten Pfeile deuten an, dass die Zeitschritte beliebig in die Vergangenheit $t-1$ und Zukunft $t+4$ verlaufen können.

Bidirektionale RNN-Architektur

Bidirektionale RNNs verarbeiten Sequenzen mit zwei RNNs, von denen eines zeitlich vorwärts läuft (aus der Vergangenheit) und eines

die Serie von hinten (aus der Zukunft) verarbeitet [392]. Diese Modelle sind sehr spezifisch und stark auf ihre spezifischen Aufgaben optimiert.

Die bidirektionale Struktur wird ohne die Rückwärtszustände (rote Boxen in Abbildung 3.37) zu einem regulären unidirektionalen Vorwärts-RNN vereinfacht, vgl. Abbildung 3.18. Wenn die Rückwärtszustände (grüne Boxen in Abbildung 3.37) herausgenommen werden, ergibt sich ein reguläres RNN mit einer umgekehrten Zeitachse.

Wenn beide Zeitrichtungen im selben Netz berücksichtigt werden, können Eingabeinformationen in der Vergangenheit und der Zukunft des aktuell ausgewerteten Zeitfensters (eine Box mit grauer gestrichelten Linie pro Zeitschritt t) direkt verwendet werden, um die Zielfunktion zu minimieren, ohne dass Verzögerungen erforderlich sind, um zukünftige Informationen einzuschließen, wie dies bei der regulären unidirektionalen Zeit der RNNs der Fall ist, d. h. Ein- und Ausgabesequenzen müssen nicht zeitlich verschoben werden. Kapitel 5 verwendet eine ähnliche bidirektionale LSTM (engl. Bidirectional LSTM, BLSTM) Architektur und beschreibt diese und deren Funktionsweise detailliert.

4 Positionsschätzung

Einordnung

Abbildung 4.1 zeigt die Zuordnung der Methode zur Positionsschätzung aus verrauschten Ankunftszeiten eines Funklokalisierungssystems in der übergeordneten Verarbeitungskette der Dissertation. Dieses Kapitel basiert auf zuvor veröffentlichten Ergebnissen [127, 247] und einer unveröffentlichten Arbeit [134]. Im Rahmen dieser Forschungsarbeiten wurde ein Patent eingereicht [132]. Details zur Abgrenzung der Beiträge werden in Abschnitt 1.4.1 erörtert. Zusätzlich werden in den Abschnitten 4.6 und 4.7 unveröffentlichte Ergebnisse präsentiert.

Kurzzusammenfassung

KF sind beliebte Methoden zur Schätzung von Positionsinformationen aus Laufzeitmessung (engl. Time of Flight, ToF) in hochfrequenzbasierten Positionierungssystemen. Solche Bayes-Filter haben sich unter Gaußschen Fehlerverteilungen mit einem Mittelwert von Null als optimal erwiesen. Bei Mehrwegeausbreitung mit gestörter Sichtlinie zwischen Sender und Empfänger treten fehlerhafte ToF-Messungen auf. Mit herkömmlichen Bayesschen Methoden wie KF ist es unmöglich, die Auswirkung der Reflexionen auf die Phasen- und Amplitudenverschiebung in einem Mehrwegeszenario, mit vielen Reflexionen auf den Ausbreitungswegen, zu bestimmen. Somit ist es unmöglich, die tatsächliche Bewegung des Senders und die

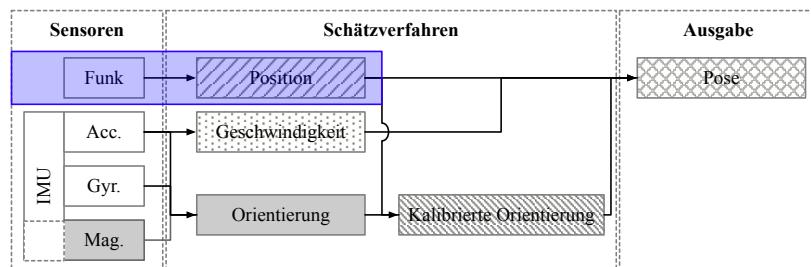


Abbildung 4.1: Einordnung der Positionsschätzung in die übergeordnete Verarbeitungskette (blaue Kästchen umschließen die beteiligten Komponenten).

Verschiebungen aufgrund der Funkumgebung zu bestimmen. Dies wird umso schwieriger, wenn die Umgebung selbst dynamisch ist, d. h. wenn sich die Reflexionen bspw. reflektierenden Objekte selbst bewegen. Unabhängig davon fließen gestörte Sichtlinienverbindungen in Form falscher Ankunftszeitstempel direkt in die Multilateration der Positionsschätzung ein. Numerische Optimierungsmethoden, wie z. B. die kleinsten Fehlerquadrate, eines KF führen dadurch direkt zu hohen Positionsfehlern. Eine Nicht-Sichtverbindung und die damit verbundene fälschlicherweise verzögerte ToF zwingt konventionelle Optimierungsmethoden wie Levenberg-Marquardt [260, 293] dazu falsche globale Optima, m.a.W. falsche Positionen, zu bestimmen, da die Spitzenerkennung des ersten eingehenden Pfads in einer Kanalimpulsantwort verzögert oder überhaupt nicht bestimmt werden kann, wenn die ToA im Rauschen des Signals verschwindet.

Zusätzlich führt die Bewegung eines Senders während der ToF-Schätzung zu sich ändernden Phasen der Mehrwegkomponenten (engl. Multipath Component, MPC), die sich konstruktiv und destruktiv überlagern. Bewegung führt auch zu schnellen Schwankungen in den MPCs mit quasi zufälligen Zusammenhängen, z. B. Mehrwegereflektionen, ausgelöst durch stationäre oder mobile diffuse oder spiegelnde Störer (engl. scatterer). Diese führen zu vielen nicht deterministischen Signalpfaden und damit zu vielen Überlagerungen, die zu unterschiedlichen Zeiten beim Empfänger ankommen. Somit ist es schwierig, die korrekte Echtzeit der Ankunft des Signals zu bestimmen.¹ Eine Drift der ToF-Messungen geht mit der Verschiebung oder dem Schwanken der Phasen zwischen den Messungen einher und führt dazu, dass die zeitliche Auflösung der absoluten Ankunftszeit des Signals falsch ist und die Bewegung des Senders nicht mehr von der Änderung der Umgebung unterschieden werden kann. Bei Betrachtung mehrerer Signale über die Zeit zeigen die ToF-Messungen eine nicht konstante Drift, die sich während der Bewegung ändert. KF kann aufgrund der Dynamik der Umgebung und der Dynamik der Sender und der tatsächlichen Bewegung des verfolgten Objekts weder zwischen driftenden Messfehlern noch verzögerter Ankunftszeit unterscheiden. Somit kann KF die Kurzzeitfehler aufeinanderfolgender Messungen nicht korrigieren und die Langzeitbewegung des verfolgten Objekts nicht genau bestimmen. Daher müssen für KF sehr starre Bewegungsmo-

¹Ein Ankunftszeitmessfehler von 1 ns führt zu einem Distanzfehler von 0,30 m.

delle verwendet werden, was zu einer Divergenz des herkömmlichen Bayesschen Filters führt und so zu ungenauen Positionen.

In diesem Kapitel wird ein sequenzbasierter, datengetriebener Ansatz vorgestellt, der mithilfe von Trainingssequenzen einen nahezu optimalen Positionsschätzer abgeleitet. Ein RNN mit LSTM-Zellen lernt, Drift sowie Umgebungs- und Bewegungsdynamik bei ToF-Messungen eines verfolgten dynamischen Objekts direkt aus ToF-Rohdaten zu korrigieren und Positionen sehr genau und zuverlässig zu bestimmen. Die Auswertung zeigt, dass der datengetriebene Ansatz sowohl bei synthetisch erzeugten als auch bei realen dynamischen Bewegungstrajektorien, die driftende ToF-Messfehler enthalten, ein optimiertes KF aus dem Stand der Technik übertrifft. Die Positionen werden zur Kalibrierung der Posenschätzung der Methoden in Kapitel 7 verwendet.

Das Kapitel ist folgendermaßen strukturiert: Abschnitt 4.1 führt die Positionsschätzung ein. Abschnitt 4.2 beschreibt die Herausforderungen der Positionsschätzung formal. Abschnitt 4.3 diskutiert vorausgegangene Arbeiten aus dem Stand der Technik. Aus der Diskussion leitet Abschnitt 4.4 die Beiträge der Positionsschätzung ab. Abschnitt 4.5 beschreibt die Verarbeitungskette der Methode und deren Architektur. Abschnitt 4.6 beschreibt das Experiment, das zur Auswertung der Methode verwendet wird. Abschnitt 4.7 präsentiert die Ergebnisse der Auswertung. Abschnitt 4.8 diskutiert die Ergebnisse und Limitierungen der Methode, bevor Abschnitt 4.9 die wesentlichen Inhalte des Kapitels zusammenfasst und einen Ausblick gibt.

4.1 Einführung

Die funk basierte Lokalisierung ist eine Schlüsselkomponente vieler Innenanwendungen und industrieller Umgebungen. Üblicherweise empfängt ein Satz synchronisierter Antennen Funk signal Schwingungen mit fester Frequenz von mobilen Sendeeinheiten, um Werte wie Laufzeitmessung, Ankunftszeit ToA und Differenz der Ankunftszeit TDoA zum Schätzen einer Position auszunutzen.

Die Mehrwegeausbreitung von Hochfrequenz-Signalen ist die bekannteste Fehlerquelle, die den ToF-Messungen eine dynamische Drift hinzufügt. In dynamischen Bewegungsszenarien ändern sich die Phasen von Mehrwegekomponenten. Dies führt zu einer ToF-Drift,

4 Positionsschätzung

da aufeinanderfolgende ToF-Messungen durch Verzögerungsfehler aufgrund von Mehrwegeausbreitung beeinträchtigt werden. Je länger eine quasi-lineare Bewegung, d. h. eine konstante Geschwindigkeit auf einer Linie, eines Objekts verfolgt wird, desto besser kann dieses Driftverhalten bspw. durch Bayessche Filter beschrieben werden.

Allerdings bewegen sich Objekte in der Praxis, z. B. in Sport- und VR-Anwendungen, dynamisch und nicht linear und ändern ihr Bewegungsverhalten sowohl kurz- als auch langfristig, z. B. bleiben sie abrupt stehen oder laufen langfristig im Kreis. Daher besteht die größte Herausforderung darin, zwischen der tatsächlichen Bewegung und der Drift der ToF-Messungen zu unterscheiden. Leider können Bayessche Filter diese überlappende Objektdynamik und die durch Mehrwege induzierte Bewegung, d. h. den Drift, nicht korrekt beschreiben, da sie nur den kurzfristigen Kontext berücksichtigen, es fehlt ihnen das Wissen über die Vergangenheit, respektive die Beschaffenheit des Signales in vorangegangenen Messungen und somit fehlt ihnen das Wissen, ob es sich um eine Positionsänderung des Senders handelt oder MPCs das Signal destruktiv beeinflussen. Die Phase und Amplitude einer Kanalimpulsantwort CIR verschiebt sich in einer Mehrwegesituation, da bspw. blockierende Objekte die Signallaufzeit verzögern, m.a.W. verlängern und somit auch der Zeitpunkt, der die ToA- oder TDoA-basierte Schätzung ermöglicht, negativ beeinflussen. Eine irrtümliche Überschätzung des Zeitstempels führt zu einer fehlerhaften Positionsschätzung von numerischen Optimierern, wie z. B. Levenberg Marquardt's kleinste Fehlerquadратsumme in der Multilaterationsphase eines KF's. In markovianischen Bewegungsmodellen [383] hängt der aktuelle Zustand nur von seinem vorherigen Zustand ab, unabhängig von allem, was zuvor passiert ist. Somit können Bayessche Filter langfristige Abhängigkeiten nicht berücksichtigen und bieten daher eine schlechte Positionsgenauigkeit in Mehrwegesituationen mit nichtlinearen Bewegungen, wie sie z. B. in Sport- und VR-Anwendungen bzw. immer dann, wenn Menschen verortet werden, auftreten.

Weiter wird die Genauigkeit von Bayesschen Filtern durch das empirische Wissen begrenzt, das bereitgestellt werden muss, um eine gute Schätzung des Bewegungsmodells, des Prozess- und Messrauschens bereitzustellen. Bayessche Filter sind jedoch nur begrenzt in der Lage, komplexe Bewegungen zu beschreiben, selbst wenn deren

Ablauf genau bekannt ist, m.a.W. wenn das empirische Wissen die Bewegungen genau beschreiben könnte. Um die langfristig abhängige Signaldrift zu überwinden, stimmt ein KF sein Bewegungsmodell so ab, dass es starr ist. Daher kann es nicht auf schnelle Bewegungsänderungen reagieren und divergiert sogar. Werden die Q- und R-Parameter von KF stattdessen so eingestellt, mithilfe einer empirischen a priori Optimierung, dass sie zu den Messungen passen, kann das KF den kurzfristigen Drift der Verzögerungsfehler folgen. Daher liefert ein KF konstruktionsbedingt ungenaue Trajektorienschätzungen, da es entweder dem Mess- und Prozessrauschen folgt oder so starr ist, dass es quasi einer Linie folgt und somit zu träge ist, dynamischen Bewegungsänderungen zu folgen.

Anstatt empirisches Wissen zu verwenden, um das System und die Bewegung eines Objekts zu spezifizieren, werden ToF-Rohdaten und hochpräzise Referenzpositionen verwenden, um ein Filtermodell zu trainieren, das ein RNN mit LSTM-Zellen verwendet. Während einfache neuronale Netze wie faltende neuronale Netze (engl. Convolutional Neural Networks, CNN) oder tiefe neuronale Netze (engl. Deep Neural Networks, DNN) nur Momentaufnahmen der Messung interpretieren, lernt ein LSTM sowohl kurzfristige Abhängigkeiten, z. B. Signalschwankungen wie Phasen- und Amplitudenveränderungen bedingt durch Mehrwege, die implizit in den ToF-Rohdaten eingebettet sind, als auch langfristige Abhängigkeiten z. B. Bewegungsverlauf und -änderungen des Senders von verrauschten Funksignalen. Das zeit- und kontextsensitive LSTM interpretiert zeitabhängige Mehrwegeszenarien und damit driftende Messungen und kann aus ToF-Rohdaten eine genaue und robuste absolute 3D-Position schätzen. In den hier vorgeschlagenen Experimenten wird das LSTM sowohl mit synthetischen Trajektoriendaten, als auch mit realen ToA-Messungen Ende-zu-Ende trainiert, um diese auf submillimetergenaue Referenzpositionen abzubilden. Die synthetischen Daten enthalten sowohl eine langfristige Signaldrift mit Rayleigh Rauschen ohne Sichtverbindungskomponenten als auch Rician Rauschen mit Sichtverbindungskomponenten. Zur Laufzeit kann dieses Modell deutlich genauere und robustere Positionen schätzen als Verfahren aus dem Stand der Technik. Für die Auswertung werden die Schätzungen mit denen eines optimierten KF aus dem Stand der Technik verglichen.

4.2 Problembeschreibung

In dieser Dissertation wird ein Ende-zu-Ende-Modell trainiert, um die aktuelle Trajektorie eines Senders aus einer Reihe von ToF-Werten zu schätzen. Das Problem der Trajektorienschätzung wird in Abschnitt 4.2.1 formal beschrieben und der Drift der ToF wird in Abschnitt 4.2.2 beschrieben.

4.2.1 Terminologie

Die Trajektorie eines Senders ist definiert als:

$$t = [p_0, p_1, \dots, p_i, \dots], \quad (4.1)$$

mit i Referenzpositionen p_i , die die Referenz der Trajektorie im euklidischen Raum darstellen ($p_i \in \mathbb{R}^3$). i bezeichnet den Zeitschritt einer Position. Referenzpositionen werden von hochgenauen Referenzsystemen erhalten. Jede Position p_i hat einen entsprechenden Satz von n ToF-Werten, d. h. n = Anzahl der Empfänger:

$$d_i = [d_{i0}, d_{i1}, \dots, d_{in}], \quad (4.2)$$

mit $d_{ij} \in \mathbb{N}$, die die Position p_i repräsentieren und von n Empfängern mit geringer Genauigkeit für den Zeitschritt i stammen. Daher gibt es für jeden Zeitschritt i eine Referenzposition p_i und einen Satz an ToF-Werten d_i . Die Notation $p_{i-s:i}$ steht für das Teilstück:

$$([p_{i-s}, p_{i-s+1}, \dots, p_{i-1}, p_i]) \quad (4.3)$$

der Trajektorie t , analog für $d_{i-s:i}$, das eine Zeitspanne von $i - s$ bis i der gesamten Trajektorie t fasst. Das Ziel ist es, ein Modell M für die Positionsschätzung $\hat{p}_i = M(d_{i-s:i})$ im Zeitschritt i anzupassen, das auf s zuletzt beobachteten Sätzen an ToF-Werten funktioniert, die von den n -Empfängereinheiten bereitgestellt werden, d. h. eine Eingabesequenz mit 100 Zeitschritten fasst 100 Sätze mit je n ToF-Werten, Dimension=100× n . Für das Training von M werden auch die exakten Referenzpositionen $p_{i-s:i}$ verwendet, um den entsprechenden Satz von ToF-Werten zu kennzeichnen.

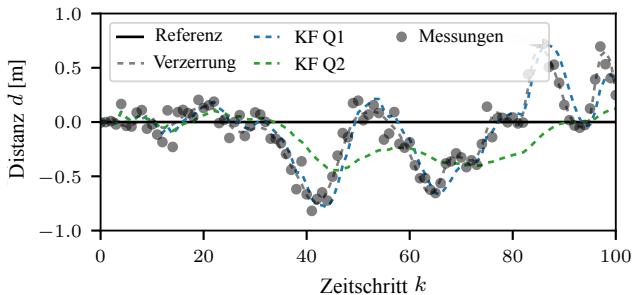


Abbildung 4.2: KF mit verschiedenen Filtereinstellungen (Q -Matrizen), die auf einen Drift angewendet werden. Agile Einstellungen bewirken, dass das KF der Drift folgt (Q_1). Starre Einstellungen führen zu driftenden Schätzungen (Q_2). Sehr starrere Einstellungen führen zu Filterdivergenz.

4.2.2 Driftende ToF-Fehler

Theoretisch kann die Trilateration die korrekte Position p_i aus einer Reihe exakter ToF-Messungen

$$d_i = \{d_{i0}, d_{i1}, d_{i2}\} \quad (4.4)$$

bestimmen, d. h. wenn Sender und Umgebung statisch sind und keine Nicht-Sichtverbindungs MPCs vorhanden sind. In der Praxis bewegen sich Sender und Umgebung. Objekte verzögern die Signallaufzeit und führen zu ungenauen ToF-Daten. In der Multilateration, treffen sich ihre Radien daher nicht in einem einzigen Punkt, im globalen Optimum, und die geschätzte Position \hat{p}_i weicht erheblich von der tatsächlichen Position ab. Das Problem besteht darin, dass das Ortungssystem Fehler und Artefakte in der Signalverarbeitungskette hervorruft. Solche Artefakt in jedem funkbasierten Signalverarbeitungssystem sind die Mehrwegeausbreitung und Signaldrift.

Üblicherweise ist der Mehrwegekanal durch eine diskrete Anzahl von MPC gekennzeichnet, siehe Abschnitt 3.3. Im Allgemeinen wird die Impulsantwort h eines Funkkanals als abgegriffene Verzögerungsleitung (engl. tapped delay line) mit L MPCs beschrieben:

$$h = \sum_{l=1}^L a_l \cdot \delta(t - \tau_l), \quad (4.5)$$

4 Positionsschätzung

wobei a_l der komplexe Koeffizient vom l -ten MPC ist und τ_l die Verzögerung des MPC ist die durch Streuung von Objekten entsteht [344].

Neuere statistische Kanalmodelle [205] für eine Streuumgebung sind üblicherweise durch Parameter für große Flächen (engl. Large Scale Parameters, LSP) wie anfängliche Verzögerungen (engl. initial delay), komplexe Koeffizienten (engl. complex coefficients) und Ankunftsinkel (engl. arrival angles) für das einzelne MPC gekennzeichnet. Um die räumliche Konsistenz sicherzustellen, müssen diese Werte entlang der Trajektorie des sich bewegenden Objekts aktualisiert werden. Andernfalls passen die Schätzungen der Kanalkoeffizienten nicht zu den realen, wenn für jede Position auf der Trajektorie ein neuer Kanal realisiert wird. Um solche Effekte zu vermeiden, wird eine Trajektorie in Segmente, von bis zu mehreren Metern, unterteilt. Innerhalb eines Segments wird angenommen, dass die Mittelwert- und Autokorrelationsfunktionen des Kanals zeitinvariant sind (engl. Wide Sense Stationary, WSS), d. h. die LSPs ändern sich nicht signifikant und lassen die Streuumgebung unverändert, d. h. die Umgebung ist statisch. Aufgrund kleiner Bewegungen ändern sich jedoch die Pfadverzögerungen. Dies bewirkt eine Änderung der komplexen Kanalkoeffizienten: die Phase des MPC ändert sich, das von Natur aus zu driftenden ToF-Fehlern führt [205].

Wird ein Sender entlang eines Segments bewegt, schwankt die Phase und die ToF-Schätzungen driften. Dieses Phänomen wird häufig in ToF-basierten Systemen [316] beobachtet. Folglich können die ToF-Fehler nicht durch eine mittelwertfreie Wahrscheinlichkeitsdichtefunktion (engl. zero-mean probability density function, PDF) beschrieben werden. Kurzfristig sind die ToF-Fehler vorgespannt und driften, d. h. ToF-Fehler folgen einer zufälligen (stochastischen) Irrfahrt (engl. random walk) Bewegung der Messpunkte, die mit der wahren Bewegung des Senders überlagert ist. Die Herausforderung besteht darin, diese Bewegungen, die in den rohen ToF-Werten eingebettet sind, zu trennen. Abbildung 4.2 zeigt die Anwendung von KF mit verschiedenen Parametern bei Messungen mit driftenden ToF-Fehlern bei statischen Störern und Empfängern und einem beweglichen Sender.

4.3 Verwandte Arbeiten

Abschnitt 4.3.1 erläutert zunächst verwandte Arbeiten, die das Kalman- und Partikel-Filter zur Positionsbestimmung verwenden. Abschnitt 4.3.2 diskutiert numerische Optimierungsmethoden und Abschnitt 4.3.3 erläutert Arbeiten, die Methoden des maschinellen Lernens verwenden. Schließlich diskutiert Abschnitt 4.3.4 hybride Prozesse, die modell- und datengetriebene Verfahren koppeln.

Hochfrequenzbasierte Positionierungssysteme leiden unter Signalausbreitungseffekten wie Dämpfung (engl. attenuation), Streuung (engl. scattering), Beugung (engl. diffraction) und Reflexion (engl. reflection), die zu Mehrwegeeffekten führen [426, 436].

4.3.1 Kalman- und Partikel-Filter

Naive eigenständige Bayessche Filter wie KF und PF verwenden ein (lineares) Bewegungsmodell und einen Übergangszustand. Sie kalibrieren und glätten die reale physikalische Position (neu), indem sie kontinuierlich Mess- und Prozesskovarianzmatrizen schätzen. Damit dies in der Praxis funktioniert, sind viel Fachwissen und eine manuelle Feinabstimmung der Filterparameter erforderlich [88].

Da sich die Sensoreigenschaften (und das Rauschen) mit der Zeit ändern, kann ein a priori parametrisiertes Filter den realen Zustand nicht korrekt darstellen und gibt falsche Positionsschätzungen zurück. Wichtiger ist jedoch, dass die Kovarianzmatrizen nur eine Momentaufnahme der Sensor- und Bewegungseigenschaften darstellen und daher keine langfristige Abhängigkeit eines sich bewegenden Objekts beschreiben können. Selbst beliebte Verbesserungen wie EKF und Sigma-Punkt-KF können stark nichtlineare Bewegungsmodelle nicht korrekt beschreiben, da sie eine unendliche Anzahl linearer Bewegungsbeschränkungen definieren müssten [528]. Während PFs die nichtlinearen Bewegungen und somit auch die Position in statischen Szenarios korrekt approximieren können, wenn genügend verlässliche Partikel vorhanden sind, liefern sie in hochdynamischen Situationen oder wenn die Sichtverbindung zwischen Sender und Empfänger eine neue Initialisierung des PF benötigt, schlechte Ergebnisse [384].

4.3.2 Numerische Optimierungsverfahren

Optimierungstechniken wie Bancroft (BC), Gradientenabstiegsverfahren und Levenberg Marquardt (LM) lösen das nicht konvexe Optimierungsproblem und schätzen eine Position, die den Fehlerterm [71] minimiert. Dabei approximieren sie das globale Optimum der Multilateration mehrerer Empfänger und eines Senders zuverlässig, wenn Signallaufzeiten eindeutig und Ankunftszeitpunkten ohne Verzögerung bestimmt werden können m.a.W. unter direkter Sichtbedingung der MPCs. Sie berücksichtigen jedoch keine aufeinanderfolgenden Messungen, die sich über die Zeit erstrecken. Dies schränkt die Genauigkeit ein. Dong et al. [104] zeigen, dass aufeinanderfolgende Schätzungen von LM nur unter Laborbedingungen ein statistisches Optimum darstellen und daher nicht für reale Szenarien anwendbar sind. Darüber hinaus liefern diese Optimierungstechniken lokale Optima, wenn es keine direkte Sichtverbindung zwischen Sender und Empfänger gibt, da hier mindestens ein Pfad und somit mindestens eine Verzögerung der Ankunftszeit geschätzt wird, die die Lösung der Optimierer verfälscht und so zu einer fehlerhaften Position führt.

4.3.3 Verfahren des Maschinellen Lernens

Entscheidungsbaum, k-nächster Nachbar, Stützvektormaschine (engl. Support Vector Machine, SVM), KNN und andere Klassifizierungsmethoden schätzen genaue Positionen. Während sie für Signalstärkedaten wie RSS [91] beliebt sind, gibt es auch Arbeiten, die auf ToF-Daten [405, 518, 91, 426, 512] aufbauen. Durch das Design extrahieren KNN automatisch relevante Merkmale, die die absolute Position beschreiben. Sie können auch die Auswirkungen der Mehrwegeausbreitung in statischen Szenarien lernen, z. B. wenn einige Funksignale von Objekten wie Wänden [312] reflektiert werden. Alle diese Modelle setzen jedoch statische Umgebungen voraus, sodass sie dynamische Änderungen der Umgebung nicht kompensieren und keine dynamischen Situationen, mit bewegten Sendern, verarbeiten können, die sich auf die ToF-Fingerabdrücke auswirken. In dynamischen Situationen bewegen sich die Sender, bspw. wenn sie am Kopf oder in der Hosentasche eines Sportlers oder VR Benutzers angebracht sind, und können (unbekannte) Reflexionen erzeugen, die zu falschen Positionsschätzungen führen.

4.3.4 Hybride Verfahren

Hybride Verfahren kombinieren die grobe Positionsschätzung eines KNNs mit einer Nachbearbeitung in einem KF [397], LM [71] oder Hidden Markov Model (HMM) [512]. Sowohl die Vor- als auch die Nachbearbeitung leiden jedoch unter komplexen Parametrisierungsanforderungen. Außerdem bieten diese Verfahren nur lineare Bewegungsmodelle, die nur kurzzeitige Ursachen, d. h. Zeit zwischen zwei Filteraktualisierungen, berücksichtigen und daher keine langfristigen Signaldrift erkennen können. Al-Mayyah et al. [6] wenden LM auf ein KNN auf simulierten Trajektorien an und zeigen, dass ihre Simulation die Fahrzeugsteuerungsparameter erfolgreich optimiert und den absoluten Positionsfehler reduziert. Während ihr Ansatz in einem Labor funktioniert, berücksichtigen sie keine reale Bewegung.

Der Bewegungskontext ist wichtig, da er die Beziehung zwischen aufeinanderfolgenden Positionen über einen (sehr) langen Zeitraum beschreibt. LSTM [191] kann solche Beziehungen modellieren [8, 300, 202]. Yousefi et al. [498] untersuchen die Leistung von RNNs und LSTM auf Funksignalstärke-(RSS)-Fingerabdrücken. Sie kombinieren die Grobpositionsschätzung aus einem LSTM mit der Feinpositionierung aus einem HMM. Während ihre Genauigkeit frühere Ansätze übertrifft, ist ihr HMM in seinem Bewegungsmodell, aufgrund seines *a priori* Designs, eingeschränkt. RNNs in Kombination mit RF-basierten Merkmalen werden in [47] untersucht, jedoch nur, um das Signal-Rausch-Verhältnis des Verfolgungssystems zu bestimmen. Basierend auf diesen Voraarbeiten wird in dieser Dissertation ein LSTM zum Identifizieren, Lernen und Rekonstruieren von Zeitabhängigkeiten von Informationen anhand von Funksignalen verwendet.

Hybrid KF [88, 466, 248, 397] adressiert das Problem der unzureichenden Schätzung des Prozess- und Messrauschen unter Kenntnis der zugrunde liegenden Sensoreigenschaften. Odelson et al. [319] können die Kovarianzmatrizen mit Auto-Kovarianz-kleinste-Fehlerquadrat-Methoden schätzen, benötigen jedoch eine große Stichprobengröße, um zu konvergieren, und sie schätzen nur entweder die optimalen Mess- und Prozessrausch-Kovarianzen [528] oder Zustandsübergangsfunktionen [319], aber nicht beide. Leider hängt die Genauigkeit der Positionsschätzung eines KFs von beiden ab. Coskun et al. [88] lernen diese Parameter aus Datenproben bei der bildbasierten

Posenschätzung. Sie verwenden drei gestapelte LSTM-Schichten, um die nichtlineare Übergangsfunktion zu lernen. Zwei zusätzliche einzelne LSTM-Schichten lernen die Prozess- und Messkovarianzmatrizen. Dieser Ansatz übertrifft alle bisherigen bildbasierten Positionierungsmethoden. Obwohl ihr Ansatz konzeptionell auch an die ToF-basierte Positionierung angepasst werden könnte, würde das Erlernen jedes (vergleichsweise) kurzfristigen Bewegungstyps viele rechenintensive Bilddaten erfordern. Da sie nur RGB-Daten verwenden, funktioniert ihr Ansatz nicht mit kleinen und begrenzten numerischen Informationssequenzen. Bei längeren numerischen Sequenzen wird nur ein kurzer Bruchteil sich wiederholender Bewegungstypen bereitgestellt.

Perez-Ortiz et al. [337] und Iter et al. [202] argumentieren, dass ihr KF-LSTM besser verallgemeinert als ein eigenständiges LSTM, da die implizite Fusion der vergangenen Bewegungszustände und der aktuellen Messungen zu komplex sei, um von einem eigenständigen LSTM gelernt zu werden. Im Gegensatz dazu lernt das hier im Rahmen der Dissertation vorgeschlagene LSTM kontextbezogene Bewegungsinformationen, um die langfristige Signaldrift optimal zu modellieren. Da das Modell genaue und präzise Positionen schätzt, erweist sich sein implizit erlerntes Bewegungsmodell als angemessen modelliert und generalisiert darüber hinaus auf verschiedene zufällige und dynamische Bewegungen in Sichtverbindungen, d. h. Rician Rauschen mit mindestens einer Sichtkomponente. Weiter kann das Modell auch unter Mehrwegeausbreitungen, d. h. Rayleigh Rauschen ohne Sichtkomponente, Positionen zuverlässig und genau schätzen.

4.4 Beiträge

Um physikalisch korrekte Modelle von Funkkanälen zu erstellen, wird die funkbasierte Verarbeitungskette in kleinere atomare Segmente aufgeteilt. Die Auswertung der Experimente zeigt, dass die Modelle lernen können, mit Signaldrift umzugehen und optimierte KF Filter des Stands der Technik übertreffen. Der vorgeschlagene datengetriebene Ansatz adressiert die Grundprobleme der Funklokalisierung: Phasen- und Amplitudenverschiebung durch Mehrwegekomponenten bei bewegten Funksendern oder gestörten Sichtbedingungen (engl. Obstructed Line of Sight, OLoS), die langfristig zu Signaldrift führen. Anstatt Ende-zu-Ende-Lernen auf die gesamte Verarbeitungskette der Funklo-

kalisierung anzuwenden, sollten die wesentlichen Komponenten, die zur Beschreibung des Kommunikationskanalmodells erforderlich sind, durch einzelne datengetriebene Komponenten modelliert und somit besser kontrolliert werden und führen so zu einer genaueren und robusteren Positionsschätzung. Da der datengetriebene Positionsschätzer die in dieser Dissertation vorgeschlagene PDR-Methode kalibriert, stellen sich folgende Forschungsfragen, RQ:

RQ2: Können datengetriebene Verfahren die Positionsschätzung eines dynamisch bewegten Funksenders aus verrauschten Ankunftszeitmessungen verbessern?

- RQ2a: Kann eine datengetriebene Methode Mehrwegepfade in Ankunftszeiten in Funksystemen auflösen, um die Positionsgenauigkeit in gestörten Sichtverbindungen zu erhöhen?
- RQ2b: Kann eine datengetriebene Methode langzeitabhängiges stochastisches Rauschen in synthetischen Ankunftszeiten auflösen, um die Positionssgenauigkeit zu erhöhen?
- RQ2c: Welche Auswirkung hat die Architektur einer rekurrenten datengetriebene Methode auf Positionssgenauigkeit?

4.5 Methode

Abschnitt 4.5.1 motiviert die Wahl der datengetriebenen Methode und stellt deren Mechanik vor. Abschnitt 4.5.2 beschreibt das Modell, das zur Positionsschätzung verwendet wird und in Abschnitt 4.7 ausgewertet wird. Abschnitt 4.5.3 beschreibt die Datenvorverarbeitung.

4.5.1 Einführung

Zunächst werden KNN eingeführt und ihre Schwierigkeiten bei der Zeitreihenanalyse diskutiert (Abschnitt 4.5.1). Anschließend werden RNN eingeführt (Abschnitt 4.5.1) und die Verwendung von LSTM-Zellen motiviert, die in Abschnitt 4.5.1 (vereinfacht) beschrieben werden. Kapitel 3 erörtert detailliertere Informationen.

Mehrschichtige Perzeptronen (MLP)

KNN wie mehrschichtige Perzeptronen MLP bestehen aus einer Eingabeschicht X , einer oder mehreren verborgenen Schichten H und

einer Ausgabeschicht Y . Eingangssignale breiten sich in Vorwärtsrichtung durch das Netz aus: Eine Verlustfunktion vergleicht den Ausgang mit der Referenzbezeichnung, um den Fehler zu berechnen. Fehlersignale breiten sich rückwärts durch das Netz aus und angepasste Gewichte reduzieren den Fehler. Positionierungsanwendungen des Stands der Technik verwenden häufig Varianten wie Radial Basis Function Networks (RBFN) [405]. Alle MLP-Varianten können jedoch nur eine feste Eingabesequenzlänge von Fingerabdrücken (engl. fingerprint), in der Literatur häufig auch als Schnappschüsse (engl. snapshots) bezeichnet, mit ToF-Werten d_i zum Zeitpunkt i annehmen, um \hat{p}_i vorherzusagen, und es fehlt ihnen die Fähigkeit, die Beziehung von zeit- und kontextsensitiven aufeinanderfolgenden Positionen einer Trajektorie zu beschreiben, da sie keine rekurrente Charakteristik besitzen, d. h. die Fähigkeit, Eingaben und den vorherigen Zustand zu berücksichtigen. MLPs bieten keine Möglichkeit, latente Bewegungseigenschaften oder stochastisches Rauschen mit langfristigem Trend zu berücksichtigen, die eine Folge von Messungen kodieren, um Zusammenhänge in Bewegung und Rauschen in einem Signal zu trennen.

Bewegung ist ein zeitlicher und dynamischer Prozess, der von verschiedenen Faktoren wie Beschleunigung, Geschwindigkeit und Richtung abhängt, die nur indirekt aus den Sätzen von ToF-Werten d_i und ihren entsprechenden Positionen p_i beobachtet werden können. Darüber hinaus sind diese Werte auch verrauscht und weisen driftende Signalartefakte auf. Daher muss das Modell M so angepasst werden, dass es die latente Dynamik dieser Faktoren aus dem Strom von ToF- und Positionsdaten erfassen kann.

Rekurrente neuronale Netze (RNN)

Während MLPs so konzipiert werden könnten, dass sie einen Stapel vergangener Messungen verarbeiten können, um \hat{p}_i vorherzusagen, sind RNNs die bessere Wahl [92]. RNNs sind eine Klasse neuronaler Netze, die Datensequenzen verarbeiten, hier: aufeinanderfolgende Sätze von ToF-Messungen, und erfolgreich auf verschiedene zeit- und kontextsensitive Aufgaben angewendet wurden, z. B. Sprache [423], Musikmodellierung [42] und Trajektorienvorhersage [11, 15]. RNNs haben persistente interne Zustände c_i , den Speicherzustand der Zelle, die aus der Eingabesequenz X_i aktualisiert werden und die über Übergänge übertragen werden, z. B. von H_i zu H_{i+1} . Dadurch kann das Modell das dynamische Verhalten aus den Eingabedaten erfassen und die Ausgabe

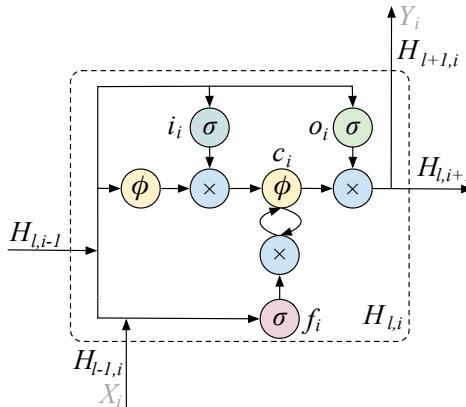


Abbildung 4.3: Grundlegende Lang-Kurzzeitgedächtniszelle.

Y_i vorhersagen. Die Aktualisierungen eines RNN-Zellenzustands, wenn neue Eingabedaten X_i eintreffen, sind definiert als:

$$H_i = g(W_h \cdot H_{i-1} + W_{in} \cdot X_i + B_h), \\ Y_i = a(W_o \cdot H_i + B_o).$$

Hier sind $g(\cdot)$ und $a(\cdot)$ (nichtlineare) Aktivierungsfunktionen, W sind gemeinsame Gewichtsmatrizen und B sind Bias-Terme. Da RNNs in der Praxis unter verschwindenden oder explodierenden Gradienten leiden, erkennen sie kaum Abhängigkeiten über große zeitliche Entfernung, d. h. zwischen zeitlich weit auseinanderliegenden Messungen.

Lang-Kurzzeitgedächtniszellen (LSTM)

LSTM-Netze [191] vermeiden diese (verschwindenden oder explodierenden Gradienten) Probleme, da sie die Berechnung von H_i durch eine Zelle ersetzen, die drei *Gatter* (engl. gate) enthält: ein Vergessensgatter f_i , ein Eingangs- i_i und ein Ausgabegatter o_i zum Zeitschritt i . Diese Zellen modellieren explizit den Informationsfluss zwischen aufeinanderfolgenden Zeitschritten. Abbildung 4.3 zeigt die interne Struktur einer grundlegenden LSTM-Zelle. Hier repräsentiert σ die Sigmoid-Gatter-Aktivierungsfunktion (engl. sigmoid gate activation function) für die Vergessen- f_i , Eingabe- i_i , Ausgabe- o_i Gatter, ϕ ist die TanH Eingabe / Ausgabe-Aktivierungsfunktion und \times ist ein

4 Positionsschätzung

Multiplikationsoperator. Der Speicherzustand c_i der Zelle wird in jedem Schritt mit Informationen sowohl vom Vergessen- als auch vom Eingangsgatter aktualisiert. H_i ist der neue Zustand, der durch das Ausgabegatter aus der Speicherzelle c_i extrahiert wird. Informationen fließen durch den LSTM von links nach rechts und von unten nach oben. Die anfängliche Eingabe X_i geht in die unterste LSTM-Schicht $l=0$ ($H_{l=0,i}$). Jede Zelle $H_{l,i}$ speist auch die benachbarte Zelle auf derselben Schicht $H_{l,i+1}$ von der Seite und die benachbarte Zelle auf der nächsthöheren Schicht $H_{l+1,i}$ von unten.

Während der Trainingsphase lernt das Modell M , latente Bewegungseigenschaften, langfristige Trends und Trajektorien aus einer aufeinanderfolgenden Folge von Positionen p_i und entsprechenden Sätzen von ToF-Werten d_i zu nutzen, indem es die rohen ToF-Werte auf korrespondierende Positionen abbildet. Während der Live-Phase schätzt ein trainiertes M genaue und präzise Schätzungen der aktuellen Position und Trajektorie des Senders auf Sätzen von rohen ToF-Messungen.

Die folgenden Gleichungen definieren den Vorwärtsdurchlauf einer LSTM-Zelle für den verborgenen Zustand H_i in Schritt i . U sind weitere Gewichtsmatrizen, die zwischen den Zellen geteilt werden und \times ist der elementweise Multiplikationsoperator aus Abbildung 4.3.

$$\begin{aligned} i_i &= \sigma(W_i X_i + U_i H_{i-1} + B_i); \\ o_i &= \sigma(W_o X_i + U_o H_{i-1} + B_o); \\ f_i &= \sigma(W_f X_i + U_f H_{i-1} + B_f); \\ c_i &= f_i \times c_{i-1} + i_i \times \tanh(W_c X_i + U_c H_{i-1} + B_c); \\ H_i &= o_i \times \tanh(c_i). \end{aligned}$$

Die Breite eines Modells wird durch die Größe einer LSTM-Zelle definiert. Die Größe einer einzelnen Zelle wird durch die Anzahl der möglichen Gewichte und Bias definiert, die jede Zelle verarbeitet. In Abbildung 4.3 repräsentiert jedes Tröpfchen (engl. blob) (Funktion, Operator, Gatter oder Zelle) eine fest bezeichnete Funktionalität, verkörpert aber auch einen Stapel von Gewichten und Bias gemäß der Zellengröße, d. h. eine Zellengröße von 128 bedeutet, dass jedes Tröpfchen in der Zelle 128 Gewichte und 128 Bias enthält.

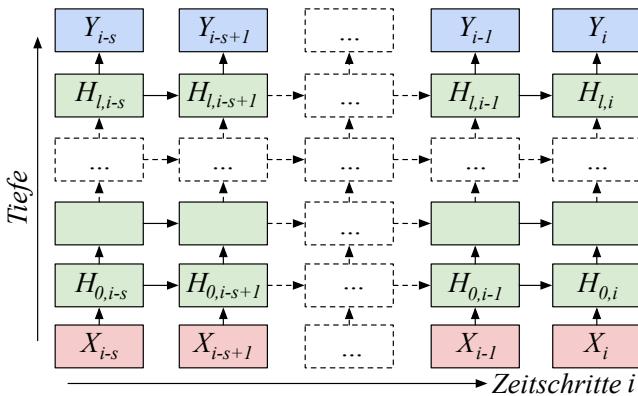


Abbildung 4.4: Vorgeschlagene entrollte Viele-zu-Viele LSTM-Modellarchitektur mit l verborgenen LSTM-Schichten H für einen Teil s der Trajektorie, der i Zeitschritte abdeckt. Die Eingaben X versorgen die verborgenen Schichten mit Sätzen von ToF-Messungen. Die vorhergesagten Positionen sind in Y . (Vollständig verbundene- und Verwerfungsschichten werden nicht explizit beschrieben.)

4.5.2 Architektur

Abbildung 4.4 zeigt das vorgeschlagene LSTM-Modell. Zur Vereinfachung für jeden Zeitschritt i (Spalte) wurden beide Verwerfungsschichten zwischen den verborgenen Schichten H_0 bis H_l und auch die dichte, vollständig verbundene Schicht kurz vor der Ausgabe-Schicht Y wurde für eine übersichtliche Darstellung weggelassen.

Während der Trainingsphase nimmt das Modell Schnitte aufeinanderfolgender Trajektorienpositionen $p_{i-s:i}$ und ihre entsprechenden Sätze von ToF-Werten $d_{i-s:i}$ als Eingabe auf. Die Sequenzlänge s definiert, wie viele solcher aufeinanderfolgenden Bündel das Modell in jeder Iteration verarbeitet. Je länger die Sequenz ist, desto mehr Bewegungsabhängigkeiten, oder Zusammenhänge des stochastischen Rauschen und langfristige Trends, lernt das Modell richtig zu interpretieren, aber desto komplexer wird das Modell. Das Modell lernt nicht nur, wie eine einzelne Position aus einem einzelnen Satz von ToF-Werten erhalten wird, sondern auch, wie aufeinanderfolgende Positionen aus Sätzen von aufeinanderfolgenden ToF-Werten vorhergesagt werden. Während X_i eine Eingabe darstellt, die zu den Zeitschritten i gesetzt wurde, repräsentiert Y_i die resultierende Positionsschätzung. Mit einem zufäl-

lig initialisierten Anfangszustand H_0 hat das Modell keine Vorkenntnisse über die Bewegung und lernt folglich, die Position aus lokalen Informationen zu schätzen. Als Ausgabe Y schätzt das Modell die aktuellen Positionen des Senders \hat{p}_i .

4.5.3 Datenvorverarbeitung

Modelle für maschinelles Lernen leiden unter Daten mit hoher Varianz und Abweichung. Ein numerischer winziger (lokaler) Gradient oder Gewicht explodiert, wenn dem Neuron sehr große Werte zugeführt werden. Andererseits verschwindet ein großes Gewicht mit sehr kleinen Aktualisierungswerten. Da solche Modelle am besten funktionieren, wenn die Eingabedaten *Mittelwertfrei* (engl. zero mean) und *Einheitsvarianz* (engl. unit variance) haben, wird der Datensatz normalisiert auf $[-1; +1]$ durch Subtrahieren des *Mittelwertes* und Skalieren mit der *Standardabweichung* über die Trainingssätze für jedes einzelne Merkmal (ToF-Wert) [75]. Nach dem Training kann das Modell auf Sätze mit unbekannten ToF-Daten angewendet werden.

Für das Training von M wird die gesamte Trajektorie in aufeinanderfolgende Sequenzen der Länge s geschnitten und Teiltrajektorien, sogenannte Segmente, ausgehend von zufälligen Startpunkten in der Trajektorie, wie ein Schiebefenster, erstellt. Zu beachten ist, dass das Modell in der Live-Phase aufeinanderfolgende Positionen $p_{i-s:i}$ nur für aufeinanderfolgende Sätze von ToF-Messungen $d_{i-s:i}$ vorhersagt, um eine zusammenhängende Trajektorie zu schätzen.

Datenstapelung

Abbildung 4.5 bietet einen detaillierten Einblick in die Eingabedaten-Stapelverarbeitungsmethode. Insgesamt gibt es für jeden Zeitschritt i in der Sequenz s einen Stapel G , unabhängige Datengruppen. Ein Stapel besteht aus j Gruppen $g_{j,i}$ für jeden Zeitschritt i , von $i-s$ zu i . Jede Gruppe verkörpert die Position p_i , nur während der Trainingsphase, und die entsprechenden ToF-Werte d_i , während der Trainings- und Live-Phase. Jede Gruppe g in einem Stapel G repräsentiert eine Position einer anderen Trainingstrajektorie t_j , d. h. der Stapel enthält Daten zum Zeitschritt i von allen aufgezeichneten Trainingstrajektorien. Zu beachten ist, dass während der Live-Phase nur $G_{X_{i-s:i}}$ mit jeweils einem einzelnen $g_{j=0, X_{i-s:i}}$ von M verarbeitet wird. Die drei Reihen

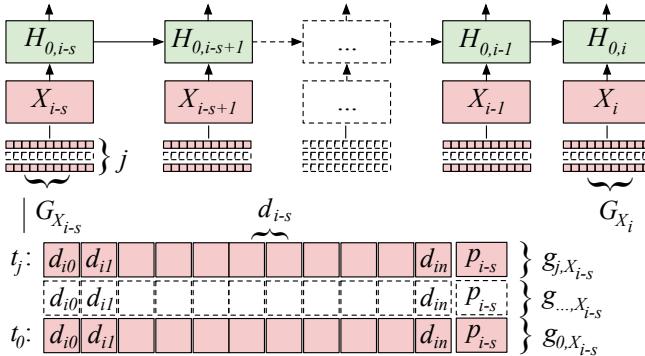


Abbildung 4.5: Datenvorbereitungsprozess, der Stapel G von ToF-Werten d zu jedem Zeitschritt i vorbereitet und diese als Eingabe X_i der ersten verborgenen Schicht H_0 des LSTM-Netzes zuführt.

kleiner Kästchen in Abbildung 4.5 zeigen ein einfaches Beispiel dafür, wie die Stapel G für jeden einzelnen Zeitschritt von $i-s$ bis i aussehen.

Eine Sequenz wird in $i-s : i$ Stapel G_{X_i} geteilt, die j Gruppen $g_{j,X_{is}}$ enthalten. Während des Trainings bezeichnet j die Anzahl der verschiedenen Trainingstrajektorien und während der Live-Phase gilt $j=1$. Jede Gruppe $g_{i,X}$, Gruppe g im Zeitschritt i mit Eingabe X , mit X enthält entweder p_i und d_i während des Trainings oder nur d_i während der Live-Phase, enthält eine Reihe von ToF-Werten d_i , die eine Position p_i von einer Trajektorie t_j beschreiben. Die Anzahl der verschiedenen Trajektorien j definiert die Größe des Stapels G mit

$$G_{X_{is}} = \{g_{0,X_{is}}, g_{1,X_{is}}, \dots, g_{j,X_{is}}\}. \quad (4.6)$$

Die Sequenzlänge s definiert die Anzahl der Zeitschritte, die das Modell berücksichtigt, um neue Positionen, d. h. wie viele ToF- und Positionssätze dem Netz zugeführt werden, pro Iteration vorherzusagen. So mit enthält jede Sequenz s Stapel G und jeder Stapel G enthält j Gruppen g von n ToF-Werten d . Jedes g_j in G repräsentiert einen unabhängigen Teil der Bewegungstrajektorie, sodass das Modell verschiedene zeitbasierte Bewegungsbeziehungen lernt. Solche zeitabhängigen Beziehungen umfassen Änderungen in bspw. Beschleunigung, Geschwindigkeit, Versatz, Trajektorie, Rauschen und andere Bewegungsarten. Eine Zunahme der Tiefe, d. h. der Anzahl der Schichten, führt zu einer Zunahme der Vielfalt verschiedener zeitbasierter Eigenschaften, d. h. je

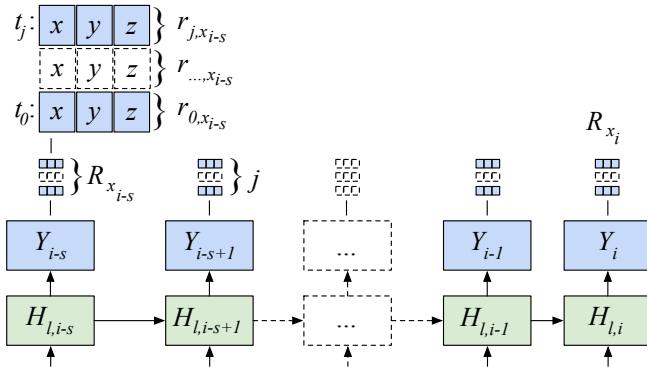


Abbildung 4.6: Vollständig verbundene Schicht, die die Ausgabe der letzten verborgenen Schicht H_l des LSTM-Modells auf Y verdichtet und dreidimensionale absolute Positionen $\hat{p}_{i-s:i}$ zurückgibt.

tiefer das Netz ist, desto anspruchsvollere zeitabhängige Beziehungen können durch das Netz modelliert werden [180].

Dateneinspeisung

Im Training werden Sätze von ToF-Werten X_i und ihre entsprechenden Referenzpositionen und live werden Sätze von ToF-Werten X_i nacheinander an die LSTM-Zellen weitergegeben, siehe Abbildung 4.5. Z. B. werden 200 LSTM-Zellen der ersten verborgenen Schicht $H_{0,i-s:i}$ mit Eingaben $X_{i-s:i}$ mit einer Sequenzlänge von $s=200$ gespeist. Mit jedem Schritt i wird der Zustand $H_{i-s:i}$ aktualisiert und sammelt Informationen über die Änderungen in den Positionen. Eine größere Zelle wird verwendet z. B. 256 Neuronen, um komplexere Probleme zu lösen [331]. Nachdem die vollständige Sequenz verarbeitet wurde, enthält der endgültige LSTM-Zustand H_i eine Darstellung der beobachteten Bewegung und des Rauschens im LSTM-Zustandsraum.

Verdichtung

In einem letzten Schritt verwendet das Modell eine vollständig verbundene Schicht, um die Endzustände $H_{i-s:i}$ auf den Ausgang $Y_{i-s:i}$ zu verdichten, der die Positionen $\hat{p}_{i-s:i}$ des Senders enthält, siehe Abbildung 4.6. Entsprechend der Anzahl j der Gruppen g im Stapel G gibt

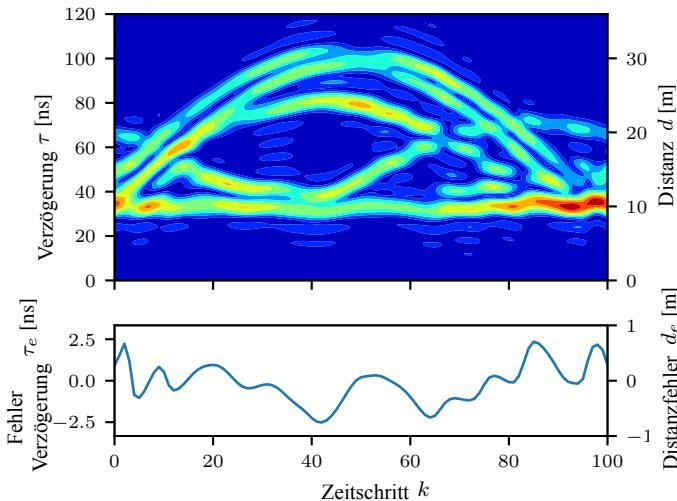


Abbildung 4.7: Beispielhafte ToA-Drift in einer Mehrwegeumgebung. Die Kanalimpulsantwort entwickelt sich im Laufe der Zeit. Eine allmähliche Änderung der Verzögerung und der Phasen führt zu driftenden ToF-Schätzungen. ToF-Fehler im kurzfristigen Kontext driften und sind fehlerhaft. Daher dürfen ToF-Fehler nicht durch eine mittelwertfreie PDF charakterisiert werden, wie dies von KF gefordert wird. QuaDRiGa positioniert die Störer zufällig.

die dichte Schicht die gleiche Anzahl j der Ergebnisse r , hier Positionen mit x -, y -, z -Koordinaten, im resultierenden Stapel R zurück, mit

$$R_{X_{i-s}} = \{r_{0,X_{i-s}}, r_{1,X_{i-s}}, \dots, r_{j,X_{i-s}}\}. \quad (4.7)$$

Nur das neueste (letzte, siehe R_{x_i} respektive Y_i rechts oben in Abbildung 4.6) Element $r_{j,X_{i-s}}$ wird als Positionsschätzung für \hat{p}_i verwendet.

4.6 Experiment

Die Experimente vergleichen ein optimiertes KF-Modell aus dem Stand der Technik mit dem LSTM-Modell aus Abschnitt 4.5.2 sowohl für synthetische (Abschnitt 4.6.1) als auch für reale Datensätze (Abschnitt 4.6.2) mit driftenden ToF-Messungen. Abschnitt 4.6.3 gibt die Parameterkonfiguration der KF- und LSTM-Modelle an.

4.6.1 Simulationsumgebung

Um die Signalausbreitungseffekte des Funkkanals angemessen zu modellieren, wird QuaDRiGa (QUAsi Deterministic Radlo channel GenerAtoR) [205] verwendet, da es Eigenschaften bietet, die für typische Ausbreitungsszenarien, mit Mehrwegeffekten, in Innenräumen gelten. Im Folgenden wird kurz das Verfahren der Kanalsimulation skizziert.

Aus LSP mit L dominanten Pfaden, der Verzögerungsspreizung DS und der Winkelspreizung (engl. angular spread) wird eine bandbreitenbegrenzte Kanalimpulsantwort konstruiert, um die ToF-Werte zu schätzen. Dazu werden die (1) Verzögerungen (engl. delay) τ_l generiert, (2) ihre Leistung (engl. power) P_l berechnet, (3) Ankunftsinkel ϕ_l generiert und (4) diese Werte zu Kanalkoeffizienten a_l kombiniert. Die wichtigsten Kanalparameter, die QuaDRiGa verwendet, sind: Anzahl der MPC $L=6$, Quadratisches Mittel der (engl. Root Mean Square, RMS)-DS $\sigma_\tau=10$ ns und der Rician-K-Faktor $kF=0.5$. Für die ToF-Schätzung wird eine Bandbreite von $B_w=80$ MHz verwendet, die der realen Funkumgebung [158] gleicht, siehe Abschnitt 4.6.2.

Abbildung 4.7 zeigt, wie sich ein typischer Funkkanal im Innenraum im Laufe der Zeit entwickelt, siehe auch Abbildung 4.2, wenn sich ein mobiler Sender entlang eines Kreises mit einem Radius von 10 m, d. h. Pfadlänge bei direkter Sichtverbindung zwischen Sender und Empfänger, bewegt und ein Empfänger in der Mitte des Kreises platziert wird. Während es Teile des Signals gibt, die mit einer Verzögerung von 30 ns (=9 m) ankommen, gibt es auch einen signifikanten Teil, der eine höhere Verzögerung (bis zu 120 ns, d. h. 36 m) aufweist. Diese Verzögerung ändert sich auch, während sich der Sender durch die Mehrwegeinstellung bewegt. Im unteren Teil von Abbildung 4.7 sind die resultierenden ToF-Fehler dargestellt. Offensichtlich führt die Drift zu einer sich langsam ändernden Drift der ToF-Werte. Abbildung 4.7 zeigt eine zufällige (stochastische) Irrfahrt der ToF-Werte.

Synthetische Datensätze

Das Simulationsframework generiert zwei synthetische Datensätze: D_{sd} mit sinusförmigen Bewegungsstrajektorien des Senders und D_{rd} mit zufällig geformten Bewegungsstrajektorien des Senders. beide Datensätze werden mit einer verrauschten Gaußschen Verteilung erzeugt (*mean*

$\mu=0.0$ und einer Standardabweichung $\sigma=0.1$), d. h. jede Trajektorie ist einzigartig. Im Datensatz D_{sd} folgt ein Sender einem Sinuswellenlauf:

$$A \cdot \sin\left(2\pi \frac{n}{w} + \Phi\right) + D, \quad (4.8)$$

mit Amplitude $A=1$, eine Datenlänge von $n=10.000$ Abtastwerte, eine Winkelfrequenz von 2π , eine Wellenlänge von $w=1.000$ Abtastwerten, eine Phase von $\Phi=0$ und eine nicht nullzentrierte Amplitude $D=3$. Referenz in Abbildung 4.12(a) zeigt eine exemplarische Trajektorie von D_{sd} . QuaDRiGa erzeugt einen Datenstrom von ToF-Werten eines Senders, der sich entlang einer Sinuskurve mit einer (idealen) statischen ToF von 3 ns, d. h. mit 1 m Abstand zu einem Empfänger bewegt. Stattdessen lässt D_{rd} den Sender eine zufälligen Trajektorie folgen, die kumulative Summe von $n=10.000$ variiert zufällig mit der Skala $\sigma=0.1$, die auf [2; 4] in m beschränkt wird.

Neben dem Gaußschen Rauschen wird beiden Datensätzen eine langfristige Signaldrift hinzugefügt. Dem Datensatz D_{sd} wird ein langfristiges sinusförmiges Rauschen hinzugefügt, indem die Amplitude $A=2$ verschoben wird und die anderen Parameter gleich bleiben, siehe 'Mess. Verz.' in Abbildung 4.12(a). Dem Datensatz D_{rd} wird ein langfristiges Rauschen in Form einer zufälligen (stochastischen) Irrfahrt hinzugefügt, indem die Grenzen der Parameter [1; 5] in m verändert werden, siehe Mess. Verz. in Abbildung 4.13(a).

QuaDRiGa generiert jeweils 100 synthetische Trajektorien für D_{sd} und D_{rd} mit jeweils 10.000 Positionen ($=100 \cdot 10.000 = 1.000.000$ Positionen pro Datensatz). Diese Eingangsdaten enthalten 1D-ToF-Messungen und ihre entsprechende Position, d. h. die Entfernung des Senders zum Empfänger über die Zeit. Die beiden synthetischen Datensätze umfassen 200 Trajektorien t , von denen jede durch einen Strom von ToF-Messungen d von $n=1$ Empfängern und den entsprechenden Referenzpositionen p dargestellt wird mit einer maximalen Geschwindigkeit von 60 km/h und einer maximalen Beschleunigung von 6 m/s². Die beiden Datensätze werden zufällig in einen Trainingsdatensatz (70%), Validierungsdatensatz (10%) und einen Testdatensatz (20%) aufgeteilt. Zu beachten ist, dass die Aufteilung für alle Experimente für eine leichte Vergleichbarkeit zwischen einzelnen Modellen fixiert wird.

4 Positionsschätzung

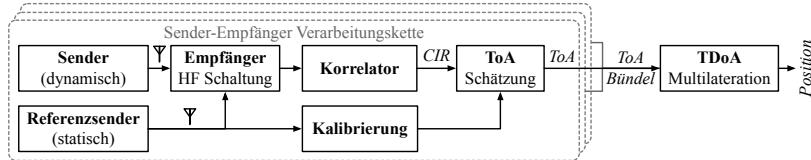


Abbildung 4.8: Verarbeitungskette des RedFIR Funklokalisierungssystems.

4.6.2 Reale Umgebung

Dieser Abschnitt beschreibt die Messinfrastruktur 4.6.2 der realen Umgebung und die Datensätze 4.6.2, die in typischen realistischen Anwendungen erzeugt wurden.

Messinfrastruktur

Die Daten wurden im Fraunhofer IIS L.I.N.K. Testzentrum (Lokalisierung, Identifizierung, Navigation, Kommunikation) in Nürnberg gesammelt, das ein einzigartiges Testgelände auf 1.400 m² bietet. Abbildung 4.11 zeigt den realen Innenraumbereich des L.I.N.K. Testzentrums. Die folgenden Mess- und Referenz-Plattformen werden verwendet, um die Trainingsdaten zu sammeln:

Funksystem. Für die Experimente wird ein funkbasierter Ortungssystem benötigt, das ToAs bereitstellt. Die ToA-Daten werden mit der innovativen funkbasierten Ortungstechnologie des Fraunhofer IIS (RedFIR) [158, 307] generiert, das im weltweit lizenzenfreien ISM-Band (industriell, wissenschaftlich und medizinisch) von 2.4 GHz läuft und eine Bandbreite von ca. 80 MHz verwendet [307]. Miniaturisierte Sender nutzen die verfügbare Bandbreite, um kurze Breitbandsignal-Bursts zusammen mit Identifikationssequenzen zu erzeugen, mit denen auf den Antenneneinheiten korreliert wird, um CIRs zu erhalten. Aus diesen CIRs werden mithilfe der IFP Methode die ToA-Werte extrahiert. Die IFP Methode wird in Abschnitt 3.3.4 detailliert beschrieben. Sätze von 12 ToF-Werten, bzw. ToA, werden von RedFIR mit 12 zeitsynchronisierten Empfängern an festen Orten bereitgestellt.

Abbildung 4.8 zeigt die Signalverarbeitungskette. Die Informationen fließen von links nach rechts. Das System unterscheidet feste Referenzsender von beweglichen Sendern und nutzt die Referenzsender für Kalibrierungszwecke. Die Uhren der Empfänger starten

zu unterschiedlichen Zeitpunkten und müssen zu Beginn und fortwährend zueinander synchronisiert werden. Alle Sender senden Signale aus, die von den Empfangsantennen empfangen werden. Die Installation verwendet 12 Antennen, die Signale von bis zu 144 verschiedenen Sendern empfangen. Mobile Sender senden bis zu 2.000 Signale pro Sekunde aus. In den Studien werden Sender mit 200 Signale pro Sekunde verwendet. Das Ortungssystem ermöglicht den Empfang von 50.000 dieser Signalbursts pro Sekunde. Für jede der 12 Empfängerleitungen korrelieren FPGAs die Signal-Sequenzen, um die Korrelation, die Kanalimpulsantwort, zu erhalten. Da die Empfänger synchronisiert sind, haben alle extrahierten ToAs eine gemeinsame Zeitbasis. Die synchronisierten ToA-Bündel können unter Verwendung des TDoA-Verfahrens in Positionen multilateriert werden.

Referenzsystem. Die korrespondierenden Referenzen (engl. labels), bzw. die Referenzpositionen und Referenztrajektorien werden mit einem submillimetergenauen optischen laserbasierten Positionsbestimmungssystem für Innenräume von Nikon- (engl. Indoor GPS, iGPS-)System aufgezeichnet. iGPS stellt Positionen mit 30 Hz bereit mit einem mittleren durchschnittlichen Fehler sowohl vertikal als auch horizontal unter 1 mm. Die beiden Systeme RedFIR und iGPS sind über einen gemeinsamen NTP Server zeitlich synchronisiert. Zur Synchronisation werden in einem Datenvorverarbeitungsschritt bei Bedarf Zwischenpositionen, 200 Hz RedFIR gegenüber 30 Hz iGPS, linear interpoliert. Das optische Referenzsystem wurde für alle Experimente verwendet.

Positioniersystem. Das Positionierungssystem ist ein kranähnliches Gerät, das sich jeder 3D-Position im Bereich mit einer Wiederholungsgenauigkeit von <2 mm bei einer Höchstgeschwindigkeit von 11 km/h nähert. Abbildung 4.9(a) zeigt den Kran beim Durchgang durch eine Konstruktion aus Absorberwänden. Das Positionierungssystem wurde verwendet, um einen deterministischen Datensatz, mit homogen verteilten Positionen, zu erfassen, der einen größeren Bereich in fester Höhe abdeckt. Am Positionierungssystem werden die RF-Sender und die iGPS-Sender angebracht. Da die Aktualisierungsrate von iGPS niedriger ist als die von RLTS, werden die Referenzpositionen interpoliert. Die Mäander- und Rechteckexperimente verwenden das Fraunhofer-Positionierungssystem, um deterministische Trajektorien



(a) Positioniersystem.

(b) Mobiler Roboter.

(c) Mensch.

Abbildung 4.9: Plattformen zur Generierung von Trainingsdaten.

zu erstellen.

Mobiler Roboter. Der Fraunhofer Segway RMP- 210, siehe Abbildung 4.9(b), erfasst hochdynamische Bewegungsdaten mit einer Höchstgeschwindigkeit von 30 km/h und einer Beschleunigung von 2.0 m/s². Als Referenz werden die interpolierten iGPS-Positionen verwendet. Das Zickzackexperiment verwendet das Fraunhofer-Segway Positionierungssystem, um deterministische Trajektorien zu erstellen.

Mensch. Mehrere iGPS-Sender werden verwendet, um die Position eines mobilen RF-Senders zu bestimmen, der sich in der Nähe des Halses der Person befindet [125]. Die Experimente Spaziergang und Gitter verwenden einen am Körper montierten Apparat, siehe Abbildung 4.9(c), der Bewegungen von Personen erfasst. Die Geschwindigkeiten liegen unter 24 km/h.

Datensätze

Im Rahmen der Experimente wurden fünf reale Datensätze aufgezeichnet: Mäander, Zickzack, Spaziergang, Gitter und Rechtecke. Abbildung 4.10 zeigt exemplarische Messverläufe und Trajektorien. Die Tabelle 4.1 listet die Konfigurationen der Datensätze auf. Die Mess- und Referenzplattformen folgen der Trajektorie und zeichnen 200 Korrelationssignale, bzw. ToAs, pro Antenne und Sekunde auf. Da das iGPS-System die Referenzpositionen nur mit 30 Hz liefert, werden sowohl die Referenzpositionen als auch die Zeitstempel interpoliert.

Aus der Gesamtzahl der aufgezeichneten Trainingsmuster werden nur vollständige Sätze an ToAs ausgewählt, wenn alle 12 Antennen

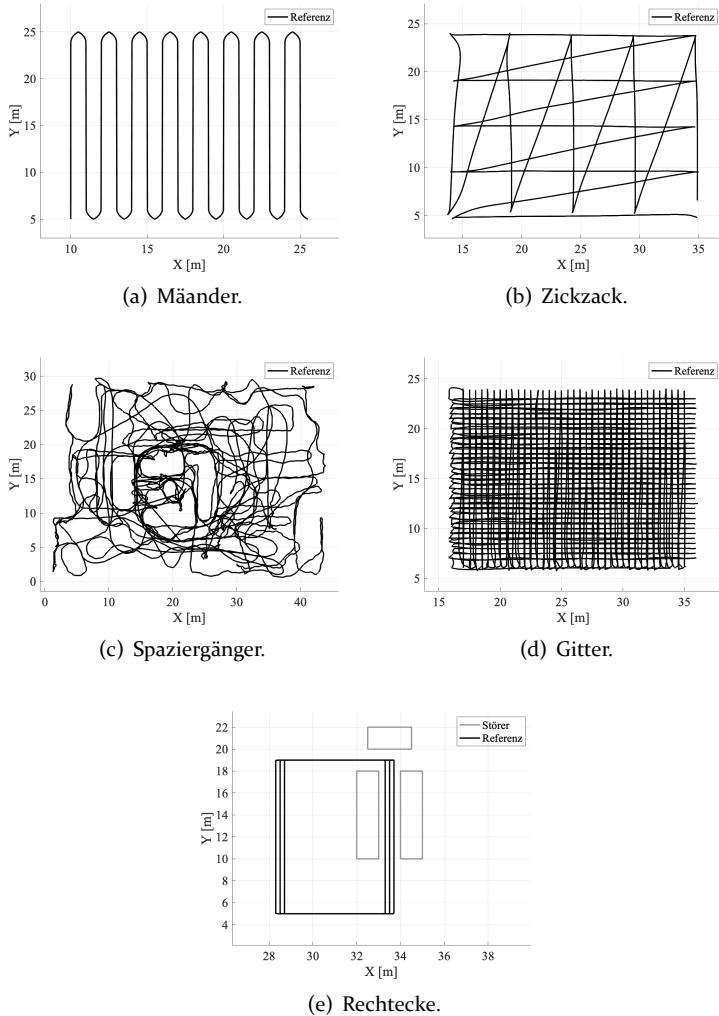


Abbildung 4.10: Datensätze, die für die Experimente in realen Umgebungen verwendet werden: Mäander wurde mit dem Fraunhofer-Positioniersystem aufgezeichnet; Zickzack wurde mit dem Segway RMP-210 aufgenommen; Spaziergang und Gitter wurden mit einem Menschen aufgenommen, der sich im Testzentrum frei bewegt; Rechtecke wurde mit dem Fraunhofer-Positioniersystem aufgezeichnet, siehe auch Abbildung 4.9.

4 Positionsschätzung

Tabelle 4.1: Beschreibung der Datensätze (ein ToA Bündel besteht aus zeitlich synchronisierten ToA Werten von 12 Empfängern).

Datensatz	ToA Bündel [#]	Lokalisierungs- Bereich (w×h) [m]	Höhe [m]	Plattform
Mäander	254.876	13.0 × 20.0	2.50	Positioniersystem
Zickzack	304.120	22.0 × 19.0	0.53	Segway
Spaziergänger	404.595	45.0 × 30.0	0.96 - 2.01	Mensch
Rechtecke	92.724	5.0 × 14.0	2.80	Positioniersystem
Gitter	283.529	21.0 × 20.0	1.49 - 1.55	Mensch

Korrelationen und ToAs liefern. In einigen Fällen sind Korrelationen verfälscht oder es wurden niedrige Signal-Rausch-Verhältnisse festgestellt. Solche Messungen werden verworfen. Für alle Plattformen, außer Mensch, wurde die z -Koordinate, die Höhe festgelegt, da die sub-optimale Geometrie der HF-Antennen in der Testumgebung die z -Koordinate nicht genau auflösen kann.² Allerdings konnte die Höhe der Sender für den zufälligen menschlichen Gang in den Datensätzen Spaziergang ($SD=0.78$ m) und Gitter ($SD=0.03$ m) nicht festgelegt werden daher sind dort unterschiedliche Höhen eingebettet.

Datenvorverarbeitung

Bewegungstrajektorien werden mit der jeweiligen Plattform (siehe Tabelle 4.1), bspw. für Zickzack mit einem ferngesteuerten mobilen Roboter (Segway RMP-210) generiert. Für den Zickzack Datensatz folgt der Roboter 220 Mal einer vordefinierten Zickzack-Trajektorie (siehe Abbildung 4.15(a)) mit einer maximalen Geschwindigkeit von 12 km/h und einer maximalen Beschleunigung von 2 m/s². Im Gegensatz dazu verfolgen die jeweiligen Plattformen die Trajektorien der anderen Datensätze nur ein Mal. Der resultierende Zickzack Datensatz D_{rw} enthält etwa 504 Minuten bereinigte Bewegungsdaten (= insgesamt 302.400 Positionen und 3.628.800 ToA-Werte = 12 · Anzahl der Positionen), bestehend aus 220 Trajektorien t (durchschnittliche Dauer von 2.29 min) von denen jede durch einen Strom von ToF-Messungen $d=16.495$ von

²Die Antennen decken nur zwei vertikale Ebenen ab, d. h. es stehen nur zwei Antennen für die Bestimmung der Höhe zur Verfügung, sodass bereits kleinste Fehler zu niedrigen Positionsgenauigkeiten führen.

$n=12$ Empfängern und den entsprechenden Referenzpositionen $p=1.375$ dargestellt wird. Positionen mit fehlerhaften ToA-Sätzen werden verworfen. Die Konfiguration ähnelt der Zickzack-Konfiguration. Für eine einfache Übersicht werden die Konfigurationen der anderen Datensätze in Tabelle 4.1 aufgelistet. Für alle Datensätze werden die einzelnen Trajektorien auf dieselbe Länge geschnitten, d. h. Überhänge vor dem Start und nach dem Ende werden so verworfen, dass alle t dieselbe Anzahl an Positionen und vollständigen ToA-Sätzen fasst.

Die Datensätze der realen Umgebung werden zufällig in einen Trainingsdatensatz (70%), Validierungsdatensatz (10%) und einen Testdatensatz (20%) aufgeteilt. Ein fester Startwert für den Pseudozufallszahlengenerator stellt sicher, dass bei jeder Generierung des Datensatzes dieselben Stichproben bereitstehen. Zu beachten ist, dass diese Aufteilung für alle Experimente für eine leichte Vergleichbarkeit zwischen einzelnen Modellen fixiert wird.

4.6.3 Parametrisierung der Verfahren

In diesem Abschnitt werden die Parameterkonfigurationen für das KF (Abschnitt 4.6.3) und das LSTM (Abschnitt 4.6.3) Verfahren beschrieben, die in Abschnitt 4.7 zur Bewertung verwendet werden.

Basisverfahren - Kalman-Filter, KF

Es wird ein lineares KF-Modell aus dem Stand der Technik verwendet. Diese Modelle gelten unter der Annahme driftfreier Signale als optimale Zustandsschätzer [88]. Dieses Bayessche Modell beschreibt die Entwicklung des Zustands x und dessen Unsicherheit P mit der Bewegungsübergangsfunktion $F(t)$, das Messrauschen R und Prozessrauschen Q , als lineare Funktionen, die durch Gaußsches Rauschen beeinflusst werden.

Zunächst wird KF mit einem Startzustand $x_0=0$, einer Kovarianz $P=1$, Prozessrauschen $Q=0.1$, Messrauschen $R=\sigma=0.1$ und der Übergangsfunktion $F(t)$ parametrisiert, die eine konstante Geschwindigkeit (engl. constant velocity) modelliert. Da empirische Kenntnisse über die Daten vorhanden sind, m.a.W. der Trainingsdatensatz, wird das KF optimal, bspw. mit $x_0=0$ und $R=\sigma=0.1$ für D_{sd} ohne langfristiges Rauschen, auf den jeweiligen Trainingsdatensatz konfiguriert. So passt die jeweilige

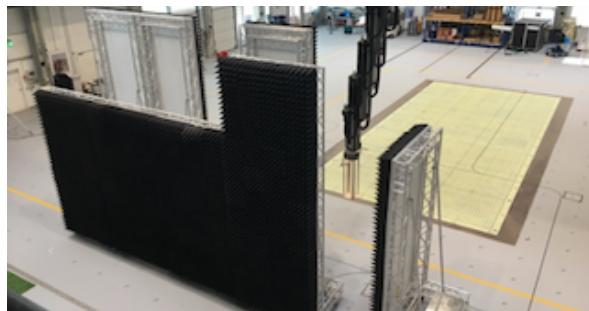
4 Positionsschätzung



(a) Draufsicht (Umgebung ohne Signalstörer).



(b) Seitenansicht des Spaziergang Datensatzes (Umgebung mit Signalstörer; Mensch mit Messplattform auf den Schultern).



(c) Seitenansicht des Rechtecke Datensatzes (Umgebung mit Signalstörer; Positioniersystem mit Messplattform).

Abbildung 4.11: Reale Studienumgebungen: $40\text{m} \times 40\text{ m}$ Innenraum mit RF-Verfolgungsbereich und optischem Referenzsystem.

Tabelle 4.2: Ergebnisse der Rastersuche nach optimalen LSTM-Parametern.

Parameter	Suchraum	Beste Konfiguration
Stapelgröße	[1, 2, 5, 10, 20, 50, 100, 200]	100
Epochen	[10, 50, 100, 1000, 2000]	1000
Tiefe/Schichten	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20]	4
Sequenzlänge	[1, 2, 5, 10, 20, 50, 100, 200]	20
Optimierer	[SGD, RMSprop, Adagrad, Adam]	Adam
Verwerfungsrate	[0.0, 0.0001, 0.001, 0.01, 0.1]	0.0
Breite/Zellgröße	[1, 8, 16, 32, 64, 128, 256, 512]	256
Gucklöcher (engl. peepholes)	[An, Aus]	Aus
Lernrate	[0.00001, 0.0001, 0.001, 0.01, 0.1, 0.2, 0.3]	0.0001
Aktivierungsfunktion	[linear, softmax, Tanh, ReLu, sigmoid]	Tanh
Gewichtsinitialisierung	[uniform, normal, Null]	normal

Konfiguration optimal zu den Eingabedaten, und das jeweilige KF Modell ist dann ein perfekter Optimierer für den jeweiligen Datensatz.

LSTM-Verfahren, LSTM

Da das LSTM-Modell nichtlineare Aktivierungsfunktionen einsetzt, kann dieses Modell ein Kandidat sein, der alle Datensätze perfekt modelliert, d. h. D_{sd} , D_{rd} mit und ohne langfristige Drift und D_{rw} . Mit einer Rastersuche [357] wird eine optimale Modellarchitektur entwickelt, d. h. es werden optimale Modellparameter für jeden Datensatz gesucht: die Anzahl der Schichten, die Breite, die Stapelgröße, die Sequenzlänge, d. h. die Anzahl der Zeitschritte, Optimierer, die Menge von Verwerfungen pro Schicht und die Wirkung von Gucklöchern. Die optimale Konfiguration, die einen guten Kompromiss zwischen Rechenaufwand und Leistung darstellt, wird in Tabelle. 4.2 aufgelistet. Die folgende Parameterkonfiguration hat für alle Datensätze die besten Ergebnisse erzielt: Stapelgröße = 100 (entspricht der Anzahl der verschiedenen Trajektorien), Epochen ≥ 1000 , Schichten ≥ 4 , Sequenzlänge ≥ 20 , Adam-Optimierer, kein Verwerfen, Breite ≥ 256 , keine Gucklöcher, Lernrate ≤ 0.001 , Sigmoid und TanH als Aktivierung, einheitliche Gewichte und frühes Stoppen.

Im Gegensatz zu KF wird das LSTM-Modell nicht auf allen aufeinanderfolgenden Messungen einer Trajektorie trainiert, sondern auf zufällig aufeinanderfolgenden Teilstücken (Segmente) einer Trajektorie. Die Segmente werden mithilfe eines quasi-gleitenden Fensteransatzes über den Eingabedatensatz erzeugt. Für jeden Eingabeschritt X_i des LSTM Modells wird ein Eingabedatensatz erstellt, indem ein Segment mit

der Größe s aus der Trajektorie extrahiert wird. Dafür teilt ein gleitendes Fenster das Segment in Stapel unabhängiger Trajektoriengruppen auf. Daher repräsentieren jeder Stapel G und ihre jeweiligen Gruppen g_j einen Satz aufeinanderfolgender ToF-Werte und ihre entsprechende Position eines Segments einer Trajektorie. Zu beachten ist, dass jedes Segment (aus Sicht des LSTM-Modells wird von Sequenzen gesprochen), das aus einer Trajektorie extrahiert wird, bei unterschiedlichen Startversätzen beginnt. Somit wird garantiert, dass das LSTM-Modell völlig unabhängige Trajektorien lernt. Um die Gewichte des LSTM zu aktualisieren, wird der Verlust (engl. loss) berechnet, d. h. der Konvergenzfehler zwischen der vorhergesagten Verzögerung oder Position auf den ToF-Werten und der Referenzverzögerung oder Referenzposition. Dazu wird die Verlustfunktion für den MSE verwendet:

$$L_{MSE} = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2. \quad (4.9)$$

4.7 Auswertung

Abschnitt 4.7.1 beschreibt die Fehlermetriken, die zur Bewertung der Verfahren verwendet werden. Abschnitt 4.7.2 beschreibt die Ergebnisse der Verfahren auf den synthetischen Datensätzen und Abschnitt 4.7.3 beschreibt die Ergebnisse auf den realen Datensätzen.

4.7.1 Fehlermetriken

Um die Ergebnisse der Modelle KF und LSTM sowohl für die synthetischen Datensätze D_{sd} und D_{rd} , jeweils mit und ohne langfristige Drift, als auch für die realen Datensätze D_{rw} und weitere zu vergleichen werden Fehlermetriken gegen die Referenz berechnet. Dazu werden einzelne Sätze von ToA-Daten von den Modellen verarbeitet, um eine Verzögerung, für die synthetischen Daten, oder eine absolute Position, für die Daten der realen Welt, zu schätzen. Die Verzögerungen und Positionen werden dann mit der Referenzverzögerung oder der Referenzposition verglichen. Es werden Fehlermetriken zur Bewertung der Genauigkeit, nämlich MAE , CEP_{50} , CEP_{75} und CEP_{95} ³, und dem Umgang mit Ausreißern, nämlich MSE und $RMSE$, für beide Modelle und alle Datensätze bestimmt.

³Zirkuläre Fehlerwahrscheinlichkeit, engl. Circular Error Probability, CEP .

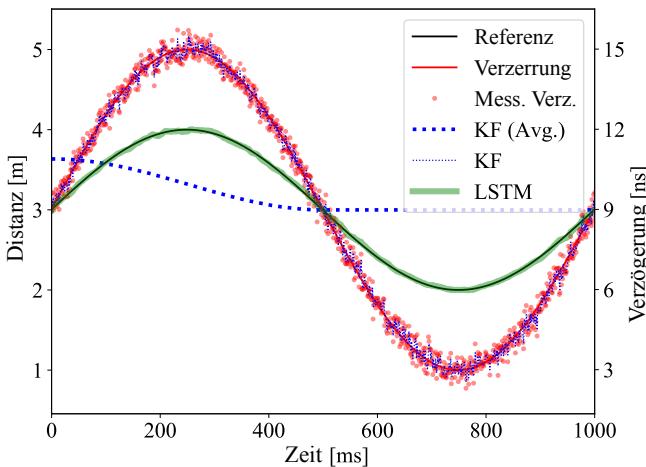
(a) Synthetischer sinusförmiger Datensatz D_{sd} , mit langfristiger Drift.

Abbildung 4.12: Ergebnisse der Schätzer KF und LSTM auf je einer exemplarischen Trajektorie des synthetischen sinusförmigen Datensatzes. Die Ergebnisse visualisieren die Referenz (schwarze Linie), ohne langfristige Drift (rote Linie), mit langfristiger Drift (rote Punkte), optimiertes KF (blaue Linie), optimiertes starres KF (blaue Punkte) und LSTM (grüne Linie).

4.7.2 Synthetische Datensätze

Die beiden synthetischen Datensätze D_{sd} und D_{rd} mit und ohne langfristige Drift sind eindimensional, d. h. sie bilden die Verzögerung in einer Dimension ab, und daher zeigen die Fehlermetriken die Genauigkeit der KF und LSTM Modelle nur in einer Dimension, nämlich Verzögerung oder Distanz.

Daten mit langfristiger, sinusförmiger Drift, D_{sd} . Abbildung 4.12(a) zeigt exemplarische 1000 ms der Ergebnisse für den synthetischen Datensatz mit der Sinusdrift D_{sd} . Die schwarze Linie zeigt die sinusförmige Referenz ohne Verzögerungsfehler. Die rote Linie zeigt die driftenden ToF-Messungen die durch additives weißes Gaußsches Rauschen (AWGN) beeinträchtigt werden. Die roten Punkte sind die driftenden ToF-Messungen, die einer langfristigen (sinusförmigen) Drift folgen, d. h. sie leiden unter AWGN und der langfristigen sinusförmigen Drift, die über dem sinusförmigen Signal liegt. Die blau

gepunktete Linie zeigt die Schätzungen des KF Modells. Während das KF das Gaußsche Rauschen glättet, folgt es genau der Drift, da dies den optimalen Schätz Zustand ergibt. Stattdessen folgt das optimierte starre KF (Avg.) seinem internen Zustand, glättet über 50 Messpunkte und liefert ebenfalls schlechter Ergebnisse. KF führt zu einem MAE von 2.13 ns, d. h. einem Abstandsmessfehler von circa 0.64 m. Da das LSTM (grüne linie) der Linie der Referenzwerte entspricht, übertrifft es nicht nur das hochoptimierte KF-Modell deutlich, sondern folgt auch den realen ToF-Werten sehr genau, was insgesamt zu einem MAE von nur 0.029 ns oder 0.00009 m führt.

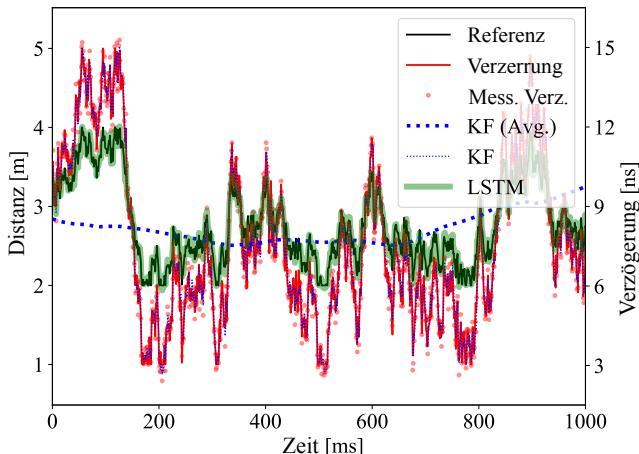
Obwohl das KF für die Trainingsdaten optimiert wurde, enthalten die Testdaten Kombinationen aus weißem und langfristigem Rauschen, die das KF nicht erkennen und verfolgen kann, da es nur die sehr grobe (kurzzeitige) Beziehung zwischen der aktuellen und der letzten Messung betrachtet und daher schlechtere Ergebnisse liefert. Im Gegensatz dazu berücksichtigt LSTM sowohl kurzzeitige als auch langzeitige Abhängigkeiten und kann die Langzeitänderungen, die sich über die Datensequenzen entwickeln, erkennen und verfolgen und liefert wesentlich genauere Schätzungen.

Daten mit langfristiger, zufälliger Irrfahrt Drift, D_{rd} . Abbildung 4.13(a) zeigt die Ergebnisse für den synthetischen Datensatz D_{rd} mit zufälligen Bewegungsformen und mit zufälliger (stochastischer) Irrfahrt Drift. Die Darstellung der Ergebnisse folgt der gleichen Farbcodierung wie in Abbildung 4.12(a). Wiederum folgt das KF der Drift in den Messungen, während es das zufällige Rauschen glättet (KF) oder die Messungen vollständig ignoriert (KF (Avg.)). Die KF-Schätzungen haben eine MAE von 1.72 ns oder 0.51 m. Wiederum approximiert das LSTM die realen ToF-Werte genau und ergibt einen viel kleineren MAE von 0.12 ns oder 0.036 m.

Die Ergebnisse der synthetischen Datensätze zeigen, dass das LSTM-Modell in beiden Szenarien das Referenzverfahren KF deutlich übertrifft und nahezu optimale Schätzungen liefert.

4.7.3 Reale Datensätze

Tabelle 4.3 listet die Ergebnisse für die Datensätze Mäander, Zickzack, Spaziergang, Gitter und Rechtecke für die Methoden KF und LSTM entlang der Fehlermetriken auf. Die wesentlichen Erkenntnisse



(a) Synthetischer Datensatz mit zufälliger Bewegungsform D_{rd} , mit langfristiger Drift.

Abbildung 4.13: Ergebnisse der Schätzer KF und LSTM auf einer exemplarischen Trajektorie des synthetisches Datensatzes mit zufälligen Bewegungsformen. Sie zeigen die Referenz (schwarze Linie), ohne langfristige Drift (rote Linie), mit langfristiger Drift (rote Punkte), optimiertes KF (blaue Linie), optimiertes starres KF (blaue Punkte) und LSTM (grüne Linie).

se der Methoden für die Datensätze werden im Anschluss diskutiert. Es ist zu beachten, dass die Ergebnisse das KF berichten, dass für jeden einzelnen Datensatz auf den Trainings- und Validierungsdaten optimiert wurde und somit das optimale Gleichgewicht zwischen KF und KF (Avg.) repräsentiert. Lediglich Abbildung 4.15 zeigt beispielhafte Ergebnisse für das optimierte starre KF (Avg.) und das optimierte dynamische (KF). Zu beachten ist, dass die Fehlermetriken diesmal zweidimensionale Ergebnisse repräsentieren, da die absoluten Positionen in der x- und y-Achse geschätzt werden.

Mäander Datensatz

Der Mäander-Datensatz bietet Trajektorien, die statische und ruhige Bewegungsformen mit langsamem Kurvenfahrten und wenigen Änderungen enthalten. Abbildung 4.14 visualisiert die Ergebnisse der Positionsschätzung. Das KF wird typischerweise unter solchen Rahmenbedingungen zur Lokalisierung verwendet. Dennoch, dieses Experiment zeigt, dass LSTM deutlich höhere Genauigkeiten ($MAE=0.037$, $MSE=0.003$ [m]) als KF ($MAE=0.559$, $MSE=1.141$ [m]) erreichen kann und somit eine sehr gute Alternative in leicht modellierbaren Bewegungen und Umgebungen darstellt.

Tabelle 4.3 führt die Ergebnisse für KF und LSTM entlang der Fehlermetriken für den Mäanderdatensatz auf. Die niedrigeren MSE- und RMSE-Werte des LSTM deuten darauf hin, dass die Fehlervarianz aufgrund des impliziten datengetriebenen Bewegungsmodells niedriger ist als die des KF. Ebenso könnte die etwas langsamere Einschwingzeit des KF die MSE und RMSE Werte erhöhen. Abbildung 4.14 zeigt die langsame Einschwingzeit des KF bei Koordinaten $x=21$ und $y=5$. Das KF benötigt deutlich länger nach dem Überschießen der Kurve, um zurück auf die korrekte Spur zu kehren. Die nichtdeterministischen Abweichungen des KF von der Referenz könnten auch darauf hindeuten, dass die ToA-Werte durch eine Störquelle drifteten, bspw. durch ein Objekt, das die Sender-Empfänger-Verbindung unterbricht und verzögert.

Zickzack Datensatz

Der reale Datensatz D_{rw} wird genutzt, um die Genauigkeit von KF und LSTM in stark dynamischen Situationen zu vergleichen. Die Höhe des zu verortenden Senders ist durchgehend konstant (0.75 m, mit Messungenauigkeiten $SD=\pm 0.001$ m) und wird deshalb vernachlässigt.

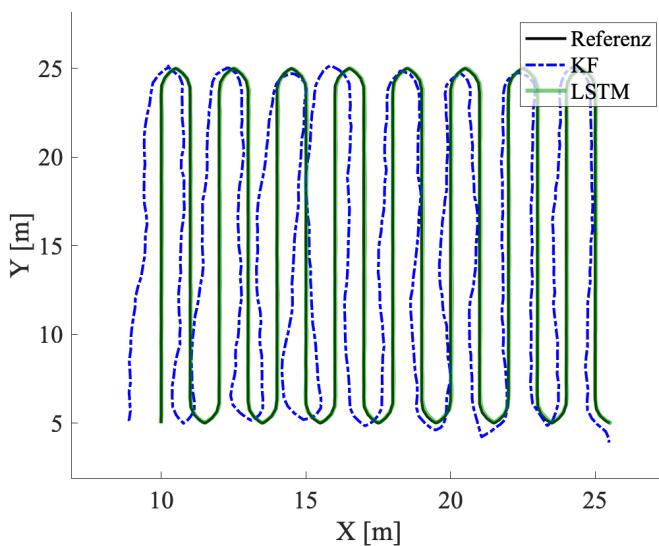


Abbildung 4.14: Ergebnisse der Schätzer KF und LSTM auf einer exemplarischen Trajektorie des realen Datensatzes Mäander. Sie zeigen die Referenz (schwarze Linie), optimiertes KF (blaue Linie) und LSTM (grüne Linie).

Abbildung 4.15(a) visualisiert die Ergebnisse der Positionsschätzung. Die Ergebnisse des realen Datensatzes werden analog zu den Ergebnissen der synthetischen Datensätze in Abbildungen 4.12(a) und 4.13(a) visualisiert. Allerdings sind die Drift der Messungen in der realen Umgebung nicht messbar, daher werden diese nicht dargestellt. Zu beachten ist, dass die Messung (rote Punkte) die Positionsschätzungen der Multilateration des RedFIR Systems darstellen. Ein Bancroft Optimierer [26] schätzt Positionen aus einer Reihe entsprechender ToFs.

Tabelle 4.3 listet die Ergebnisse für KF und LSTM entlang der Fehlermetriken für den Zickzack Datensatz auf. Insgesamt liefert das LSTM-Modell ($MAE=0.058\text{ m}$, $MSE=0.006\text{ m}$) deutlich genauere Positionen als das optimierte KF ($MAE=0.127\text{ m}$, $MSE=0.024\text{ m}$). Wiederum folgt das KF den driftenden ToF-Werten, während es sich um die Referenztrajektorie windet. Darüber hinaus überschießt das KF auch immer dann, wenn sich die Bewegung in den Ecken ändert, d. h. wenn sich der Roboter in eine andere Richtung dreht. Hier ist klar zu erkennen, dass selbst ein auf die zugrunde liegenden Daten hochoptimiertes KF dennoch große Schwierigkeiten hat mit sich schnell und abrupt ändernden Drehungen, bspw. bei Richtungswechseln, und Bewegungen, bspw. bei schnellem Abbremsen und Anfahren, umzugehen, wie sie auch in Sport- und VR-Anwendungen vorkommen. Hier dauert die Neukalibrierung, d. h. das Einschwingen, von KF sehr lange. Im Gegensatz dazu nähert sich das LSTM nicht nur der tatsächlichen Trajektorie auf den geraden Linien an, sondern funktioniert auch sehr gut in Situationen mit abrupten Bremsvorgängen und Richtungsänderungen in den Ecken.

Menschliche Bewegung

Die Spaziergang- und Gitterdatensätze bieten Trajektorien, die sowohl dynamische als auch ruhige Bewegungsformen mit vielen Richtungsänderungen enthalten. Abbildung 4.16 visualisiert die Ergebnisse der Methoden auf den Testdatensätzen der beiden Experimente.

Das Spaziergang-Experiment zeigt, dass LSTM in Situationen mit dynamischen und häufigen Höhen- und Richtungsänderungen signifikant höhere Genauigkeiten ($MAE=0.078$, $MSE=0.011\text{ [m]}$) als KF ($MAE=3.261$, $MSE=7.299\text{ [m]}$) erreicht und somit die Modellierung menschlicher Bewegungen über sehr lange Zeiträume mit hochkomplexen und dynamischen Bewegungsformen viel genauer als das KF ermöglicht. Dagegen liefern LSTM ($MAE=0.050$, $MSE=0.005\text{ [m]}$) und KF ($MAE=0.116$,

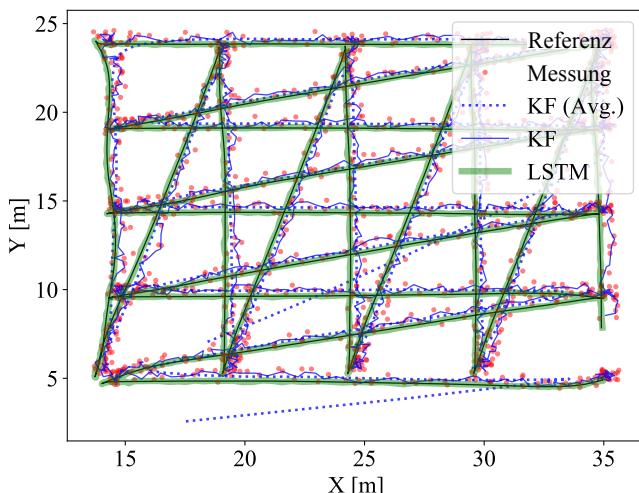
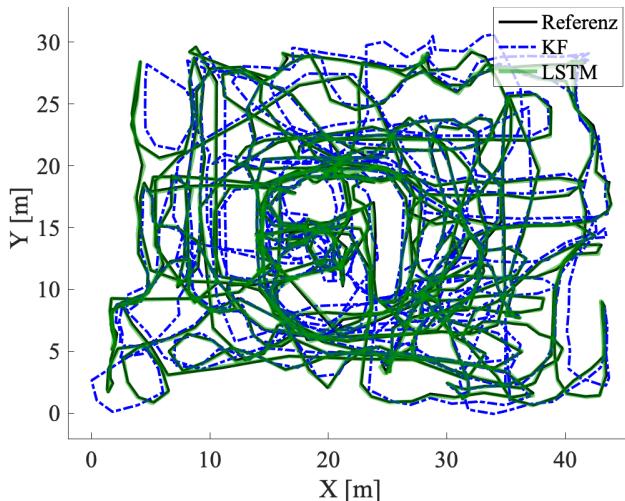
(a) Echtwelt Zickzack Datensatz D_{rw} .

Abbildung 4.15: Ergebnisse der Schätzer KF und LSTM auf je einer exemplarischen Trajektorie des realen Datensatzes. Die Ergebnisse des Datensatzes visualisieren die Referenz (schwarze Linie), ohne langfristige Drift (rote Linie), mit langfristiger Drift (rote Punkte), optimiertes KF (blaue Linie), optimiertes starres KF (blaue Punkte) und LSTM (grüne Linie).

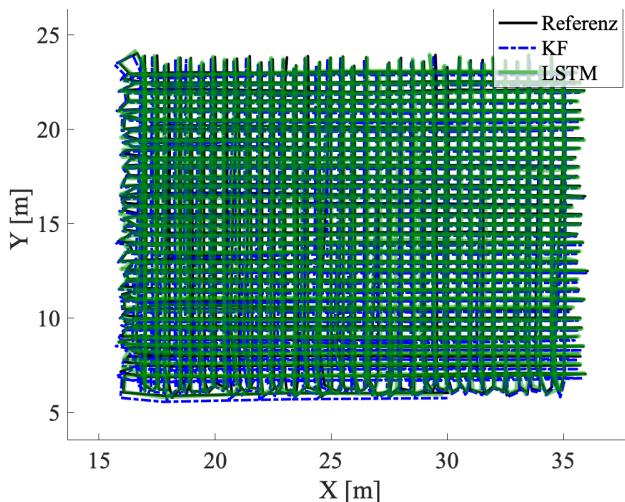
Tabelle 4.3: Ergebnisse in der realen Umgebung (Positionfehler in [m], je Datensatz werden die niedrigsten CEP_{95} Werte **fettgedruckt**).

Dataset	Methode	MAE	MSE	RMSE	CEP_{50}	CEP_{75}	CEP_{95}
Mäander	KF	0.559	1.141	1.068	0.266	0.521	3.104
	LSTM	0.037	0.003	0.052	0.029	0.058	0.108
Zickzack	KF	0.127	0.024	0.319	0.306	0.321	0.353
	LSTM	0.058	0.006	0.079	0.041	0.088	0.129
Spaziergang	KF	3.261	7.299	5.225	1.814	4.500	11.599
	LSTM	0.078	0.011	0.106	0.051	0.098	0.207
Gitter	KF	0.116	0.017	0.131	0.115	0.158	0.222
	LSTM	0.050	0.005	0.070	0.038	0.068	0.137
Rechtecke	KF	1.042	2.448	1.565	0.586	1.666	3.120
	LSTM	0.024	0.001	0.035	0.017	0.028	0.069

4 Positionsschätzung



(a) Spaziergänger.



(b) Gitter.

Abbildung 4.16: Ergebnisse der Schätzer KF und LSTM auf je einer exemplarischen Trajektorie der realen Datensätze Spaziergänger (oben) und Gitter (unten). Die Ergebnisse des Datensatzes visualisieren die Referenz (schwarze Linie), optimiertes KF (blaue Linie) und LSTM (grüne Linie).

$MSE=0.017$ [m]) hohe Genauigkeiten für das Gitter-Experiment mit ruhigen Bewegungen ohne Höhen- und Richtungsänderungen. Das unterstreicht die Erkenntnisse des Spaziergang-Experiments, dass sich dynamische Bewegungen mit einem datengetriebenen Ansatz genauer und zuverlässiger ermitteln lassen.

In der Tabelle 4.3 sind die Ergebnisse für KF und LSTM in den Fehlermetriken für beide Datensätze aufgeführt. Die niedrigeren MSE- und RMSE-Werte des LSTM zeigen, dass das Bewegungsmodell, das das LSTM gelernt hat, gegenüber Messfehlern des ToA wesentlich robuster und auch gegenüber zufälligen Richtungsänderungen wesentlich robuster ist. Die niedrigen MAE- und CEP-Werte des LSTM sind beeindruckend für Sequenzen mit solch einer Länge, der Spaziergang dauerte 34 min und Gitter dauerte 24 min. Die hohe Genauigkeit legt nahe, dass die Positionsinformationen keine besonders hohen Anforderungen an die Kapazität des Kontextvektors des LSTMs stellen. In Kapitel 7 wird gezeigt, dass bereits signifikant kürzere Sequenzen (2.56 s) von Trägheitssensormessungen, mit hoher Informationskomplexität, die Kapazität des Kontextvektors überladen.

Mehrwegeausbreitung

In der Realität gibt es viele Situationen, in denen das Signal gedämpft, blockiert oder verschlechtert wird. In Umgebungen der realen Welt werden häufig Objekte und Hindernisse gestreut, die eine Mehrwegeausbreitung und eine Blockierung des LoS-Signals verursachen. Um zu sehen, ob der datengetriebene Ansatz auch den Effekt von Mehrwege abschwächt, wird der Datensatz RECHTECKE verwendet. Abbildung 4.17 visualisiert die Ergebnisse Positionsschätzung. Die Rechtecke veranschaulichen die Trajektorie des Testdatensatzes. Abbildung 4.9 zeigt, wie die Absorberwände auf der rechten Seite des Datensatzes platziert werden. Der senkrechte Teil auf der rechten Seite der Rechtecke leidet stark unter Mehrwegeausbreitung und Hindernissen.

Generell gilt, dass die klassische ToA-Schätzung mit KF stark unter NLoS-Situationen leidet, siehe blaue Linie in Abbildung 4.17 rechts. Die klassische ToA-Schätzung mit KF liefert hohe Genauigkeiten auf der linken Seite in Abbildung 4.17 ($MAE=0.154$, $MSE=0.097$ [m]), leidet aber stark unter NLoS-Situationen auf der rechten Seite ($MAE=2.116$, $MSE=1.455$ [m]). Der LSTM-Ansatz mit seinem impliziten Bewegungsmodell liefert sowohl links im LoS-Bereich ($MAE=0.018$, $MSE=0.001$ [m]) als auch rechts im NLoS- Bereich ($MAE=0.029$, $MSE=0.002$ [m])

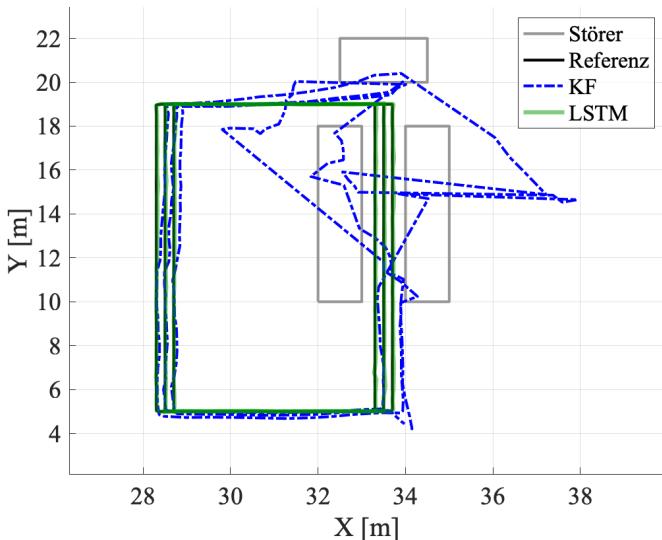


Abbildung 4.17: Ergebnisse der Schätzer (KF und LSTM) auf je einer exemplarischen Trajektorie des realen Datensatzes Rechtecke. Die Ergebnisse des Datensatzes visualisieren die Referenz (schwarze Linie), optimiertes KF (blaue Linie) und LSTM (grüne Linie). Die realen Störer (graue Rechtecke) sind in Abbildung 4.9(a)) dargestellt.

höchste Genauigkeiten im Durchschnitt (Gesamt-MAE=0.024, Gesamt-MSE=0.001 [m], siehe Tabelle 4.3). Möglicherweise dienen die MPCs in den Signalen des NLoS-Szenarios als quasi exogene Kontextinformation, bzw. virtuelle Anker oder Stationen, die von datengetriebenen Methoden implizit für eine genauere Positionsschätzung ausgenutzt werden könnten [497, 439, 4]. Dieses Experiment zeigt, dass der datengetriebene Ansatz das NLoS-Szenario effizient handhabt und im Gegensatz zu modellgetriebene Lokalisierungstechniken auch in dynamischen Bewegungssituationen ein hohes Maß an Genauigkeit liefert. In Zukunft lohnt es sich, Szenarien mit sich bewegenden Objekten zu betrachten, um zu sehen, wie sie die Genauigkeit der Positionsschätzung und die Langzeitstabilität des LSTM-Modells beeinflussen.

Zusammenfassung

Die Ergebnisse der Experimente (siehe Tabelle 4.3) in realen Umgebungen zeigen, dass LSTM bei allen Datensätzen: Mäander, Zackzack, Spaziergang, Gitter und Rechtecke höhere Genauigkeiten als hochoptimierte KF erzielt. Selbst bei hohen Geschwindigkeiten und starker Bewegungsdynamik im Zackzack-Datensatz kann das Modell die Position außergewöhnlich gut abschätzen ($CEP_{95} = 0.129 \text{ m}$). Die Genauigkeiten sind selbst für die Spaziergang- und Gitterdatensätze sehr gut ($CEP_{95} = 0.207$ und $CEP_{95} = 0.137$ in [m]), obwohl die Höhe des Funksenders, der auf den Schultern eines Menschen platziert wurde, stark variiert und die Bewegungsbahnen sehr lange Zeitabhängigkeiten aufweisen. Das LSTM bietet den größten Mehrwert bei starker Mehrwegausbreitung im Rechtecke-Datensatz. Das implizite Bewegungsmodell des LSTM und seine Fähigkeit, die Signalinterferenz zu entstören, liefern eine hervorragende Positionsgenauigkeit ($CEP_{95} = 0.069 \text{ m}$). Und obwohl der horizontale Abstand zwischen den Rechtecken im Rechteck-Datensatz nur 0.2 m beträgt, kann LSTM dennoch die Positionen der Rechtecke trennen und gut schätzen ($\text{RMSE}=0.035 \text{ m}$).

4.8 Diskussion

Es ist wahrscheinlich, dass das LSTM-Modell Zeitabhängigkeiten aufgrund der Tiefe (Anzahl der verborgenen Schichten), der Breite (Anzahl der Zellen, Neuronen und Gewichte) und der Größe des Zeitabschnitts, den es berücksichtigt, außergewöhnlich gut verarbeiten kann. Je tiefer das Netz ist, desto mehr Zeitabhängigkeiten und damit Bewegungsableitungen wie Beschleunigung, Geschwindigkeit, Verschiebung, Orientierung und Bewegungsabsicht können damit behandelt werden.

Während der Experimente wird festgestellt, dass flacheren Modelle, mit weniger Schichten, das langfristige hochkomplexe Rauschen schlechter lernen, d. h. das Rauschen von der Bewegung schlechter trennen können und somit schlechtere Ergebnisse liefern als vergleichbare tiefere Modelle und verallgemeinern das Bewegungsverhalten schlechter, d. h. Kurvenfahrten waren deutlich ungenauer. Flache Modelle bilden die ToF-Eingabedaten lediglich auf die Ausgabedaten (Verzögerungen oder Positionen) ab, ohne tiefere Zwischenbeziehungen, bspw. Beschleunigung, Geschwindigkeit und Richtung, zusammenhängender Messungen zu berücksichtigen. Flache Modelle trennen auch das langfristige Rauschen nicht von den Bewegungs-

informationen. Im Gegensatz dazu schätzt die hier vorgeschlagene optimierte tiefe LSTM-Architektur erfolgreich rechteckige, sinusförmige, elliptische und Trajektorien mit zufälligen Bewegungsformen.

In Vorstudien werden die Grenzen der Methode in realen Anwendungen genauer untersucht. Abbildung 4.18 zeigt vier Bildausschnitte (Zeitpunkte (1) bis (4) oben rechts im Bild) der 3D-Visualisierungsoberfläche, die die Positionsgenauigkeit eines KF und LSTM einer realen Modelleisenbahn in einer Funkumgebung mit und ohne Störung (LoS und NLoS) zeigen. In realistischen Anwendungen liefert dieselbe LSTM Architektur wesentlich präzisere und zuverlässigere Positionen als ein hochoptimierter KF. Das datengetriebene Verfahren liefert besonders außergewöhnlich genaue Positionen in Situationen mit hochdynamischen Bewegungen und menschlichen Bewegungen mit zufälligen Bewegungsänderungen. Darüber hinaus war das LSTM-Modell auch in der Lage, die reale, stark gestörte Signalausbreitung zu verarbeiten und die genauen Positionen nahezu unverändert zu schätzen, alles mit nur einem Bündel von Ankunftszeiten.

4.9 Zusammenfassung

Das vorgestellte RNN mit einem gestapelten LSTM-Modell kann (nach dem Training) optimale aufeinanderfolgende Positionen aus rohen und driftenden ToF-Messungen vorhersagen, die von Ortungssystemen mit geringer Genauigkeit bereitgestellt werden.

Während gezeigt wurde, dass das LSTM-Modell lernt, mit zeitabhängigen Fehlern wie Mehrwegeausbreitung und langfristiger zufällige (stochastische) Irrfahrt Drift umzugehen, werden auch neue Schritte zum Verständnis, zur Interpretation und zum Kontextlernen in einem einzigen Positionsschätzungsmodell präsentiert.

Das vorgestellte LSTM übertrifft sowohl bei synthetisch erzeugten als auch bei realen Trajektorien die driftende ToF-Messfehler enthalten ein optimiertes KF aus dem Stand der Technik.

Die Ergebnisse der Positionsschätzung fließen in die Orientierungsschätzung (Kapitel 6) ein, um mit Hilfe von zwei aufeinanderfolgenden Positionen einen Richtungsvektor zu bestimmen, der zur Kalibrierung der Orientierung verwendet wird. Zusätzlich wird die Position verwendet, um die rekonstruierte Trajektorie zu kalibrieren (Kapitel 7).

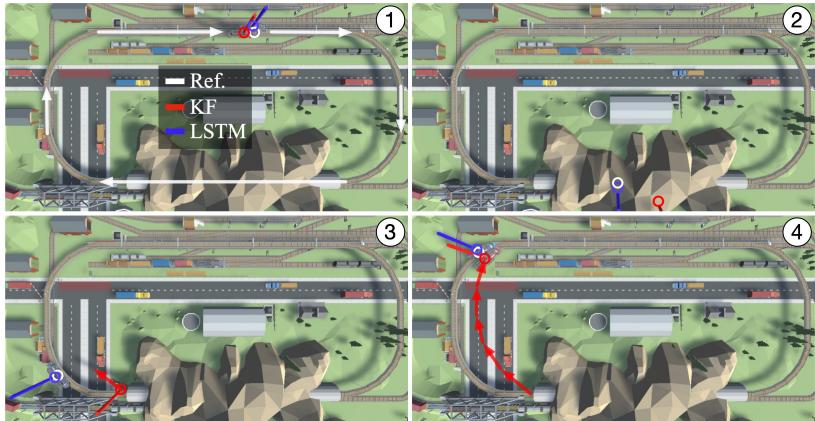


Abbildung 4.18: (Zeit)Ausschnitte (1) bis (4) einer 3D-Visualisierungsoberfläche, die die Positionsgenauigkeit einer realen Modelleisenbahn in einer Funkumgebung (15×5 m) mit und ohne Störung (LoS und NLoS) zeigen. Ein Funk- und ein Referenzsensor werden auf der Modelleisenbahn platziert und gemäß Abschnitt 7.6 verarbeitet. Ein realer Tunnel wird so vorbereitet, dass er die Funksignale erheblich streut und dämpft. Die reale Eisenbahn (Ref., weißer Kringel) fährt mit quasi konstanter Geschwindigkeit entlang der Referenzbahn (weiße Pfeile in (1)). Im Tunnel (untere Mitte jedes Bildes) verliert die reale Eisenbahn (der Funksensor) immer mehr die Sichtverbindung zu den Empfängern, sodass der virtuelle Zug, der mit einem optimierten KF (roter Zug und Kringel) positioniert ist, zunehmend unsicherer wird, falsche Positionen schätzt und sich nur noch verzögert bewegt, siehe die Distanz zwischen dem weißen und roten Kringel in (2) und (3). Im Gegensatz dazu bewegt sich der virtuelle Zug, der mit einem LSTM (blauer Zug und Kringel) positioniert ist, fast identisch zum Referenzzug, sowohl außerhalb (LoS) als auch innerhalb (NLoS) des Tunnels. Sobald der reale Zug den Tunnel verlässt, die Sichtverbindung zu vielen Empfängern ist wieder verfügbar, liefert auch das KF wieder Positionen, die aufgrund der hohen Unsicherheit des KF jedoch stark verzögert werden. Ein verzögertes Einschwingen des KF ist zwischen den Ausschnitten (3) und (4) zu erkennen: Es dauert > 4 s, bis das KF die gültige Bahnlinie erneut schätzen kann, in (4). Im Gegensatz dazu schätzt das LSTM nahezu fehlerfreie Positionen. Während das KF selbst bei konstanten Geschwindigkeiten keine plausible Position innerhalb des Tunnels schätzen kann, kann das LSTM selbst bei unterschiedlichen Geschwindigkeiten und Beschleunigungen oder bei vollständigem Stillstand präzise und zuverlässige Positionen schätzen. Dies liegt daran, dass das LSTM die Störungen im Tunnel als zusätzliche Positionsinformationen verwenden kann, während das KF keine physikalisch korrekten Beziehungen aus den stark verzögerten Ankunftszeiten (Entfernung) extrahieren kann.

5 Geschwindigkeitsschätzung

Einordnung

Abbildung 5.1 zeigt die Zuordnung der Geschwindigkeitsschätzung in der übergeordneten Verarbeitungskette der Dissertation. Aus verrauschten Messungen eines lose platzierten Trägheitssensors schätzt die Methode ressourcenschonend und hoch genaue Geschwindigkeiten der menschlichen Bewegung in dynamischen Anwendungen. Dieses Kapitel baut auf veröffentlichten Ergebnissen [131, 128] auf.

Kurzzusammenfassung

Fußgänger Koppelnavigation (engl. Pedestrian Dead Reckoning, PDR) verwendet Trägheitsmesseinheiten (engl. Inertial Measurement Units, IMU), um Schätzungen der aktuellen Geschwindigkeit und Ausrichtung eines Objekts zu kombinieren, um seine aktuelle Position zu bestimmen. Während klassische PDR Ansätze die Orientierung einer Person über kurzfristige Zeiträume ziemlich genau und präzise schätzen und mithilfe des hier vorgestellten Verfahrens (siehe Kapitel 6) langfristig kalibrieren und somit stabilisieren, ist die Schätzung der Geschwindigkeit immer noch schwierig, hauptsächlich aufgrund des Fehlers, der sich durch die Integration von verrauschten Beschleunigungs- und Winkelgeschwindigkeitssignalen über einen langen Zeitraum ansammelt. Darüber hinaus führt eine ungenaue Kalibrierung des relativen Koordinatensystems der IMU auf das ab-

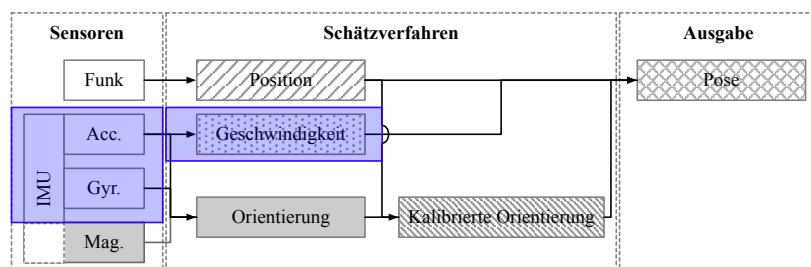


Abbildung 5.1: Einordnung der Geschwindigkeitsschätzung in die Verarbeitungskette (blaue Kästchen umschließen die beteiligten Komponenten).

5 Geschwindigkeitsschätzung

solute Koordinatensystem der Umgebung schnell zu explodierenden Schätzfehlern. Klassische Ansätze zur Schätzung der Geschwindigkeit optimieren diese für bestimmte Anwendungen, bspw. laufende Menschen. Die Platzierung der Sensoren und Bewegungsarten müssen ebenfalls räumlich kalibriert werden oder erfordern eine umfassende Parametereinstellung. Daher sind diese Ansätze nicht für unterschiedliche Bewegungsarten, bspw. laufen, rennen und springen geeignet. Darüber hinaus verursachen unterschiedliche Dynamiken, wie sie in Sportanwendungen zu finden sind, abrupte und unvorhersehbare Fehler bei der Kalibrierung zwischen Sensor und Umgebung. Die Schrittzahl und die Schrittlänge führen dann zu falschen Geschwindigkeitsschätzungen.

In diesem Kapitel wird daher eine neuartige Hybridfilterarchitektur vorgestellt, die sich nahtlos in die atomare Verarbeitungskette der Dissertation eingliedert und eine der wesentlichen Komponenten dieser darstellt, siehe Abbildung 5.1. Die Architektur kombiniert tiefes Lernen (DL) mit einem sequentiell nachgelagerten Bayesschen Filter zur Schätzung der Geschwindigkeit einer Person in rotationsinvarianten Signalströmen. Eine Kombination aus einem CNN und einem bidirektionalen LSTM (BLSTM) extrahiert räumliche Merkmale aus den Sensorsignalen und verfolgt deren zeitliche Beziehungen. Ein LKF wird dann verwendet, um die Varianz der Geschwindigkeitsschätzungen zu glätten.

Die Evaluation der Geschwindigkeitsschätzung (siehe Abschnitt 6.8) zeigt, dass der hier vorgeschlagene Ansatz auch in hochdynamischen Situationen, bspw. Bewegungen im Sport, robust gegenüber unterschiedlichen Bewegungszuständen, z. B. stehen, laufen, joggen und rennen, und Orientierungsänderungen, z. B. Kopf- oder Körpertrehung, ist. Bei Daten von einer einzelnen nicht kalibrierten IMU übertrifft diese Architektur den Stand der Technik in Bezug auf Geschwindigkeit ($\leq 0.16 \text{ m/s}$) und zurückgelegte Strecke ($\leq 3 \text{ m/km}$). Der Ansatz lässt sich auch gut auf unterschiedliche und variierende Bewegungsgeschwindigkeiten übertragen und liefert genaue und präzise Geschwindigkeitsschätzungen. Die neue Architektur wird mit konventionellen Methoden und mit Verfahren des maschinellen und tiefen Lernens aus dem Stand der Technik mit zeitlich unabhängigen Schnappschüssen der Daten verglichen. Die Geschwindigkeiten wer-

den zur Posenschätzung in Kapitel 7 verwendet.

Das Kapitel ist folgendermaßen strukturiert: Abschnitt 5.1 führt die Geschwindigkeitsschätzung ein. Abschnitt 5.2 beschreibt die Herausforderungen der Geschwindigkeitsschätzung formal. Abschnitt 5.3 diskutiert Arbeiten aus dem Stand der Technik. Aus der Diskussion leitet Abschnitt 5.4 die Beiträge der hier erforschten Geschwindigkeitsschätzung ab. Abschnitt 5.5 beschreibt die Verarbeitungskette einer ressourcenschonenden und einer hochgenauen Variante der Methode und deren Architektur. Abschnitt 5.6 beschreibt das Experiment, das zur Auswertung der Methoden verwendet wird. Abschnitt 5.7 präsentiert die Ergebnisse der Auswertung. Abschnitt 5.8 fasst die wesentlichen Inhalte des Kapitels zusammen und gibt einen Ausblick.

5.1 Einführung

Moderne PDR-Methoden können Geschwindigkeiten und Orientierungen kontinuierlich schätzen, um eine stabile Position abzuleiten. Dies funktioniert jedoch nur gut, wenn Beschleunigungsmesser und Gyroskope an den Füßen [105, 448] oder an der Wirbelsäule [286, 52, 124] montiert sind. Solche Verfahren erfordern dann keine zusätzlichen externen Sensoren oder Maschinen, die über einen längeren Zeitraum genaue Positionen liefern, um die modernen PDR-Methoden zu stabilisieren [464, 204]. Stattdessen ist eine hochpräzise Erstkalibrierung der Ausrichtung der Sensoren zur Umgebung ausreichend. Das Anbringen der Sensoren an den Füßen oder der Wirbelsäule ist jedoch (1) für den täglichen Gebrauch ungeeignet und (2) in vielen realen Anwendungsfällen, bspw. im Sport, unmöglich. Bei schnellen und abrupten Bewegungsänderungen kann die Geschwindigkeit und Ausrichtung der Sensoren nicht präzise und zuverlässig erfasst werden, da die erforderliche statische Sensorsausrichtung, bspw. bei Sportanwendungen, nicht garantiert ist. Eine andere Platzierung der Sensoren, z. B. in der Tasche eines Benutzers ist ebenfalls unmöglich, da die erforderlichen ZUPT- und ZARU-Mechanismen nur am Fuß funktionieren.

Naive Ansätze integrieren die Beschleunigung über die Zeit, um die entsprechende Geschwindigkeit zu schätzen. Das permanente Schätzen der Orientierung des Objekts und das Subtrahieren der Gravi-

5 Geschwindigkeitsschätzung

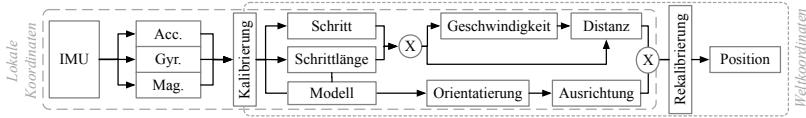


Abbildung 5.2: Klassische PDR-Verarbeitungspipeline.

tationskomponente vom Beschleunigungssignal ergibt die lineare Beschleunigung des Objekts in seinem Navigationsrahmen und die Integration über die Zeit ergibt seine Geschwindigkeit. Aufgrund von Sensorrauschen und anderen Artefakten, z. B. Sensordrift aufgrund von Temperaturänderungen und der falschen Schätzung der Gravitationskomponente ist dies jedoch nur für kurze Zeitspannen stabil. Daher sind solche Ansätze praktisch nur Erweiterungen zu einem Multisensor-Fusionssystem [147]. Fortgeschrittenere Ansätze zum Ableiten einer Position erkennen Schritte und schätzen ihre Länge anhand ihrer zeitlichen Beziehung. Sie kombinieren die Schritterkennung direkt mit einer Kursschätzung (engl. heading estimate) [124] und liefern so genauere Positionsschätzungen [106, 472]. Da sie jedoch eine (manuelle) Parametereinstellung erfordern, hängen sie stark von der Ausrichtung und Platzierung des Sensors auf dem zu lokalisierenden Objekt, bspw. Menschen, ab [464]. Dies beschränkt sie auf (Geh-)Bewegungen mit geringer Geschwindigkeitsschwankung und schließt Sport- und VR-Anwendungen aus, da sie nicht auf Laufen, Joggen und Rennen oder den Übergang zwischen ihnen angewendet werden können [472]. Abbildung 5.2 zeigt die typische Pipeline von klassischen PDR-Ansätzen. Abschnitt 1.5 beschreibt die Pipeline im Detail.

Neueste Methoden [286, 464, 204, 102, 236, 490, 128, 68, 489] verwenden maschinelles Lernen (ML) und tiefes Lernen (DL), um die Geschwindigkeit zu schätzen. Allerdings hängt hier die Genauigkeit der Geschwindigkeitsschätzung von einer genauen Orientierungsschätzung ab, um den Einfluss der Gravitationskomponente auf die Beschleunigung respektive deren Linearanteil genau zu bestimmen. Oder eine anfängliche Sensorkalibrierung und eine Schätzung der systematischen und zufälligen Verzerrung (engl. systematic bias) sind erforderlich. Da diese Methoden auch implizit die Abhängigkeit zwischen Geschwindigkeitsänderungen und Orientierungsänderungen lernen, beeinträchtigt eine leicht falsche Orientierungsschätzung die Geschwindigkeitsschätzung stark. Daher sind diese Methoden bisher

nur für homogene und quasi statische Bewegungen wie Gehen ohne abrupte oder schnelle Orientierungsänderungen genau.

Dieses Kapitel beschreibt die IMU-basierte Geschwindigkeitsschätzung als Regressionsproblem. Dazu wird eine neuartige Hybridfilterarchitektur vorgestellt, die DL zur Geschwindigkeitsschätzung nacheinander mit einem Bayesschen Filter zur Optimierung der Schätzung kombiniert. Ein CNN extrahiert räumliche Merkmale aus dem Signalstrom. Ein RNN verwendet die zeitlichen Beziehungen der Merkmale, um robuste, rotationsinvariante und hochgenaue Geschwindigkeiten zu schätzen. Der Eingabesignalstrom setzt sich dabei aus dem Betrag der schwerkraftfreien Beschleunigung und Winkelgeschwindigkeitssignalen einer einzelnen, nicht kalibrierten IMU zusammen.

In Abschnitt 5.7 werden die Ergebnisse der neuartige Hybridarchitektur mit den neuesten PDR-, ML- und DL-Ansätzen aus dem Stand der Technik verglichen. Das Verfahren lernt, den Betrag des Rohsignals auf eine entsprechende Geschwindigkeit abzubilden, arbeitet auch bei unterschiedlichen und dynamisch variierenden Bewegungsgeschwindigkeiten und -typen (Gehen, Joggen, Laufen und einer zufälligen natürlichen Kombination davon) gut und arbeitet auch unter dynamisch variierenden Ausrichtungen des Sensors robust, da das Verfahren nur den Signalgrößenvektor (SMV) verwendet. Die Methode wird anhand zeitlich unabhängiger Schnappschüsse der Daten evaluiert, um den Zusammenhang zwischen räumlichen und zeitlichen Abhängigkeiten der Informationen in den Daten hervorzuheben.

Die Ergebnisse der Experimente zeigen, dass klassische Ansätze nicht auf unterschiedliche und variierende Bewegungstypen verallgemeinert werden können, die neuartige Hybridarchitektur jedoch genaue und präzise Geschwindigkeitsschätzungen liefert. Sowohl die Erweiterung der Modelle um einen LKF als auch die Berücksichtigung der zeitlichen Abhängigkeiten in den Trainingsdaten verbessern die Gesamtergebnisse erheblich. Die Evaluation zeigt auch, dass die rotationsinvariante Architektur Geschwindigkeiten direkt aus dem Betrag der Rohbeschleunigungen und Winkelraten abschätzt und den Stand der Technik hinsichtlich der Genauigkeit der momentanen Geschwindigkeit ($\leq 0.10 \text{ m/s}^2$) und der zurückgelegten Strecke übertrifft ($\leq 29 \text{ m/km}$) [394, 147], auch in hochdynamischen Szenarien.

5 Geschwindigkeitsschätzung

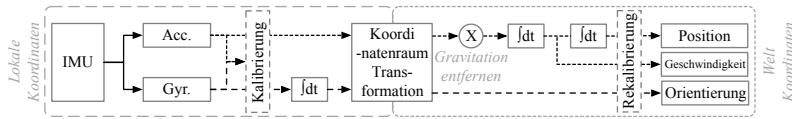


Abbildung 5.3: Naive SINS-Verarbeitungspipeline, die die Ausrichtung und Position einer einzelnen IMU ableitet.

5.2 Problembeschreibung

In diesem Abschnitt wird zunächst die Geschwindigkeitsschätzung für Trägheitsdaten unter perfekten Bedingungen vorgestellt. Dann werden Notationen eingeführt, die in dieser Arbeit verwendet werden, und die Hauptherausforderungen der IMU-basierten Geschwindigkeitsschätzung, werden diskutiert (Abschnitt 5.2.1). Schließlich wird ein Lösungsansatz vorgeschlagen (Abschnitt 5.2.2).

5.2.1 Herausforderungen

Die Trägheitsnavigation schätzt die Pose eines Objekts über die Zeit, d. h. seine Position und Ausrichtung aus einer anfänglichen Pose und mit einem dreiachsigem Beschleunigungsmesser (*acc*) und einem dreiachsigem Gyroskop (*gyr*). Konzeptionell wird die Orientierungsschätzung auf die Trägheitsmessungen angewendet, um sie in den globalen Navigationsrahmen umzuwandeln, in der Literatur oftmals auch als *Weltkoordinaten* bezeichnet. Von dort aus werden Geschwindigkeiten, Orientierungen und Positionen geschätzt. Abbildung 5.3 zeigt das Grundkonzept von SINS-Trägheitsnavigationsalgorithmen.

Eine typische SINS-Pipeline besteht aus mehreren Schritten. Zunächst werden Winkelraten und die Richtung der Gravitationskraft, in den Beschleunigungsmessungen enthalten, verwendet, um die Körperrahmenorientierung b mit Bezug zu dem globalen Navigationsrahmen n zu bestimmen. Das Einheitsquaternion $q_b^n(t)$, das eine Rotationsmatrix $R(q_b^n(t))$ impliziert, definiert die Orientierung. Die Abhängigkeit der Beschleunigungsmessungen im Zeitschritt t vom physischen Zustand des zu verfolgenden Objekts wird wie folgt dargestellt:

$$y_a(t) = R(q_b^n(t))(a^n(t) - g^n) + b_{acc}(t) + e_{acc}(t), \quad (5.1)$$

mit dem Gravitationskraftvektor $g^n = [0, 0, g]^T$ des Navigationsrahmens, dem Fehlerterm $e_{acc}(t)$ und der Fehler des Beschleunigungsmessers $b_{acc}(t)$. Zusätzliche Modelle wie Fehlausrichtung und Drift wurden aus Gründen der Übersichtlichkeit weggelassen. Die (deterministischen) Fehler werden durch eine initiale Kalibrierung beseitigt, führen jedoch im Zeitverlauf immer noch zu erheblichen Fehlern, insbesondere bei kostengünstigen Sensoren, wenn keine Rekalibrierung zur Laufzeit stattfindet. Daher wird das Orientierungsquaternion ohne systematischen Fehler bestimmt, wenn die Beschleunigung des Objekts $a^n(t)$ ausreichend klein ist und gilt, dass:

$$y_a(t) \approx -R(q_b^n(t))g^n + e_{acc}(t). \quad (5.2)$$

Dennoch die Orientierungsinformationen, die von nur einem Beschleunigungsmesser erhalten werden, der an einem dynamischen Objekt montiert ist, werden aus zwei Gründen nicht als sehr zuverlässig angesehen: Erstens kann das Objekt beschleunigt und verlangsamt werden, sodass $a^n(t) \approx 0$ nicht gilt. Zweitens kann der Gierwinkel (engl. yaw), d. h. der Euler-Winkel, der der Drehung mit Bezug zu der Körperachse entspricht, nicht bestimmt werden, da die Ausrichtung des Gravitationsvektors auf der Körperachse selbst liegt. Daher verwenden SINS normalerweise Gyroskope, um die Orientierung zu schätzen. Eine Gyroskopmessung ergibt sich aus der Beobachtung der Winkelraten $\omega(t)$ des Objekts, die durch das Sensorrauschen $e_{gyr}(t)$ und die Gyroskopfehler gestört werden $b_{gyr}(t)$:

$$y_g = \omega(t) + b_{gyr}(t) + e_{gyr}(t). \quad (5.3)$$

Die Winkelgeschwindigkeitsmessungen werden nicht durch Translationsbewegungen beeinflusst und beobachten unterschiedliche Rotationen für alle Richtungen. Allerdings repräsentieren diese Messungen nur die relative Rotation, da das Gyroskop keinen Bezug zur Ausrichtung des Weltkoordinatensystems hat. Da die Ableitung $\omega(t)$ der räumlichen Rotation beobachtet wird, führt dies zu einer Akkumulation des Fehlers $e_{gyr}(t)$ über die Zeit.

Um die absolute Ausrichtung zu schätzen, werden daher Beschleunigungsmesser, Gyroskope und andere Sensoren wie Magnetometer kombiniert [377, 287, 239]. Letztere ermöglichen die Messungen aus dem lokalen Koordinatensystem der Sensoren in das absolute Weltkoordinaten-

5 Geschwindigkeitsschätzung

tensystem zu transformieren. Die Orientierung wird dann verwendet, um Gleichung 5.1 für $a^n(t)$ zu lösen:

$$a^n(t) = R(q_b^n)^{-1}(y_a(t) - e_{acc}(t)) + g_n. \quad (5.4)$$

Somit ergibt die einfache Integration von $a^n(t)$ eine Schätzung der aktuellen Geschwindigkeit:

$$v(t) = v(t-1) + \int_{t-1}^t a^n(\tau) d\tau, \quad (5.5)$$

und die einfache Integration von $v^n(t)$ ergibt eine Schätzung der aktuellen Position:

$$p(t) = p(t-1) + \int_{t-1}^t v(\tau) d\tau. \quad (5.6)$$

Abbildung 5.3 zeigt die gesamte SINS-Verarbeitungskette: Zunächst werden die deterministischen Sensorfehlerkomponenten der Beschleunigungsmesser- und Gyroskopdaten während der Kalibrierung reduziert bzw. entfernt. Dann wird unter Verwendung der kalibrierten Sensordaten die Drehung von b mit Bezug zu n für den aktuellen Zeitschritt berechnet und das Beschleunigungsmessersignal entsprechend transformiert. Schließlich wird die Schwerkraft entfernt und die einfache (zweifache) Integration der transformierten Beschleunigungsmessersignale ergibt eine Geschwindigkeits- (Positions-)schätzung.

Während dieses Prozesses wird das Beschleunigungsmesserauschen $e_{acc}(t)$ einmal über die Zeit integriert, um die Geschwindigkeit zu schätzen, und zweimal für die Position, was zu einer signifikanten Fehlerakkumulation führt. Darüber hinaus werden im Laufe der Zeit auch falsche Fehler-Schätzungen des Beschleunigungsmessers $b_{acc}(t)$ und des Gyroskops $b_{gyr}(t)$ sowie andere deterministische Fehlerquellen integriert. Die Schwere dieser Fehler hängt stark von der Sensorqualität ab. Daher liefern Sensoren mit geringer Qualität, wie sie typischerweise für Endverbraucher bspw. in Mobiltelefonen verwendet werden, ungenaue Geschwindigkeitsschätzungen. Die Genauigkeit der Geschwindigkeitschätzung hängt auch von der Platzierung des Sensors, seiner Ausrichtung mit dem Körperteil, an dem er befestigt ist, dem Bewegungszustand und dem Sensorrauschen ab. Die deterministischen Fehlerquellen, d. h. die Kalibrierungsversätze, können jedoch nur dann anfänglich zuverlässig bestimmt werden, wenn das Sensorrauschen gering ist. Und für kostengünstige IMUs für Endverbraucher wird sogar eine aufwendige Systeminitialisierung benötigt, um sie zu kalibrieren und ihre Ausrichtung zu initialisieren.

Im täglichen Gebrauch ist eine solche Initialisierung mit geringer Genauigkeit für kostengünstige und auch für taktische Sensoren nicht möglich. Darüber hinaus erhöhen dynamische Bewegungen oder eine nicht starre Montage des Sensors die Komplexität der Modellierung [84]. Daher können bei dynamischen Bewegungen, z. B. in Sportarten, bei denen die Orientierungsschätzung ungenau ist, die lineare Komponente $a^n(t)$ überhaupt nicht für die Geschwindigkeits schätzung verwendet werden [69].

Konzeptionell treten bei der IMU-basierten Positionierung zwei Arten von Fehlern auf [107]: (1) Rein stochastische Größen ohne zeitliche Korrelation, z. B. die thermomechanischen Rauschkomponenten c (e_{acc} , e_{gyr}). Diese Komponenten enthalten keine Informationen über den physischen Zustand des Körpers und werden deshalb statistisch modelliert. (2) Zeitkorrelierte Größen, z. B. die deterministischen Rauschkomponenten der Sensoren r ($b_{acc}(t)$ und $b_{gyr}(t)$), bei denen die zeitliche Korrelation durch Sensordrift oder Rotations- und Translationsbewegungen eingeführt wird und Modellverletzungen verursacht. Das zeitliche Verhalten dieser Komponenten ist statistisch schwer zu modellieren, deshalb lernt die vorgeschlagene Methode es direkt aus sorgfältig gewählten Sensordaten.

Insgesamt erzielen klassische SINS in Kombination mit IMUs für Endverbraucher in vielen Anwendungen keine genauen Geschwindigkeitsschätzungen, da sich die durch Modellverletzungen verursachten Fehler durch die Integration ausbreiten. Es gibt folglich keine einfache Lösung für dieses Problem. Im folgenden Abschnitt wird deshalb ein Lösungsansatz vorgeschlagen, der (1) aus Daten und Erfahrungen die Auswirkungen des kurz- und langfristigen Rauschens $c + r$ auf die Geschwindigkeitsschätzung erlernt und (2) den Einfluss von Bewegung und Orientierung auf die Projektion von a reduziert.

5.2.2 Lösungsansatz

Um das oben diskutierte Problem der Geschwindigkeitsschätzung zu lösen, wird nachfolgend eine Reihe von Gegenmaßnahmen vorgeschlagen. Um die Herausforderungen typischer SINS zu bewältigen, wird die IMU-basierte Geschwindigkeitsschätzung als überwachtes Regressionsproblem [45] beschrieben: Eine funktionale Beziehung zwischen der zu schätzenden numerischen Geschwindigkeit und den

5 Geschwindigkeitsschätzung

IMU-Eingangsdaten wird abgeleitet, d. h. zwischen dem räumlichen Muster (die Form des Signals) und der zeitlichen Beziehungen innerhalb solcher Muster (die langfristigen Variationen in verschiedenen Mustern des Signals, die durch Bewegung und deterministische Sensorfehlerkomponenten verursacht werden). Das ist möglich, da die Abbildungsfunktion, die die funktionale Beziehung beschreibt, diese an markierte Daten mit bekannten zusammenhängenden Werten anpasst. Es wird eine Regressionsanalyse verwendet, um eine Funktion abzuleiten, die diese Abbildung, Trägheitsdaten zur Referenzgeschwindigkeit, unter Verwendung eines Satzes trainierter Parameter approximiert. Daher wird angenommen, dass das Restnetz (engl. Residual Network, ResNet) [176] lernt, Rauschen zu entfernen (c entfernen) und räumliche Merkmale $f_s(x)$ aus einer Folge von Trägheitsdaten zu extrahieren. Weiter wird davon ausgegangen, dass das BLSTM die Entstehung und Entwicklung dieser Merkmale $f_t(x)$ sowohl aus der Vergangenheit als auch aus der Zukunft verfolgt. Zusätzlich verfolgt ein LKF den Zustandsraum von $f_s(x)$ und $f_t(x)$ und optimiert die Endgeschwindigkeit. Details zu Rest- und BLSTM-Netz werden in Abschnitt 5.5 beschrieben.

Es werden auch die Auswirkungen von hohem $c + r$ auf die fehleranfällige Projektion a gelöst. Anstelle von Methoden, die dreiachsige Messungen verarbeiten, bei denen jede Achse einzeln um $c+r$ fehlerbehaftet ist, wird die Methoden direkt auf einen einzelnen Kanal der Messung angewendet, d. h. auf den Signalbetragvektor (SMV) oder seine charakteristischen Merkmale. Diese Bestimmung der Norm des Mehrkanalsignals, führt möglicherweise dazu, dass Informationen über die Bewegung verloren gehen, jedoch ermöglicht SMV eine Rotationsinvarianz der dreiachsischen Messungen. Somit ist das Zerlegen von linearer Beschleunigung und Schwerkraft weniger fehleranfällig, da die fehleranfällige Projektion von a umgangen wird, selbst wenn $c + r$ hoch ist. Die Gravitationskonstante g (=9.81 m/s) wird vom SMV der Beschleunigung subtrahiert und so die Norm der linearen Komponenten von a ohne den Einfluss eines falsch projizierten g abgeleitet. Eine komplexe (Neu-)Kalibrierung und Fehler-Schätzung sind nicht mehr erforderlich.

Schwankungen in der Drehung (Pose) des Sensors haben daher keinen Einfluss auf die Genauigkeit der Hybridarchitektur. Stattdessen lernt die Methode implizit die SMV von b auf n über mehrere Koordinatensysteme, bspw. wenn der Sensor in der Hosentasche platziert wird: vom Bein über die Tasche und vom Nutzer zur Umgebung, zu

projizieren. Daher ermöglicht die Methode Anwendungen mit hoher Bewegungsdynamik, bei denen davon auszugehen ist, dass sich nur ein einziger frei beweglicher Sensor auf einem Mobiltelefon bspw. in der Beintasche oder am Kopf eines Benutzers befindet. In solchen Anwendungen schlagen Methoden nach dem Stand der Technik fehl, weil sie diese Orientierungsschwankungen nicht korrekt kalibrieren und projizieren können. Die Hybridarchitektur kann in der Praxis für viele Szenarien eingesetzt werden, da sie mit natürlichen Platzierungen der Sensorik umgehen kann, wenn andere Methoden versagen.

Darüber hinaus wird das Problem behoben, dass die klassische Geschwindigkeitsschätzung Fehler kontinuierlich über die Zeit integriert. Da dieser Zyklus unterbrochen wird, indem quasi voneinander unabhängige Fenster über das Signal [189, 70] geschoben werden, sodass das Modell keine Zusammenhänge zwischen Fenstern oder Datensätze auswendig lernen kann. Jedoch bilden die Fenstergröße und die Überlappung Schlüsselparameter zur Optimierung eines Modells in Bezug auf unterschiedliche Bewegungen.

Schließlich repräsentiert der in der Bewertung verwendete Datensatz (Abschnitt 5.7) reale Szenarien mit einem Sensor (einem Mobiltelefon) in einer natürlichen Position (der Beintasche). Dieser Datensatz ist der erste, der verschiedene Geschwindigkeiten dynamischer menschlicher Bewegungen abdeckt, wie sie in typischen Sport- und VR-Anwendungen vorkommen: vom langsamen Gehen bis zum Rennen mit hoher Geschwindigkeit, einschließlich schneller Orientierungsänderungen auf zufälligen Pfaden. Im Gegensatz zu anderen Untersuchungen [489, 490, 70] wird der Einfluss von Parametern auf die Robustheit und Genauigkeit von Geschwindigkeitsschätzungen untersucht.

5.3 Verwandte Arbeiten

Die verwandten Arbeiten werden in fahrzeugfeste (engl. strapdown) Trägheitsnavigationssysteme (Abschnitt 5.3.1), stufen- und modellbasierte Fußgänger-Koppelnavigation (Abschnitt 5.3.2) und datengetriebene Methoden (Abschnitt 5.3.3) gruppiert.

5.3.1 Fahrzeugfeste Trägheitsnavigationssysteme

Fahrzeugfeste Trägheitsnavigationssysteme SINS sind die ältesten Methoden zur Schätzung der Geschwindigkeit und Orientierung einer

5 Geschwindigkeitsschätzung

Person, um eine Position dieser [435, 386] ausschließlich anhand von Trägheitssensoren bereitzustellen. Anfänglich verwendeten militärische Systeme teure und schwere Trägheitsmesseinheiten (IMUs) mit hohem Energieverbrauch [401]. Während solche militärischen Sensoren aufgrund ihrer großen Bauweise ausschließlich in Autos, Schiffen, Flugzeugen, Raketen usw. verwendet werden, sind sie für Endbenutzergeräte nicht geeignet [170]. Die jüngsten Fortschritte in der Mikroelektromechanischen Systemtechnologie (MEMS) ermöglichen kostengünstige, kleine und leichte IMUs für Endverbraucher mit geringerem Energieverbrauch, die in der Robotik und in mobilen Geräten wie Mobiltelefonen verwendet werden können und die auch für die Experimente dieser Dissertation verwendet werden können [105, 286, 52, 124, 46, 286]. SINS werden in der Literatur in zwei unterschiedliche Gruppen unterteilt: naive und optimierte SINS.

Naive SINS

Die naiven SINS integrieren IMU-Signale wie Winkelgeschwindigkeit und Beschleunigung direkt in Orientierung und Geschwindigkeit, um eine Position zu schätzen. Aufgrund der begrenzten Größe und der geringen Kosten von Endverbraucher-IMUs ist jedoch auch deren Genauigkeit begrenzt. Naive SINS können bei Verwendung mit IMUs für Endverbraucher keine Position schätzen, da signifikante Fehler und das Rauschen ihrer Beschleunigungsmesser und Gyroskope bereits zu fehlerhaften Distanzeinschätzungen führen, die sowohl unbegrenzt als auch kubisch zunehmen [46]. So führen das hohe Sensorrauschen, die Fehler und das Fehlen einer Starrkörperbefestigung bereits nach sehr kurzen Zeitspannen (≤ 1 s) zu explodierenden Schätzfehlern. In den Experimenten in Kapiteln 5, 6 und 7 werden solche fehleranfälligen Sensoren für Endverbraucher verwendet. Im Gegensatz zu einer naiven Integration der Signale entrauschen (engl. denoise) die Methode dieser Arbeit das Eingangssignal implizit, indem sie (Ende-zu-Ende) Rausch-, Fehler- und Kalibrierwerte lernen.

Optimierte SINS

Um die Schwächen naiver SINS zu beseitigen, fusionieren die meisten Methoden IMUs mit anderen Sensoren, verwenden domänenspezifisches Wissen oder begrenzen akkumulierende Integrationsfehler. Visuelle trägheitsbasierte Odometrie, VIO) [353] führt zu höherer Robust-

heit und Genauigkeit [46, 259], aber VIO erfordert nicht nur eine sorgfältige Parametrisierung, Kalibrierung und Synchronisation der Sensoren, sondern die Bildsensorik leidet unter typischen Kameraproblemen wie Verdeckung und schlechter Beleuchtung. Um diese Probleme zu bewältigen, werden die Sensoren mit Bayesschen Filtern [170] fusioniert und mit Graphoptimierungen [46, 286, 406] oder mit visueller Trägheitsbasierte simultaner Lokalisierung und Kartografie (engl. Visual-Inertial Simultaneous Localization and Mapping, VI-SLAM) behandelt [259]. Andere optimierte SINS, die keine anderen Sensoren verwenden, nutzen das Domänenwissen. Z. B. befestigen sie eine IMU am Fuß eines Benutzers, um Nullgeschwindigkeitsaktualisierungen zu erkennen, um Fehler in Trägheitssystemen zu kompensieren [259, 406]. ZUPT nutzt die Tatsache, dass ein am Fuß platzieter Sensor während eines Spaziergangs in regelmäßigen Abständen stationär ist, und nimmt diese Momente als Referenzmomente (engl. Ground Truth Moments, GTM), um akkumulierte Fehler zu korrigieren. Bei ZUPT muss der Sensor jedoch entweder am Fuß [259, 406] oder am Kopf [124] montiert werden, um genaue GTMs bereitzustellen. Im Gegensatz dazu lernt die Geschwindigkeitsschätzung, die in diesem Kapitel vorgeschlagen wird, diese GTMs direkt und implizit von einem einzelnen Sensor und erfordert daher keine komplexe Parametrisierung mehrerer Sensoren oder ZUPT für einen bestimmten Benutzer oder eine bestimmte Aktivität. Die Methode entfernt auch die Rotationsvarianz der Trägheitsdaten und ist daher nicht auf einen bestimmten Montagepunkt am zu lokalisierenden Objekt, bspw. den Benutzer, beschränkt. Ferner leidet die Methode nicht unter typischen Kameraproblemen wie VIO, da sie nur Trägheitssensoren verwendet.

5.3.2 Fußgänger Koppelnavigation

Im Gegensatz zur offenen Schleifen Integration (engl. open-loop-integration) von Trägheitssensoren in SINS verwendet die stufenbasierte Fußgänger-Koppelnavigation Trägheitsmessungen, um Schritte zu erfassen und die Schrittänge und -richtung mithilfe empirischer Formeln [403, 377] zu schätzen. Die genauesten PDR-Methoden des Stands der Technik, die zu Fehlern <0.4 m/s führen, erfordern eine starre Geräteorientierung, m.a.W. eine feste Verbindung mit dem zu lokalisierenden Objekt, die an der Bewegungs- bzw. Fahrtrichtung ausgerichtet ist [84, 472]. PDR-Systeme gehen häufig von einem periodischen Bewegungsmuster aus, bei dem Schritte

5 Geschwindigkeitsschätzung

präzise erfasst werden können [105, 454]. Typischerweise verwendet die Schritterkennung oder -segmentierung Schwellenwerte [376], erkennt Signalspitzen (engl. peak) [496], korreliert bekannte und unbekannte Schrittmuster [246, 348] oder analysiert das Signalspektrum, um Schritte aus Beschleunigungsmessungen zu erfassen [48]. Fehler häufen sich jedoch schnell mit falsch erkannten Schritten oder ungenau geschätzten Schrittängen. Solche PDRs leiden auch unter komplexeren und variierenden Bewegungsmustern, wie abrupten Bewegungsänderungen oder variierenden Geschwindigkeiten, weil die Orientierung schwankt. Eine fehlerhafte Erkennung aufeinanderfolgender Schritt führt leicht zu einer ungenauen Schrittängenschätzung. Um dies auszugleichen, müssen eine Reihe von Parametern an die Gehgewohnheiten des Benutzers und an den spezifischen Anwendungsfall angepasst werden [210].

Die aktuelle Forschung konzentriert sich auf die Fusion von PDR Verfahren mit externen Sensordaten von bspw. funkbasierten Sensoren, wie z. B. Wi-Fi [394, 188], GPS [223] oder Ultra-Breitband (UWB) [147], oder Magnetometer [178, 460] oder mit Informationen über die Umgebung [61, 232, 479], da die technischen Herausforderungen wie der Sensordrift und die Zerlegung der Beschleunigung immer noch ungelöst sind. Auch rotationsinvariante Methoden, die frequenzbasierte Parameter verwenden, leiden dennoch noch unter einer schlechten Genauigkeit [105, 52]. Im Gegensatz dazu arbeitet die vorgeschlagene Methode ohne zusätzliche Sensoren und schätzt die Geschwindigkeiten eines einzelnen Trägheitssensors, der bspw. in einer Beintasche oder am Kopf eines Menschen frei beweglich platziert wird. Da die Methode eine Technik implementiert, die rotationsinvariant ist, liefert sie genaue Geschwindigkeiten sogar für reine, nicht kalibrierte Trägheitsdaten.

Oft werden sowohl SINS- als auch PDR-Methoden mit biomechanischen Modellen erweitert, um die Schritterkennung und Längenschätzung optimieren. Um die Genauigkeit der Schrittängenschätzung zu verbessern, wurden zahlreiche Modelle untersucht: Modelle empirischer Beziehungen [206, 190], biomechanische Modelle [206, 336, 213], lineare Modelle [251], nichtlineare Modelle [467, 234, 13] und Modelle, die auf Regression basieren [251, 400, 168, 480]. Die besten Ergebnisse werden mit Modellen erzielt, die das Wissen über die Mechanik des menschlichen Körpers in der Schrittphase ausnutzen und kluge An-

nahmen treffen, um die Schrittänge besser zu schätzen. Da vor allem menschliche Bewegungen normalerweise sehr komplex sind, können Modelle solche Mechaniken nur annähern [105, 299, 233, 418]. Typischerweise werden unterschiedliche Modelle für unterschiedliche IMU-Platzierungen verwendet, z. B. Wirbelsäule, Hand, Tasche [105, 160], Fuß [105, 233] und Kopf [124]. Da sich die Mechanik auch zwischen Bewegungsarten unterscheidet, z. B. Gehen und Rennen, erfordern die Modelle spezielle Schrittlängenschätzer [233, 52, 160]. Weinberg [467] präsentiert das bekannteste biomechanische Modell, das in der Evaluation in Abschnitt 5.6 verwendet wird. Diese klugen Kombinationen erfordern jedoch eine aufwendige Kalibrierung und Parametrisierung sowohl des PDR als auch des biomechanischen Modells pro Benutzer. Sie leiden auch unter Orientierungsvarianz und bieten eine geringere Genauigkeit als datengetriebene Verfahren aus dem Stand der Technik [464, 458, 204]. Im Gegensatz dazu benötigt die hier vorgeschlagene Methode weder eine detaillierte Beschreibung des menschlichen Gangs noch eine genaue Sensorskalibrierung, da die Methode rotationsinvariant ist und implizit präzise, aber allgemeine, für alle Probanden des Experiments gültige, biomechanische Modelle direkt aus den Trägheitsrohdaten lernt. Um dies zu erreichen, wird das Modell der Methode mit Datensätzen trainiert, die eine Vielzahl von Bewegungen abdecken, einschließlich abrupter Kurven, sich schnell ändernder Beschleunigungen und sehr langsamer bis sehr schneller menschlicher Bewegungen.

5.3.3 Datengetriebene Methoden

Die meisten datengetriebenen Methoden übertreffen herkömmliche Methoden. Während ML-basierte Methoden häufig handgefertigte Merkmalsextraktion (engl. feature engineering) erfordern, extrahieren DL-basierte Methoden automatisch die Repräsentation von optimalen Merkmalen aus Daten und lernen ein allgemeineres Bewegungsmodell direkt aus Trägheitsdaten. Datengetriebene Methoden zur Verarbeitung von menschlicher Bewegung werden seit Jahrzehnten untersucht, wobei der Schwerpunkt auf der Analyse oder Rekonstruktion des Gangs und der Haltung von gehenden Menschen mithilfe von Trägheitsmessungen liegt [454, 328, 291, 290, 440, 502, 478]. Chang et al. [63] lernen intuitive Physik [124], Karl et al. [227] erstellen Zustandsraumdarstellungen (engl. state-space representations) [299] und Stewart et al. [418] überwachen KNN mit Hilfe von physikalischem Wissen [418, 377]. Andere [448, 315, 314] lernen, eine Systemdrift zu begrenzen oder eine ge-

5 Geschwindigkeitsschätzung

nauere ZUPT-Phase zu erhalten, um die Fehler von Trägheitssystemen zu kompensieren [142, 204, 464]. Die meisten dieser Ansätze konzentrieren sich jedoch auf die Analyse der menschlichen Bewegung und der Klassifikation von Bewegungsmustern. Während die in dieser Arbeit vorgestellte Methode von bekannten datengetriebenen Methoden zur Analyse menschlicher Bewegungen inspiriert ist, konzentriert sich die Methode auf die Lokalisierung basierend auf der Geschwindigkeit, d. h. sie *extrahiert* zuerst implizit Merkmale menschlicher Bewegung mit einem CNN und *verfolgt* diese dann im Zeitverlauf mit einem RNN. Im Bearbeitungszeitraum dieser Dissertation ist keine verwandte Arbeit entstanden, die Geschwindigkeit, Orientierung oder Position in einer vergleichbaren Weise schätzt. Die datengetriebenen Methoden werden in drei Gruppen untergliedert: Methoden, die mehrere Sensoren verwenden, hybride Methoden, die Modell- und datengetriebene Verfahren kombinieren und rein datengetriebene Methoden.

Mehrere Sensoren

Im Bereich der Lokalisierung von Menschen basieren die meisten DL-Methoden auf VIO und kombinieren verschiedene Sensoren [353]. Konda et al. [244] schlagen ein CNN vor und verwenden Daten von zwei synchronisierten Sensoren, um die Richtungs- und Geschwindigkeitsänderungen vorherzusagen. Carrera et al. [61] verwenden zwei oder mehr IMUs oder eine IMU in Militärqualität [204], um akkumulierende Fehler zu reduzieren und das Zuverlässigkeit (engl. confidence) ihrer Algorithmen zu erhöhen. Da sie entweder auf einer eng gekoppelten Sensorfusion [359] oder auf der Verfügbarkeit zuverlässiger Kontextinformationen wie externer Sensoren [105, 147, 178, 359] oder Karten [61, 394, 232] beruhen, nimmt ihre Genauigkeit ab, je größer der Fehler einer externen Informationsquelle wird. Stattdessen arbeitet die vorgeschlagene Methode mit einem einzelnen Trägheitssensor und leidet daher nicht unter komplexen Kopplungen von Sensoren mit möglicherweise falschen externen Informationen.

Hybridmethoden

Hybridmethoden [106, 160, 69, 68, 489, 448] kombinieren ML oder DL mit bestimmten Modellen wie ZUPT oder kombinieren Klassifizierungs- und Regressionsmodelle, um sowohl die Schrittänge als auch die Orientierung zu schätzen. Eine Kombinati-

on mit ZUPT [448, 449, 464] reduziert den Sensordrift (Fehler <0.25 m/s) [448, 232] und liefert daher eine höhere Genauigkeit. Bei höheren Bewegungsgeschwindigkeiten versagt ZUPT jedoch aufgrund von Sensorrauschen und Bewegungsartefakten [448, 160, 464, 204].

Andere Methoden klassifizieren zuerst einen Bewegungstyp mit ML- oder DL-Methoden und wählen dann einen geeigneten Schritt-längenschätzer aus. Bspw. teilt RIDI [489] die Geschwindigkeits- und Orientierungsschätzung in drei Teile auf: Zuerst klassifiziert eine Stützvektormaschine die Platzierung des Sensors, dann verwendet RIDI die Klassifizierung, um einen optimalen Stützvektor Regressor (engl. Support Vector Regression, SVR) zum Schätzen von Geschwindigkeiten auszuwählen und schließlich stützt sich RIDI auf traditionelle Sensorfusionsmethoden, um Orientierungen zu schätzen. RIDI zeigt vielversprechende Ergebnisse (MAE=7.64 m gegenüber PDR: MAE=10.28 m) für die öffentlich verfügbaren Datensätze RIDI [489], OxIOD [69] und Robust Neural Inertial Navigation (RoNIN) [490]. Die Klassifizierung der Sensorplatzierung erfordert jedoch handgefertigte Merkmale und zeitaufwendige Trainingsschritte, nämlich die Klassifizierung der Sensorplatzierung, die Auswahl der optimalen Regressionsmethode und die Regression von Geschwindigkeit und Orientierung, und fordert auch einen hohen Rechenaufwand. Selbst kleine Anomalien, bspw. unbekannte Bewegungsmuster, in den Signalen führen solche Hybridmethoden in die Irre [106, 204], das zu einer geringen Genauigkeit [489] führt. Deshalb sind solche Methoden im Allgemeinen keine Option, wenn die Benutzerfreundlichkeit für eine echte Anwendung wichtig ist.

Im Gegensatz zu diesen Hybridmethoden wird die hier vorgeschlagene DL-basierte Methode in einer überwachten Ende-zu-Ende-Weise auf einer Vielzahl menschlicher Bewegungen mit einem großen Geschwindigkeitsbereich trainiert. Daher erkennt und klassifiziert das Modell der Methode implizit die menschliche Bewegung in einem Signal, um dann die richtige Kombination von Neuronen auszuwählen und zu aktivieren, die eine entsprechende Geschwindigkeit genau vorhersagen. Aufgrund des Ende-zu-Ende-Konzeptes leidet die Methode weder unter irreführenden Klassifizierungen noch unter unterschiedlichen und stark variierenden Bewegungen.

Rein datengetriebene DL-basierte Methoden

Rein datengetriebene DL-basierte Methoden lernen Ende-zu-Ende-Bewegungsrepräsentationen direkt aus Rohdaten. Xing et al. [480], Cho et al. [77] und Martinelli et al. [294] verwenden ein einfaches NN, um die Schrittänge aus Trägheitsdaten von Endverbraucher-IMUs zu schätzen. Ihre Leistung erreicht die von naiven SINS. Um dieses zu verbessern, verwenden neuere Studien RNNs, um ein Signal zu entrauschen [210, 68, 127], um Schritte zu erkennen [160, 448, 106] oder um ihre Länge zu schätzen [448, 106, 69]. Durch die Aufteilung des Problems müssen sich die Netze jedoch mit voneinander abhängigen Fehlern befassen, deren Beziehungen sie nicht lernen können. Wang et al. [458] kombinieren daher zwei DL-Architekturen, um die Schrittänge aus verrauschten Trägheitssensordaten zu schätzen: ein Langzeit-Kurzzeitgedächtnis (LSTM) zum Extrahieren der Zeitabhängigkeiten und einen entrauschenden Auto-Encoder (engl. Denoising Autoencoder, DAE), um das Rauschen von wichtigen Merkmalen zu entfernen. Im Durchschnitt erreichen diese Verfahren einen Schrittängenfehler von 4.63% und einen Abstandsfehler von $14.3\text{cm}/m$, wenn ein Benutzer mit einer an der Wirbelsäule befestigten IMU langsam läuft. Die Kombination von LSTM und DAE erfordert nicht nur viel Rechenaufwand und ein vorgelagertes und mehrstufiges Training des DAE, der Hauptnachteil ist die Rotationsvarianz: Um genaue Positionen zu schätzen, muss der Benutzer den Sensor vor die Brust halten.

Im Gegensatz zu diesen rein DL-basierten Methoden erfordert die hier vorgeschlagene Methode weder eine bestimmte Platzierung noch eine starre Position des Sensors. Stattdessen entfernt die rotationsinvariante Methode implizit Rauschen aus einem Signal und lernt, die Geschwindigkeit aus einer Vielzahl von Bewegungen mit unterschiedlichen Geschwindigkeiten vorherzusagen. Anstelle eines LSTM verwendet die Methode in Kapitel 5.5 ein BLSTM-Netz, das lernt, Bewegungen sowohl vom vorherigen als auch vom nächsten Zeitschritt auf die Geschwindigkeit [106] abzubilden.

Um all diesen Herausforderungen zu begegnen, schlagen Chen et al. [68] und Yan et al. [490] zwei DL-Modelle (IONet und RoNIN) vor, die den neuesten Stand der Technik der Positionierungsleistung bieten. IONet rekonstruiert Trajektorien direkt aus 6-Achsen-Trägheitsdaten. Ein BLSTM-Netz regressiert sowohl Geschwindigkeit als auch Orientierung. Im Gegensatz dazu teilt RoNIN die Rekonstruktion einer

Trajektorie in zwei unabhängige Teile. Im ersten Teil schätzt ein LSTM-Netz die Ausrichtung und im zweiten Teil schätzt ein Restnetz (ResNet-18 [176]) die Entfernung.

Yan et al. [490] verglichen die Leistung von RoNIN, IONet und PDR mit öffentlichen Datensätzen [69, 489, 490]. Wenn Benutzer im Kreis oder auf quasi zufälligen Pfaden gehen, litt IONet unter höheren Positionsschätzungsfehlern (14.60 m) als eine klassische PDR-Methode (10.28 m), während RoNIN den Stand der Technik vorgibt (Fehler 6.71 m), aber auch die Genauigkeit der anderen Methoden übertrifft (3.04 m). IONet und RoNIN erbrachten bei den Datensätzen, für die sie explizit entwickelt wurden, die beste Leistung. Diese Tatsache weist daraufhin, dass beide Methoden möglicherweise zu stark an die Daten angepasst sind. Sowohl IONet als auch RoNIN erfordern einen anfänglichen (Neu)Kalibrierungsschritt, um die endgültigen Trajektorien zu schätzen. Da die Neukalibrierung am Ende einer Trajektorie stattfindet, können akkumulierte Fehler zu Trajektorien führen, die von ihren Referenzen abweichen. Der Einfluss des langfristigen Drifts in Geschwindigkeit und Orientierung auf die Schätzleistung der beiden Methoden bleibt unklar. Das Hauptproblem der RoNIN-Architektur besteht darin, dass sie ausschließlich auf einem CNN basiert und daher nur räumliche Merkmale berücksichtigen kann, zeitliche Abhängigkeiten jedoch nicht gut erfassen kann [490]. Dies beschränkt seine Fähigkeit, auf unbekannte Umgebungen zu verallgemeinern [68, 70].

5.4 Beiträge

Im Gegensatz zu IONet und RoNIN schlägt die im Rahmen der Dissertation entwickelte Methode [128] eine sequentielle Kombination von CNNs und BLSTMs vor. Diese Methode ist rotationsinvariant, da sie direkt auf den Betragsvektor des Signals von Trägheitsrohdaten wirkt, der die Rotationsinvarianz ermöglicht. Das CNN der vorgeschlagenen Methode extrahiert zuerst räumliche Merkmale, und die BLSTMs erfassen dann das zeitliche Auftreten dieser Merkmale, um die gesamte Bewegungsdynamik der Trägheitsmessungen einzubetten. So schätzt die Methode die Geschwindigkeit auch in unbekannten Szenarien genau, insbesondere in Bewegungsszenarien mit abrupten Änderungen in Bewegung und Orientierung und mit verschiedenen Bewegungsaktivitäten wie Stehen, Gehen, Joggen, Laufen und einer Kombination

5 Geschwindigkeitsschätzung

auf zufälligen Trajektorien, wie sie typischerweise in Sport- und VR-Anwendungen vorkommen. Im Rahmen der hier vorgeschlagenen Methode werden CNNs und RNNs verknüpft, um die Bewegungsdynamik und die zeitlichen Abhängigkeiten von Trägheitsdaten zu modellieren. Alle Fehler entlang der Prozesskette der Geschwindigkeitsschätzung werden somit von der Methode Ende-zu-Ende in einem schwarzen Kasten (engl. blackbox) abgebildet, d. h. das Modell lernt den Umgang aller auftretender Fehler von der Beschleunigung bis zur Referenzgeschwindigkeit. Die Methode stellt einen direkten Trägheits-Odometer, mit höherer Zuverlässigkeit und Genauigkeit, für einen einzelnen rotationsinvarianten und nicht kalibrierten Trägheitssensor in Situationen mit dynamischer Bewegung bereit. Die 1D-CNN-Architektur der früheren Arbeit [128], die im Rahmen der Dissertation entstanden ist, wird in Abschnitt 5.5 durch eine 1D-ResNet-18 [335]-Architektur, inspiriert von RoNIN [490], ersetzt und mit einer BLSTM, inspiriert von IONet [68], zusammengeführt und die Ausgabe mit einem LKF optimiert. Dabei stellen sich folgende Forschungsfragen, RQ:

RQ3: Können datengetriebene Verfahren die Geschwindigkeitsschätzung eines Fußgängers aus verrauschten MEMS-Trägheitsmessungen verbessern?

RQ3a: Kann die Gravitationskraft eines lose befestigten Beschleunigungsmessers so eliminiert werden, um in Sport- und VR-Anwendungen mit hochdynamischen Bewegungsänderungen die Geschwindigkeit genau zu bestimmen?

5.5 Methode

Dieser Abschnitt diskutiert die neuartige Verarbeitungskette im Detail. Es wird die Datenvorverarbeitung beschrieben (Abschnitt 5.5.1), die Hauptverarbeitung mit den innovativen DL-Architekturen (zwei Varianten: eine ressourcenschonende in Abschnitt 5.5.2 und eine hochgenaue in Abschnitt 5.5.3) präsentiert und der optimierende Nachverarbeitungsschritt mit einem LKF eingeführt (Abschnitt 5.5.4).

Abbildung 5.4 skizziert die drei Schritte der Verarbeitungskette. Die neue DL-basierte Architektur benötigt die ausgegrauten Teile nicht, die nur für den Vergleich der Methode mit Methoden des Stands der Technik verwendet werden, bzw. benötigte Bestandteile der Posenschätzung

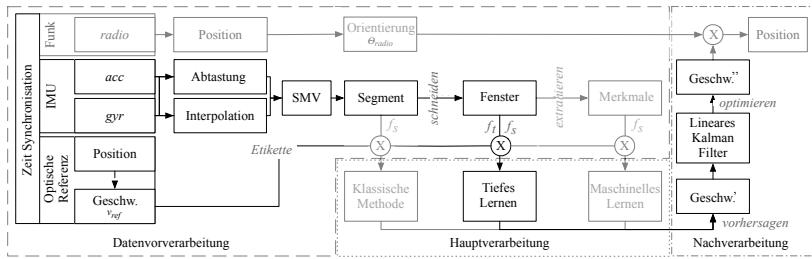


Abbildung 5.4: Verarbeitungskette der Hybridarchitektur zur Geschwindigkeitsschätzung. Die grauen Elemente visualisieren Schritte, die nur zur Bewertung der Methode verarbeitet werden. Die Daten fließen vom Eingang (links) zum Ausgang (rechts). v' und v'' repräsentieren Geschw.' und Geschw. im Text.

(Kapitel 7) sind, siehe Abschnitt 5.6. Die Vorverarbeitung synchronisiert die Eingabedatenströme (*radio*, *acc*, *gyr* und *ref*), bereinigt sie, bereitet sie vor durch Stichprobewiederholung (engl. resampling) und Interpolation und bündelt die Datenströme, damit später verschiedene Methoden des Stands der Technik (hier klassisch: ML und DL) sie verwenden können, um die Geschwindigkeiten v' zu schätzen. Bei der Nachbearbeitung gibt ein LKF Geschwindigkeiten v'' zurück, die noch weiter optimiert sind. Jeder dieser Schritte wird später im Detail erklärt.

5.5.1 Datenvorverarbeitung

Abbildung 5.5 skizziert die Datenvorverarbeitung im Detail. Die Daten fließen von den Eingangssensoren (links) zu den vorverarbeiteten Datenbündeln (Segmente zu Merkmalen, rechts). Die Vorverarbeitung erfolgt pro Bündel (engl. bundle), d. h. die Daten für eine einzelne Aktivität eines einzelnen Benutzers.

Interpolation und Stichprobewiederholung

Die Vorverarbeitung beginnt mit dem Bündel von Rohdaten vom Beschleunigungsmesser *acc*, Gyroskop *gyr*, optischem Referenzsystem *ref* und Hochfrequenzlokalisierungssystem *radio*. Die Positionen *ref* und *radio* werden auf 1000 Hz linear interpoliert. Dann werden alle Messungen von *acc* und *gyr*, mit einer konstanten Abtastrate von 100 Hz, mit beiden Referenzen synchronisiert. Um die synchronisierten Messungen zu bereinigen, werden beschädigte Datensätze, die durch un-

5 Geschwindigkeitsschätzung

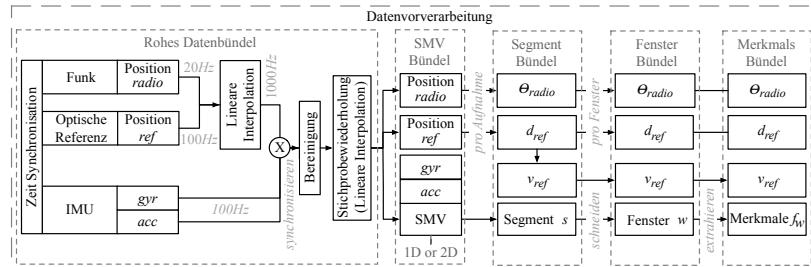


Abbildung 5.5: Eine vergrößerte Ansicht der Vorverarbeitungskette.

terbrochene oder verlorene Signale verursacht wurden, visuell und statistisch identifiziert. Kurze Aussetzer (<20 ms) werden durch erneutes Abtasten und erneutes Interpolieren behoben. Andere beschädigte Daten werden aus dem Messsatz gelöscht und durch Daten aus einer wiederholten Messung ersetzt. Um mehrere Datensätze für die Evaluation zu generieren, werden die sauberen Messsätze bei 50 Hz, 100 Hz, 200 Hz und 400 Hz erneut abgetastet. In Abschnitt 5.6.3 wird die beste Konfiguration erläutert, die die höchste Genauigkeit der Geschwindigkeits schätzung bietet. Dann werden die Signalbetragsektoren (SMV) für die sauberen, neu abgetasteten und interpolierten Daten berechnet. Da es zwei unterschiedliche Eingabestreams gibt (acc und gyr), werden die folgenden zwei Verarbeitungsoptionen untersucht: eine intuitive 2D-Darstellung:

$$SMV(acc) = \sqrt{(acc_x^2 + acc_y^2 + acc_z^2)} \quad (5.7)$$

und

$$SMV(gyr) = \sqrt{(gyr_x^2 + gyr_y^2 + gyr_z^2)}, \quad (5.8)$$

sowie eine 1D-Kombination aus acc und gyr:

$$SMV(acc, gyr) = \sqrt{(acc^2 + gyr^2)}. \quad (5.9)$$

Dies führt zu einem sogenannten SMV-Bündel, das die 1D- oder 2D-SMVs und die entsprechenden Positionen *ref* und *radio* einbettet. Aus dem SMV-Bündel wird das *Segmentbündel* (engl. segmentation bundle) erstellt, um klassische Referenzmethoden des Stands der Technik zu bewerten, die nur mit Eingabedatenreihen funktionieren, die mindestens zwei aufeinanderfolgende Spitzen, m.a.W. Schritte, enthalten. Von

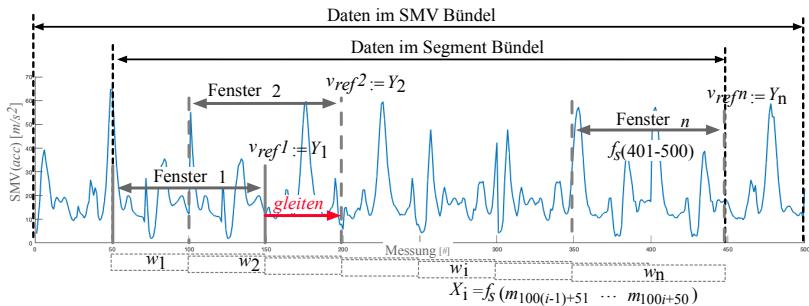


Abbildung 5.6: Beispielhafter Ausschnitt eines Segments, das einen $SMV(acc)$ einer einzelnen Gehaktivität eines Benutzers zeigt. Es werden Fenster aus einem sauberen Segment ausgeschnitten und die Referenzgeschwindigkeiten am Ende jedes Fensters berechnet.

dort aus wird das *Fensterbündel* erstellt, um die neuesten DL-Methoden des Stands der Technik und die hier vorgeschlagene Hybrid-Methode zu bewerten, und von dort aus wird das *Merkmalbündel* erstellt, um die ML-Referenzmethoden des Stands der Technik zu bewerten.

Segmentbündel

Während das SMV-Bündel das gesamte anfängliche Rohdatenbündel darstellt, enthält das Segmentbündel nur das Segment der Daten, das für die Aktivität des Benutzers von Bedeutung ist, d. h. es werden Teile am Anfang und am Ende entfernt, die möglicherweise versehentlich vor und nach der Aktivität aufgezeichnet wurden, und nur die sich wiederholenden Strecken der Bewegung beibehalten, siehe Abbildung 5.6 für ein Beispiel. Zusätzlich zu den 1D- und 2D-SMV-Daten enthält ein Segmentbündel den Referenzabstand d_{ref} und die Referenzgeschwindigkeit v_{ref} für jeden Datenpunkt. d_{ref} und v_{ref} werden durch Differenzieren der 2D-Geschwindigkeiten berechnet, d. h. der Translationsschicht, die von den aus dem Referenzsystem erhaltenen x - und y -Achsen, mit Bezug zu Zeit und Norm, aufgespannt wird. Das Segmentbündel enthält auch die Orientierungen in Polarkoordinaten $\theta_{radio} = \theta = \arctan(p(x), p(y))$. Die Positionen p werden in kartesischen Koordinaten durch die *radio*-Daten angegeben, siehe Abschnitt 5.6.1.

Fensterbündel

Das Fensterbündel ist eine andere Darstellung der Daten im entsprechenden Segmentbündel. Die Daten werden in aufeinanderfolgende Fenster aufgeteilt, die sich je nach Vergleichsmaßstab (engl. benchmark) überlappen oder nicht, siehe Abschnitt **Parameter** unten. Abbildung 5.6 skizziert den Schneidevorgang eines Segments in ein Bündel überlappender Fenster. Die Werte d_{ref} , v_{ref} und θ_{radio} in einem *Fensterbündel* werden auf die gleiche Weise wie zuvor für Segmentbündel berechnet, aber diesmal wird jedes Fenster einzeln integriert, um ein einzelnes d_{ref} , v_{ref} und θ_{radio} für jedes Fenster bereitzustellen.

Merkmalsbündel

Für jedes Fenster in einem Fensterbündel enthält das Merkmalsbündel die Merkmale f_w , die aus den Daten extrahiert werden. Abschnitt 5.6.4 erörtert Details zu den extrahierten Merkmalen.

Datensatzvarianten

Zur Bewertung der Methoden werden drei Varianten dieser Datensatzpakete erläutert. Variante V_1 ist die, wie sie im vorangegangenen Abschnitt besprochen wurde. Bei den beiden anderen Varianten werden die Datenbündel der Aktivität eines einzelnen Benutzers nicht mehr isoliert behandelt. Stattdessen werden für die Variante V_2 die Bündel der anderen Benutzer berücksichtigt, die dieselbe Aktivität ausführen. Variante V_3 führt alle Aktivitäten aller Benutzer zusammen.

Variante V_2 stellt sicher, dass alle SMV-Bündel aller Benutzer dieselbe Länge haben. Infolgedessen haben die resultierenden Segment-, Fenster- und Merkmalsbündel auch die gleiche Länge. Dazu wird das kürzeste SMV-Bündel verwendet und alle anderen SMV-Bündel auf dessen Größe beschnitten und überhängende Proben werden verworfen, d. h. etwaige zusätzliche Schritte eines Benutzers werden verworfen.

Variante V_3 verwendet alle Fenster aller Benutzeraktivitäten und sortiert sie mit Rücksicht auf die Geschwindigkeiten. Ein Bin-Packung-Algorithmus teilt die Fenster in Gruppen auf. Eine solche Gruppe enthält dann alle Fenster mit derselben Geschwindigkeit, unabhängig davon, zu welchem Benutzer sie gehören. Es werden kleine Gruppen unter 200 Einträge verworfen. Aus Gruppen von Fenstern werden Gruppen von Merkmalen wie zuvor beschrieben berechnet.

Die Varianten V_1 - V_3 werden verwendet, um die ML-Referenz, die DL-Referenz und die hybride DL-Methode zu bewerten. Während die DL-Ansätze direkt mit den Fensterbündeln, oder Gruppen in V_3 , arbeiten, funktioniert die ML-Referenz mit den Merkmalsbündeln, oder Gruppen in V_3 . Die klassische Referenzmethode funktioniert mit den Segmentbündeln und gilt nicht für V_3 , da in V_3 keine Gruppen von Segmenten vorhanden sind.

Parameter

Zusätzlich zu den Varianten gibt es weitere Konfigurationsstellschrauben, die in den Experimenten verwendet werden können: Es werden verschiedene Abtastraten f_s (50, 100, 200 und 400 Hz) und verschiedene Fenstergrößen N_w (64, 128, 256 und 512 Hz) von *acc* und *gyr* bewertet, um Kombinationen von f_s und N_w zu finden, die rechnerisch effizient sind und die genauesten Geschwindigkeitsschätzungen liefern. Weiter besteht die Möglichkeit, wenn ein Segmentbündel in Fenster aufgeteilt wird, konsekutive Fenster um $N_w/2$ überlappen zu lassen. Dadurch werden Fensterbündel mit den Längen 32, 64, 128 und 256 Hz für jedes Segmentbündel generiert.

Schließlich werden für alle Varianten alle Segmentbündel, Fensterbündel-/gruppen und Merkmalsbündel-/gruppen in Trainings-, Validierungs- und Testdatensätze aufgeteilt, um die zu evaluierenden Verfahren mithilfe dieser bekannten und beschrifteten Datensätze zu trainieren und zu evaluieren. Zur Laufzeit werden Geschwindigkeiten aus unbekannten Eingabedaten mit derselben Pipeline prognostiziert. Dies geschieht unabhängig für V_1 , V_2 und V_3 , um spezifische Forschungsfragen zu adressieren, siehe Abschnitt 1.3 für Forschungshypothesen und Abschnitt 5.7 für Antworten.

Weitere Details zum Datensatz und zur Gestaltung der Benchmark-Studie werden in Abschnitt 5.6.2 aufgeführt.

5.5.2 Ressourcenschonende Hauptverarbeitung

KNN wie mehrschichtige Perzeptrone (MLPs) bestehen aus einer Eingabeschicht X , einer oder mehreren verborgenen Schichten H und einer Ausgabeschicht Y . Eingangssignale breiten sich durch das Netz aus, eine Verlustfunktion vergleicht die Ausgabe mit der Referenzbezeichnung, um den Fehler zu berechnen, Fehlersignale breiten sich rückwärts durch das Netz aus und Gewichte werden angepasst, um den

5 Geschwindigkeitsschätzung

Fehler zu reduzieren [395]. Da alle MLP-Varianten nur eine feste Länge von Eingabemesswerten m , d. h. acc und gyr , aufnehmen, können sie im Zeitschritt i zur Vorhersage einer Geschwindigkeit v_i die Beziehung der Zeit und Kontext über aufeinanderfolgende Messungen nicht beschreiben. Diese sind jedoch für die Beschreibung von Bewegung relevant, da es sich um einen zeitlich-dynamischen Prozess handelt, der von verschiedenen Faktoren wie Beschleunigung, Geschwindigkeit und Richtung abhängt.

Daher wird im Rahmen dieser Forschungsarbeit vorgeschlagen, zeit- und kontextsensitive KNN zu verwenden, um solche Abhängigkeiten innerhalb eines einzelnen Fensters und über mehrere Fenster hinweg zu lernen. Da RNN wie LSTM oder bidirektionale LSTM-Netze (BLSTMs) sowohl kurzfristige Abhängigkeiten, innerhalb eines einzelnen Fensters, als auch langfristige Abhängigkeiten, zwischen Fenstern, lernen, erfassen Sie (a) die Beziehung jeder Messung in einem Fenster zu allen anderen Messungen in demselben Fenster, d. h. wie sich jede Messung auf andere Messungen in demselben Fenster auswirkt, und (b) ihre Abhängigkeiten innerhalb des gesamten Segments, d. h. wie jede Messung die Messungen in einem anderen Fenster beeinflusst, z. B. wie sich Bewegungen zwischen Fenstern ändern [127]. In Kapitel 4 wird gezeigt, dass dies auch für extrem langfristige Abhängigkeiten in extrem langen Trajektorien gilt.

Ein naheliegender weitverbreiteter Ansatz besteht darin, handgefertigte Merkmale (siehe Abschnitt 5.6) als Eingabe für das RNN zu verwenden. Jedoch erweisen sich CNN als viel bessere Merkmalsextraktoren, da sie die Merkmale während des Trainings vollautomatisch extrahieren und dabei direkt die optimalen Merkmale identifizieren, die am besten zur Lösung des Optimierungsproblems beitragen, und den Geschwindigkeitssfehler für jede Eingabesequenz reduzieren. Allerdings können CNNs diese Merkmale nur in einen räumlichen Kontext, bspw. zur Extraktion von Merkmalen in Bildern und nicht in einen zeitlichen Kontext, bspw. wie Bildinformationen aufeinander wirken, setzen. Um räumliche Merkmale optimal zu extrahieren und diese optimal in einen zeitlichen Kontext zusetzen, wird daher ein Modell vorgeschlagen, das ein CNN verwendet, um High-Level-Merkmale aus rohen Eingangssignalen zu extrahieren und sie über die Zeit mit einem RNN-Modell zu verbinden.

Um latente Verbindungen zwischen Bewegungseigenschaften und Datenmerkmalen wiederherzustellen, kann ein LSTM zeitliche Abhängigkeiten ausnutzen, indem verborgene Zustände über die Dauer eines Fensters beibehalten werden. Das Netz fungiert effektiv als eine Funktion f , die Sensormessungen über ein Fenster abbildet. Die LSTM-Architektur wird um die bidirektionale LSTM-Architektur (BLSTM) erweitert, die den dynamischen Kontext nutzt und das aktuelle Fenster sowohl aus der Vergangenheit als auch aus der Zukunft untersucht, um eine höhere Darstellung der IMU-Daten zu berücksichtigen.

Lang-Kurzzeitgedächtnis-(LSTM)-Netze

Die Eingabedaten sind unabhängige Fenster aufeinanderfolgender, stark zeitabhängiger IMU-Messungen, die Körperbewegungen mit unterschiedlichen Geschwindigkeiten darstellen. Um latente Verbindungen zwischen Bewegungseigenschaften und Datenmerkmalen wiederherzustellen, kann ein RNN zeitliche Abhängigkeiten ausnutzen, indem verborgene Zustände über die Dauer eines Fensters beibehalten werden. Zu beachten ist jedoch, dass latente Zustände nicht zwischen Fenstern übertragen werden. Das LSTM fungiert effektiv als eine Funktion f , die Sensormessungen über ein Fenster abbildet. Es werden unterschiedliche Abtastraten (50, 100 Hz) und Fensterlängen (64, 128 Hz) verwendet. Letztere entsprechen einer einzelnen Sequenzlänge pro Iteration, die sich auf 2.56 s (128/50 Hz) summiert.

In einem realen physikalischen Modell wirken sich Bewegungstransformationen auf alle nachfolgenden Ausgaben aus. Es wird daher das LSTM verwendet, um langfristige Fenstersegmentabhängigkeiten zu modellieren und die Probleme mit explodierenden und verschwindenden Gradienten der ursprünglichen-RNN Architektur zu bewältigen, da es Langzeitabhängigkeiten besser nutzen kann [155]. Da sowohl frühere als auch zukünftige Fenster und Segmente für die Aktualisierung des aktuellen Status von entscheidender Bedeutung sind, wird die BLSTM untersucht, die den dynamischen Kontext nutzt und das aktuelle Fenster sowohl aus der Vergangenheit als auch aus der Zukunft untersucht, um eine genauere Darstellung von IMU-Daten zu ermöglichen. Eine vorläufige Studie zeigt, dass es nützlich ist, mindestens zwei Schichten vertikal zu stapeln [331, 154], wobei die Ausgabesequenzen der ersten Schicht die Eingabesequenzen der nächsten Schicht liefern. Die letzte Schicht gibt einen Geschwindigkeitsvektor aus, um die Transformationsbeziehung in der verarbeiteten Sequenz

5 Geschwindigkeitsschätzung

darzustellen. Jede Schicht wird in Bezug auf 64, 128, 256 verborgene Zustände optimiert. Es wird die letzte Geschwindigkeitsschätzung Y^n als endgültige Geschwindigkeitsschätzung pro Fenster verwendet.

Ein tiefes, hierarchisches Modell kann einige Funktionen exponentiell effizienter darstellen als ein flaches. Das Erhöhen der Tiefe des Netzes erfordert weniger Neuronen und trainiert schneller. Letztendlich ist das Hinzufügen von Tiefe eine Art Repräsentationsoptimierung. Erstellen eines tiefen RNN durch Stapeln mehrerer rekurrenter verborgener Zustände übereinander ermöglicht es, dass der verborgene Zustand auf jeder Schicht zu unterschiedlichen Zeiten arbeitet. Angesichts der Tatsache, dass LSTMs mit Sequenzdaten arbeiten, bedeutet dies, dass durch Hinzufügen von Schichten die Abstraktionsebenen der Eingabebobachtungen im Laufe der Zeit erhöht werden. Tatsächlich werden Beobachtungen im Laufe der Zeit aufgeteilt oder das Problem auf verschiedenen Zeitskalen dargestellt. Tiefere LSTM Architekturen benötigen also weniger Neuronen, trainieren daher schneller und können ein Problem auf unterschiedlichen Zeitskalen lösen [331]. RNNs sind von Natur aus zeitlich tief, da ihr verborgener Zustand eine Funktion aller vorherigen verborgenen Zustände ist. Außerdem profitieren RNNs von der Tiefe im Raum, d. h. vom Stapeln mehrerer rekurrenter verborgener Schichten. Graves et al. haben gezeigt, dass die Tiefe des Netzes wichtiger ist als die Anzahl der Speicherzellen in einer bestimmten Schicht, um Fähigkeiten zu modellieren [154].

Während der Trainingsphase lernt das Modell M , latente Bewegungseigenschaften aus aufeinanderfolgenden Sequenzen von Datenpunkten pro Fenster v_i und ihrem entsprechenden Referenzgeschwindigkeitswert v_{ref} zu nutzen, um eine genaue und präzise Schätzung der aktuellen Geschwindigkeit eines Subjekts bereitzustellen an Sätzen von rohen Beschleunigungs- und Gyroskopmessungen während der Live-Phase.

Hintergrund

Während typischerweise MLP entworfen werden, die einen Stapel vergangener Messungen verwenden, um v^i vorherzusagen, sind RNN die bessere Wahl [92]. RNN sind eine Klasse KNN, die Datensequenzen verarbeiten (hier: aufeinanderfolgende Sätze von Beschleunigungs- und Gyroskopmessungen) und erfolgreich auf verschiedene zeit- und kontextsensitive Aufgaben angewendet werden. RNNs haben persistente interne Zustände c_i , den Speicherzustand der Zelle, die von der Einga-

besequenz X_i aktualisiert werden und die über Übergänge übertragen werden, z. B. von H_i zu H_{i+1} . Dies ermöglicht es dem Modell, dynamisches Verhalten aus den Eingabedaten zu erfassen und es zur Vorhersage der Ausgabe Y^i zu verwenden. Da RNNs in der Praxis unter verschwindenden oder explodierenden Gradienten leiden [36], erkennen sie kaum Abhängigkeiten über große Entfernnungen. LSTM-Netze vermeiden dieses Problem [191], da sie die Berechnung von H_i durch eine Zelle ersetzen, die drei Gatter enthält: ein Vergessensgatter f_i , ein Eingangsgatter i_i und ein Ausgangsgatter o_i im Zeitschritt i . Diese Zellen modellieren explizit den Informationsfluss zwischen aufeinanderfolgenden Zeitschritten. Abbildung 3.27 in Kapitel 3 zeigt die interne Struktur einer Basis-LSTM-Zelle. Der Speicherzustand c_i , sogenannter Kontextvektor (engl. context vector), der Zelle wird in jedem Schritt mit Informationen sowohl vom Vergessen als auch vom Eingangsgatter aktualisiert. H_i ist der neue Zustand, der durch das Ausgangsgatter aus der Speicherzelle c_i extrahiert wird. Informationen fließen durch das LSTM von links nach rechts und von unten nach oben. Die anfängliche Eingabe X_i geht in die niedrigste LSTM-Schicht $l=0(H_l=o,i)$. Jede Zelle $H_{l,i}$ speist auch die benachbarte Zelle auf derselben Schicht $H_{l,i+1}$ von der Seite und die benachbarte Zelle auf der nächsthöheren Schicht $H_{l+1,i}$ von unten. Die Breite eines Modells wird durch die Größe einer LSTM-Zelle definiert. Die Größe einer einzelnen Zelle wird durch die Anzahl der möglichen Gewichte und Bias definiert, die jede Zelle verarbeiten kann. In Abbildung 3.27 in Kapitel 3 repräsentiert jeder Kreis eine fest bezeichnete Funktionalität, verkörpert aber auch einen Stapel von Gewichten und Bias gemäß der Zellengröße, d. h. eine Zellengröße von 128 bedeutet, dass jeder Kreis in der Zelle 128 Gewichte und 128 Fehlerschätzungen hält.

Von LSTM zu BLSTM

Während Abbildung 3.27 in Kapitel 3 eine grundlegende LSTM-Zelle zeigt, die auch in BLSTMs verwendet wird, zeigt Abbildung 3.35 in Kapitel 3 die Verbindung zwischen Schichten in einem gestapelten LSTM-Modell. Zur Vereinfachung für jeden Zeitschritt i (Spalte) wurden beide Verwerfungs-Schichten zwischen jeder der verborgenen Schichten H_0 bis H_I weggelassen, aber die dichte, vollständig verbundene Schicht kurz vor der Ausgabeschicht Y belassen. Während der Trainingsphase nimmt das Modell Schnitte (Sequenzen) aufeinanderfolgender Messwerte, eines einzelnen Fensters, $v_{i-s:i}$ und deren

5 Geschwindigkeitsschätzung

entsprechenden Referenzgeschwindigkeitswert $r_{i-s:i}$ als Eingabe. Die Sequenzlänge s , d. h. die Fensterlänge, definiert, wie viele aufeinanderfolgende Messwerte das Modell in jeder Iteration verarbeitet. Je länger die Sequenz ist, desto mehr Bewegungsabhängigkeiten lernt das Modell richtig zu interpretieren, aber desto komplexer wird das Modell.

Das Modell lernt, wie man eine einzelne Geschwindigkeit aus einem einzelnen Satz von Messwerten erhält und wie man aufeinanderfolgende Geschwindigkeiten vorhersagt. Während X_i eine Eingabe darstellt, die zu den Zeitschritten i gesetzt wird, ist die resultierende Geschwindigkeitsschätzung Y^i . Mit einem zufällig initialisierten Anfangszustand H_0 hat das Modell keine Vorkenntnisse über die Bewegung und lernt folglich, die Geschwindigkeit aus lokalen Informationen zu schätzen. Als Ausgabe Y schätzt das Modell die aktuelle Geschwindigkeit v eines Subjekts. Wichtig zu beachten ist, dass die Länge der Eingabesequenz zwar die potentielle Länge der Abhängigkeiten in der Sequenz erhöht, mit der Länge aber auch die Komplexität der Information ansteigt und somit die notwendige Kapazität des LSTM ansteigt. Mit steigender Kapazität verliert der Kontextvektor des LSTM allerdings seine Aussagekraft, da der Vektor über alle verborgenen Schichten Informationen sammelt und verteilt. Je größer das Modell, desto verwischter werden also die durch den Kontextvektor repräsentierten Informationen. Folglich ist die korrekte Wahl der Eingabesequenzlänge und Kapazität des LSTM eine entscheidende Stellschraube in der Wahl der optimalen Architektur für ein zugrundeliegendes spezifisches Problem.

Die ressourcenschonende CNN-BLSTM Architektur

Abbildung 3.27 in Kapitel 3 zeigt eine LSTM-Zelle (H_l), die von BLSTM verwendet wird. Siehe auch die farbigen Kreise (H_l) in Abbildung 5.7. Die CNN-BLSTM-Architektur wird so entworfen, dass sie Merkmale auf hoher Ebene aus Faltungsschichten extrahieren und zeit- und kontextsensitive Merkmale, d. h. Merkmalskarten auswendig lernen, in der BLSTM-Schicht komprimieren und verfolgen kann.

Abbildung 5.7 zeigt die Architektur des ressourcenschonenden CNN-BLSTM-Netzes. Daten breiten sich von unten nach oben aus. Nach einer anfänglichen Entfaltung der Daten aus einem Stapel mit Eingabemessungen extrahieren 1D-Faltungskerne Merkmale auf hoher Ebene, während ein 1D-Filterkern f_k über eine Sequenz, d. h. den

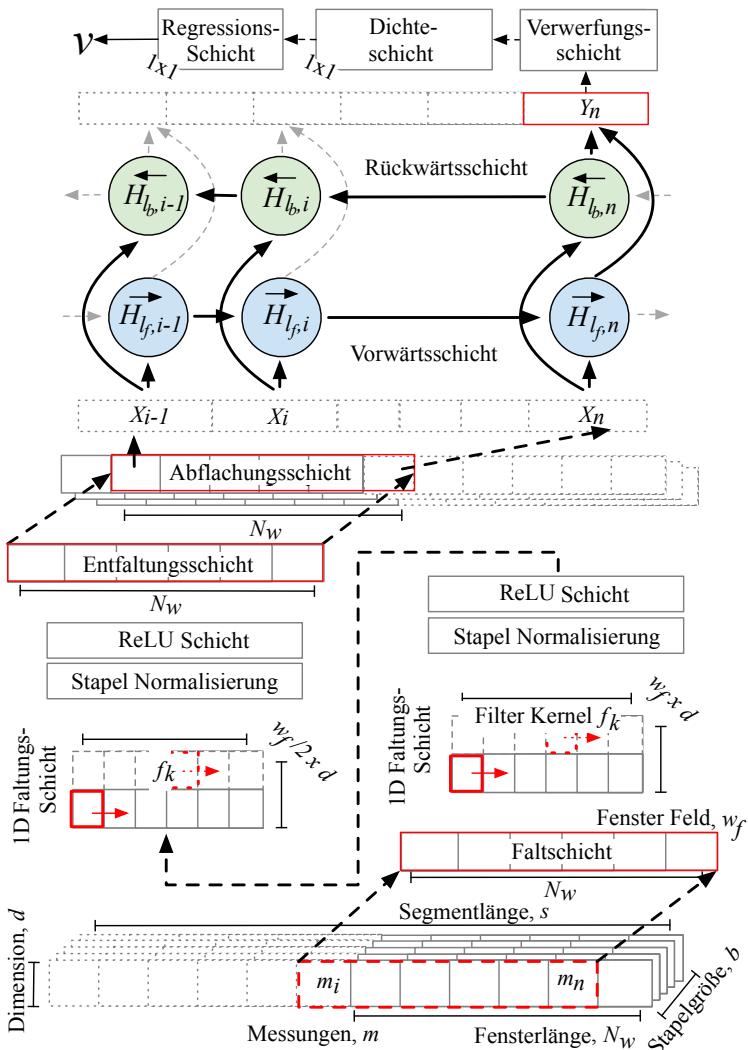


Abbildung 5.7: Die Architektur des ressourcenschonenden CNN-BLSTM.

5 Geschwindigkeitsschätzung

Fenstervektor w_v , gleitet, um Merkmale an verschiedenen Positionen zu erkennen [221]. Die Idee hinter der 1D-Faltung besteht darin, eine Element-für-Element-Multiplikation eines 1D-Filterkerns f_k mit jedem 1D w_v in d -Fenstern durchzuführen, um eine zeitkritische Merkmalskarte [264] zu erhalten. Es werden zwei solcher Schichten verwendet, um Funktionen auf hoher Ebene zu erhalten. Im Detail durchlaufen die Eingabemessungen das Netz von unten nach oben: Zuerst wird ein Fenster mit der Dimension $d \times N_w$ zu einem entsprechenden Array ($w_f \times d$) gefaltet. Dazu wendet die erste 1D-Faltungsschicht $k=128$ Filterkerne $f_k=N_w$ mit der Größe 1×3 an, d. h. $\#k=N_w$. Im Rahmen von Forschungsarbeiten dieser Dissertation wird festgestellt, dass ein Filterkern f_k , d. h. ein Merkmal [221] pro Messwert m genaue Ergebnisse liefert, gefolgt von einer Stapelnormalisierung- (BN) und einer ReLU-Schicht. Die zweite 2D-Faltungsschicht ($w_f/2 \times d$) wendet N_w Filterkerne mit der Größe 1×3 an, gefolgt von BN und ReLU. Wenn $d==2$, projiziert eine sich entfaltende Schicht die Merkmalslandkarte (engl. feature-map) zurück zu ihrer ursprünglichen Sequenzstruktur gemäß den Informationen des Stapels, die durch die (nicht) faltbaren Schichten geleitet werden. Eine Abflachungsebene liefert eine flache Sequenz für die Eingabeebene des BLSTM. Die letzte Schicht des BLSTM fungiert als Engpass und liefert das neueste Y_n , das durch eine Verwerfungsschicht verarbeitet wird. Eine vollständig verbundene Schicht liefert eine 1×1 -Ausgabe an die Regressionsschicht, die den Verlust durch den MSE berechnet.

Nachdem die High-Level-Merkmale aus dem SMV extrahiert wurden, werden sie ohne zusätzliche Gruppierung (engl. Pooling) direkt in das BLSTM eingefügt. Maxima- und Durchschnitts-Gruppierung entfernen wertvolle sequentielle Organisation [264, 412]. Somit verfolgt das BLSTM die Entstehung von Merkmalen im Laufe der Zeit. Es werden Engpassmerkmale mit einer Viele-zu-Eins-BLSTM-Architektur extrahiert, d. h. $X > Y = 1$. Im Gegensatz zu LSTM verarbeitet das BLSTM Informationen über die Zukunft und die Vergangenheit, indem es kurzfristige langfristige Eingabekontexte für die aktuelle Ausgabe sowohl in Vorwärts- als auch in Rückwärtsrichtung abruft [221]. Das BLSTM fungiert als zeit- und kontextsensitiver Detektor und Verfolger von Bewegungsmerkmalen auf hoher Ebene, die die Faltungsschichten aus der Signaleinbettung (acc und gyr) extrahieren.

Zwei verborgene Schichten (H_l und H_{l+1}) bieten Vorwärts- und Rückwärtsrichtungen. In Anlehnung an frühere Arbeiten [264] wird die

letzte verborgene Schicht als Engpass-Schicht verwendet. Zu beachten ist, dass die Sequenzlänge s definiert, wie viele aufeinanderfolgende Messwerte m das Modell in jeder Iteration verarbeitet. Je länger die Sequenz ist, desto mehr Bewegungsabhängigkeiten lernt das Modell richtig zu interpretieren, aber desto komplexer wird das Modell [127]. Das hier vorgeschlagene Modell lernt daher, wie man aus einem einzigen Satz von Messwerten eine einzelne Geschwindigkeit erhält und aufeinanderfolgende Geschwindigkeiten vorhersagt. Während X_i eine im Zeitschritt i festgelegte Eingabe darstellt, ist die resultierende Geschwindigkeitsschätzung Y_i . Bei einem zufällig initialisierten Anfangszustand H_0 hat das Modell keine Vorkenntnisse über die Bewegung und lernt folglich, die Geschwindigkeit aus lokalen Informationen zu schätzen. Während der Trainingsphase lernt das Modell, latente Bewegungseigenschaften aus einer aufeinanderfolgenden Sequenz $s=w$ von Datenpunkten m pro Fenster w_i und ihrem entsprechenden Referenzgeschwindigkeitswert v_{ref} zu nutzen. Zur Laufzeit wird die aktuelle genaue Geschwindigkeit eines Subjekts in einem Fenster w_i mit rohen acc - und gyr -Messungen vorhergesagt. Die Regressionsschicht schätzt die (aktuellste) Geschwindigkeit Y_n , die als die endgültige Geschwindigkeitsschätzung v pro Fenster gewählt wird.

5.5.3 Hochgenaue Hauptverarbeitung

In diesem Abschnitt wird die hybride DL-Methode (Hybrid) im Detail beschrieben. Details zu den klassischen und den ML-Referenzmethoden werden in Abschnitt 5.6.4 aufgeführt.

Bevor die Details der neuartigen Hybrid DL-Architektur beschrieben werden, werden zunächst einige wichtige Notationen und Definitionen vorgestellt. Unterhalb der x-Achse zeigt Abbildung 5.6, wie die Fenster aus der Vorverarbeitung in Sequenzen umgewandelt werden, die als Eingabe in das Netz dienen. Im Beispiel enthält das Fenster w 100 Messungen m , die in 100 Zeitschritten durchgeführt werden. Während die Verfahren in der Vorverarbeitung mit den einzelnen Messdaten arbeiten und den Index i für jeden Messpunkt verwenden, um auf dessen Zeitstempel zu verweisen, wird jetzt abstrakter mit vollständigen Fenstern, die mehrere Messpunkte enthalten, gearbeitet, sodass sich der Index i auf die Nummer eines Fensters bezieht.

X_i repräsentiert dann das Fenster w_i mit all seinen m -Messungen. X_i repräsentiert die Eingabe der Methoden. Es enthält alle Funktionen

5 Geschwindigkeitsschätzung

f_s für $m \in [100 \cdot (i-1) + 51, 100 \cdot i + 50]$. Y_i ist die Ausgabe des KNN, d. h. die Referenzgeschwindigkeit v_{ref} während der Trainingsphase oder die vorhergesagte Geschwindigkeit v' zur Laufzeit.

Hintergrund

KNN wie MLP bestehen aus einer Eingabeschicht X , einer oder mehreren verborgenen Schichten H und einer Ausgabeschicht Y . Eingangssignale breiten sich durch das Netz aus, eine Verlustfunktion vergleicht den Ausgang mit der Referenzbezeichnung, um den Fehler zu berechnen, Fehlersignale breiten sich rückwärts durch das Netz aus und Gewichte werden angepasst, um den Fehler zu reduzieren [395]. Alle MLPs nehmen nur einen festen Satz von Eingabemessungen m , z. B. *acc* und *gyr* Datenpunkte, pro Zeitschritt i (eine Momentaufnahme) entgegen. Daher können sie die Beziehung von Zeit und Kontext (zeitliche Merkmale f_t) über aufeinanderfolgende Eingaben nicht beschreiben, um eine Geschwindigkeit v_i vorherzusagen. Stattdessen beschreiben sie räumliche Merkmale $f_s(X_i)$ eines Schnappschusses, die keine Verbindung oder Beziehung zu einem vorherigen $f_s(X_{i-1})$ oder einem nachfolgenden Schnappschuss $f_s(X_{i+1})$ haben.

Im Rahmen der Dissertation hat sich herausgestellt, dass diese Beziehungen relevant sind, da sie Änderungen im Laufe der Zeit darstellen, z. B. die Bewegungsänderungen aufgrund verschiedener Faktoren wie Beschleunigung, Geschwindigkeit und Richtung.

Rekurrente neuronale Netze (RNN)

Daher werden zeit- und kontextsensitive KNN verwendet, um Abhängigkeiten von f_s in einem einzigen Fenster und ihre kurz- und langfristigen Abhängigkeiten f_t zu lernen. LSTM-Netze oder BLSTM-Netze können solche zeitlichen Abhängigkeiten f_t erfassen und speichern.

Daher erfassen sie (1) die Beziehung jeder Messung m_j in einem Fenster w_i zu allen anderen $m_k \in w_i$ in demselben Fenster, d. h. wie jede Messung andere Messungen in demselben Fenster beeinflusst, und (2) merken sie sich deren Abhängigkeiten und Auswirkungen, d. h. wie sich jede Messung über einen langen Zeitraum auf andere Messungen auswirkt, z. B. wie eine Bewegung f_s mit der gleichen Geschwindigkeit für alle Benutzer unterschiedliche f_t zwischen sich hat [127]. Für einen fairen Vergleich werden Eingabesequenzen mit einem einzigen Fenster verarbeitet, um Rechenaufwand einzusparen und das Verfahren mit

dem Stand der Technik und den Referenzmethoden zu vergleichen (Abschnitt 5.6 beschreibt Auswirkungen der Eingabesequenz).

Um latente Verbindungen zwischen Bewegungs- und Dateneigenschaften zu rekonstruieren, nutzt ein BLSTM außerdem zeitliche Abhängigkeiten f_t , da es seine verborgenen Zustände über die Dauer eines Fensters beibehält. f_t ermöglicht es auch den dynamischen Kontext in den Daten zu nutzen und BLSTM untersucht das aktuelle Fenster sowohl aus der Vergangenheit als auch aus der Zukunft. Das NN fungiert effektiv als eine Funktion f , die zusammenhängende Sensormessungen eines Fensters, m.a.W. einer Sequenz von Datenpunkten, auf eine entsprechende Geschwindigkeit abbildet.

In den Ergebnissen, siehe Abschnitt 5.7, wird gezeigt, dass diese Art von RNN in Kombination mit einer hochmodernen Merkmalsextraktion die langfristigen Abhängigkeiten f_t der Datenpunkte in den Fenstern von Stapeln, die mehrere zufällig aneinander gereihte Fenster einer Bewegung nutzen kann. Die Experimente zeigen, dass das Fehlen von f_t zu deutlich schlechteren Ergebnissen führt.

Merkalsextraktion

RNNs verfolgen die Signaleigenschaften über einen längeren Zeitraum hinweg gut. Daher muss der beste Weg gefunden werden, um solche Eigenschaften aus Rohsignalen abzuleiten. Ein naheliegender Ansatz besteht darin, handgefertigte Merkmale, die für die ML-Methoden verwendet werden, siehe Abschnitt 5.6, zu verwenden und sie als Eingabe für das rekurrente Netz zu verwenden.

Dabei erwiesen sich CNNs als viel bessere Extraktoren für räumliche Merkmale, obwohl sie diese Merkmale nicht in einen zeitlichen Kontext stellen können. Abschnitt 5.5.2 führt eine flache 1D-CNN-Architektur ein, die geeignet ist, um möglichst effizient (Rechenaufwand gegenüber Genauigkeit) f_s aus 1D-Eingaben zu extrahieren [128]. Es ist bekannt, dass breitere oder tiefere CNNs charakteristischere und repräsentativere f_s extrahieren können, die zu einer höheren Genauigkeit der Schätzungen führen können [504]. Aber je tiefer das CNN ist, desto mehr leidet es unter einem verschwindenden Gradienten und desto höher ist seine Rechenlast [43]. Restnetze mit mehreren Faltungsschichten (wie ResNet-18 [176]) lösen dieses Problem. Ihre extrem tiefen Netze leiden nicht unter verschwindenden Gradienten und sind in verschiedenen Anwendungen genau und rechnerisch effizient.

Abbildung 5.8 zeigt, dass unabhängig vom Restnetztyp der Gradient \hat{X}_i zuerst dupliziert wird. Während Restnetze vom Typ I die Merkmale \hat{f}_s nur in einer der Kopien \hat{X}_i falten, falten Netze vom Typ II beide \hat{X}_i , bevor sie am Ende die Ergebnisse beider Pfade addieren. Somit parametrisieren Resteinheiten (engl. residual units) vom Typ II die Faltungsschichten neu, z. B. Schrittänge S , Filterkerngröße F_s und Anzahl der Filterkerne F . Beide Typen nutzen auch Stapel-Normalisierungs-Schichten, zwischen Faltungsschichten, und ReLUs, um das Training zu beschleunigen und um die Empfindlichkeit der Modelle auf ungünstige Initialisierung zu verringern. Im Allgemeinen beginnen Restnetze mit einer moderaten Anzahl von Merkmalen f_s am Eingang ($=X_i$). Mit zunehmender Tiefe eines Netzes wächst die Dimensionalität von f_s .¹ Zu beachten ist, dass in der obigen Beschreibung \hat{X}_i , \hat{X}'_i , und \hat{f}_s für Eingaben, Ausgaben und Merkmale verwendet werden, die sich im NN ausbreiten. Das $\hat{\cdot}$ Symbol wird verwendet, um Zwischenwerte (engl. intermediate values) anzugeben. Größen, Dimensionalität und Merkmalssätze unterscheiden sich normalerweise erheblich zwischen den Schichten.

Architektur

Die vorgeschlagene Hybridarchitektur verwendet ein modifiziertes ResNet, um hochdimensionale \hat{f}_s aus den rohen Eingangssignalen zu extrahieren, ein einzelnes Fenster von $f_s(X_i)$ für jeden Zeitschritt i . Diese werden dann in einen sogenannten zeitlichen Merkmalsverfolger eingespeist, der aus zwei BLSTMs aufgebaut ist. Abbildung 5.9 zeigt die Architektur. Es ist zu beachten, dass die Gradienten von links nach rechts fließen, d. h. von der Eingabe X_i ($=w_i$) zur Ausgabe Y_i ($=v_{ref}$). Im Gegensatz zum ursprünglichen ResNet-18 [176] (1) wird die erste Faltungsschicht an die optimalen Eingabedimensionen angepasst, (2) am Ende des sogenannten räumlichen Merkmalsextraktors werden die Softmax- und Klassifizierungs-Schichten durch eine Abflachungsschicht (engl. flatten layer) ersetzt, und (3) am Ende zwei zusätzliche BLSTM-Schichten, eine vollständig verbundene (dichte) Schicht (engl. fully connected dense layer) und eine letzte Regressionsschicht (engl. regression layer) hinzugefügt.

¹Das Restnetz ResNet-18 enthält für bestimmte Konfigurationen bis zu etwa 10 Millionen aktive Parametern.

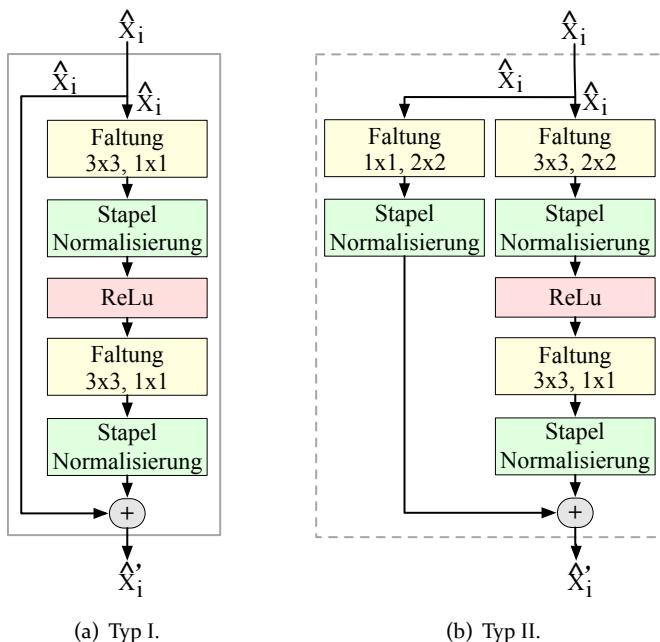


Abbildung 5.8: Architektur von Resteinheiten.

5 Geschwindigkeitsschätzung

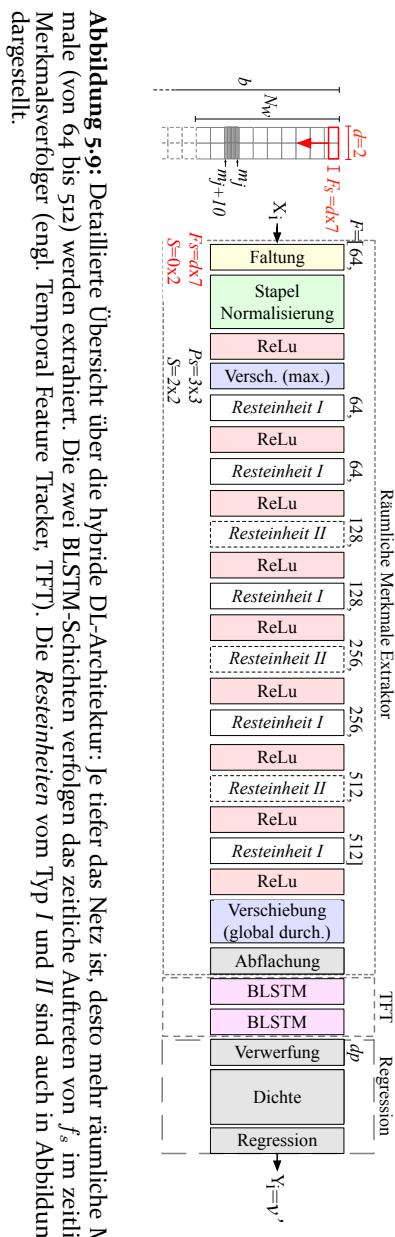


Abbildung 5.9: Detaillierte Übersicht über die hybride DL-Architektur: Je tiefer das Netz ist, desto mehr räumliche Merkmale (von 64 bis 512) werden extrahiert. Die zwei BLSTM-Schichten verfolgen das zeitliche Auftreten von f_s im zeitlichen Merkmalsverfolger (engl. Temporal Feature Tracker, TFT). Die Resteinheiten vom Typ I und II sind auch in Abbildung 5.8 dargestellt.

Die erste Faltungs-Schicht wendet 64 schrittweise Faltungsfilter ($F=64$) mit der Größe $F_s=d \times 7$ auf die Eingabe X_i von w_i an, um sie zu falten. Es schreitet jedes F vertikal mit Schrittlänge $S=1x2$ durch die Eingabe. Abbildung 5.9 zeigt dies in rot für $d=2$, $S=1x2$ und $F_s=2x7$: Es werden 14 Messungen aus einem Fenster verwendet, das Punktprodukt der Gewichte und der Messungen wird berechnet und dann ein Verzerrungsterm (engl. bias term) hinzugefügt. Die Faltungs-Schicht gibt einen mehrdimensionalen Satz von Werten \hat{X}_i aus und enthält \hat{f}_s . Eine Stapel-Normalisierungs-Schicht normalisiert dann jeden Eingangskanal über den gesamten Eingabestapel. Als Aktivierungsfunktionen wird eine ReLu-Schicht angewendet, die alle Elemente der Eingabe, die unter Null liegen, auf Null setzt, um dem Netz Nichtlinearität hinzuzufügen.² Die erste Gruppierungsschicht unterteilt die Eingabe in rechteckige Bereiche (mit der Größe $P=3x3$) und berechnet das Maximum jeder Region, um die Eingabe abzutasten, d. h. um die räumliche Größe der Darstellung zu verringern und um somit die Anzahl der Parameter und Berechnungen zu reduzieren. Das Einbetten von Restnetzen in die Struktur eines NN ermöglicht die Stabilisierung des Gradienten und vermeidet somit das Verschwinden des Gradienten, da dieser in jeder Resteinheit über den linken Pfad unberührt mitgeführt wird.

Dann werden mehrere Rest-Schichten der Typen I und II verkettet, um die Anzahl potenzieller interner Merkmale \hat{f}_s (von 64 auf 512) zu erhöhen. Eine globale Maximagruppierungsschicht (engl. global max pooling layer) berechnet das Maximum der Höhe- und Breite der Eingabedimension, um sie niedriger abzutasten, d. h. um \hat{f}_s zu reduzieren, um nur markante Merkmale beizubehalten. Eine Abflachungsschicht bietet eine flache Sequenz von \hat{f}_s , um den Eingabedimensionen der ersten nachfolgenden BLSTM-Schicht zu entsprechen. Sie reduziert die hohen räumlichen Dimensionen der Eingabe \hat{f}_s in die Kanaldimension \hat{x}_i der BLSTM, d. h. Reduktion von der mehrdimensionalen Ausgabe der Gruppierungsschicht \hat{f}_s zum 1D-Eingang $\hat{x}_{(1,n)}$ der BLSTM-Schicht. Abbildung 5.10 skizziert das Ende der Architektur: von der Abflachungsschicht bis zum BLSTM-Netz. Hier repräsentieren \hat{x} und \hat{y} (Kleinbuchstaben) die Ein- und Ausgabe der BLSTM-Schichten. Für die erste BLSTM-Schicht ist \hat{x} die abgeflachte Darstellung von \hat{f}_s ,

²Neuronale Netze verwenden nichtlineare Aktivierungsfunktionen, um komplexe Zuordnungen zwischen den Ein- und Ausgängen des Netzes zu erstellen, die zum Lernen und Modellieren komplexer nichtlinearer Daten wie z. B. Text-, Bild-, Ton-, Film- und Sensorsignale oder hochdimensionale Daten erforderlich sind.

5 Geschwindigkeitsschätzung

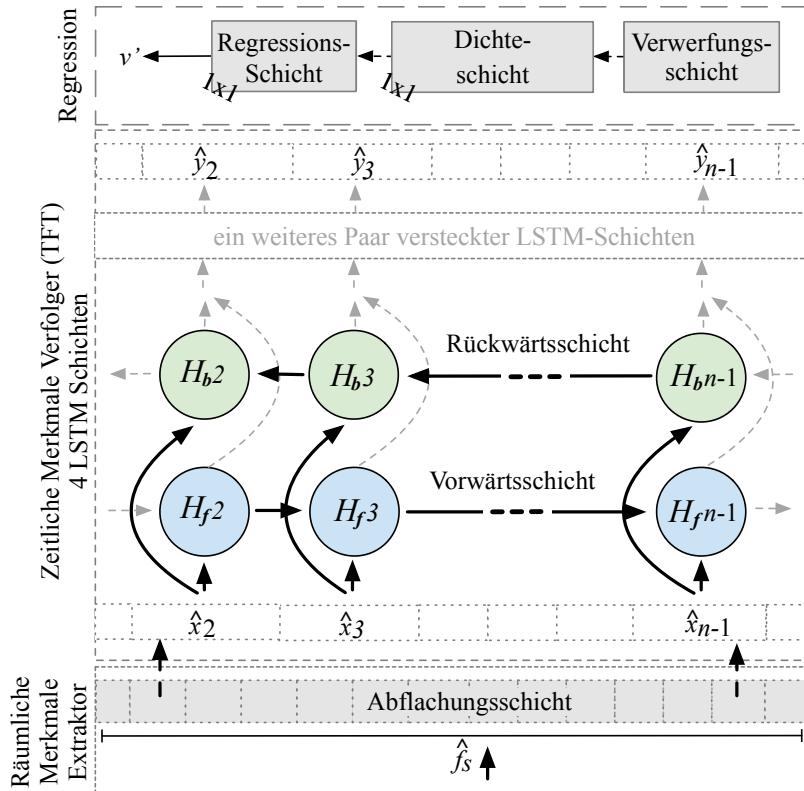


Abbildung 5.10: Eine vergrößerte Übersicht über die Architektur des zweischichtigen BLSTM-Netzes. Mit zwei beispielhaften verborgenen forward (H_f) und backward (H_b) LSTM-Schichten (l) (= 4-LSTM- für 2-BLSTM-Schichten). Die LSTM-Zellen des BLSTM, um Daten aus der Vergangenheit und Zukunft zu betrachten, sind beschnitten: die Zellen ganz links $\vec{H}_{f,l}, 1$ und $\vec{H}_{b,l}, 1$ (Vergangenheit) und die Zellen ganz rechts $\vec{H}_{f,l}, n$ und $\vec{H}_{b,l}, n$ (Zukunft). Die zweite BLSTM-Schicht wurde abgeschnitten, d. h. zwei $\vec{H}_{f,l}$ - und $\vec{H}_{b,l}$ LSTM-Schichten. Die gestrichelten grauen Pfeile über \vec{H}_b geben die Verbindung zur zweiten BLSTM-Schicht bzw. zur nächsten $\vec{H}_{f,l}$ LSTM-Vorförwärtsrichtung an. Der Pfeil auf der linken und rechten Seite zeigt die Verbindungen zu den ersten (vergangenen) und letzten (zukünftigen) LSTM-Zellen an.

d. h. es gibt erneut eine weitere Änderung der Darstellung: \hat{x}_1 zu \hat{x}_n repräsentiert das abgeflachte \hat{f}_{s1} zu \hat{f}_{sn} .

Die zwei BLSTM-Schichten lernen bidirektionale Langzeitabhängigkeiten f_t zwischen den \hat{f}_s , die mit ResNet extrahiert werden, d. h. lernen zwischen verschiedenen Bewegungen zu unterscheiden, die dieselbe Geschwindigkeit darstellen. Jede der beiden BLSTM-Schichten besteht aus einer einzelnen verborgenen vorwärts (engl. forward) LSTM-Schicht (H_f) und einer rückwärts (engl. backward) LSTM-Schicht (H_b) ($= 4 \cdot$ LSTM- für $2 \cdot$ BLSTM-Schichten). Der Einfachheit halber zeigt Abbildung 5.10 nur die ersten beiden der 4 LSTM-Schichten. Die grauen Pfeile deuten die Fähigkeit von BLSTM an, die Daten sowohl aus der Vergangenheit als auch aus der Zukunft zu betrachten. Für eine bessere Lesbarkeit gibt es keine LSTM-Zellen für \hat{x}_1 (die Vergangenheit) und \hat{x}_n (die Zukunft). Die Pfeile auf der linken und rechten Seite zeigen die Verbindungen zu den ersten (vergangenen) und letzten (zukünftigen) LSTM-Zellen an. Die gestrichelten grauen Pfeile (über \vec{H}_b) geben die Verbindung zur zweiten BLSTM-Schicht bzw. zur nächsten \vec{H}_f / H Vorwärtsschicht an. Die letzte BLSTM-Schicht fungiert als Engpass und liefert das letzte \hat{y}_n , das über eine Verwerfungsschicht verarbeitet wird. Die Verwerfungsschicht setzt die Eingabeelemente mit einer bestimmten Wahrscheinlichkeit auf Null, um eine Überanpassung zu verhindern. $dp=0.5=50\%$ definiert die Anzahl der verworfenen Elemente. Eine vollständig verbundene Schicht verbindet jedes ihrer Neuronen mit allen Aktivierungen in der vorherigen Schicht, um eine übergeordnete Argumentation (engl. high-level reasoning) als Ausgabe (Dimension: 1×1) für die Regressionsschicht bereitzustellen. Schließlich berechnet die Regression-Schicht den Verlust des halben mittleren quadratischen Fehlers (engl. half-MSE) für das Regressionsproblem und gibt die Geschwindigkeit $v' = Y_i$ zurück. Die Parameterkonfiguration, die die besten Ergebnisse liefert, wird in Abschnitt 5.6 diskutiert.

Entwurfsentscheidungen

Die ResNet18-BLSTM-Architektur wird so entworfen, dass sie zuerst hochdimensionale Merkmale f_s aus Faltungsschichten extrahiert und generiert und dann die Quintessenz dieser über Zeit und Kontext f_t verfolgt, d. h. sie merkt sich die Entwicklung von Merkmalskarten, die die Abbildung von Bewegung auf Geschwindigkeit darstellen, in den BLSTM-Schichten. Die Filterkerne F des 1D-ResNet-18 extrahieren

5 Geschwindigkeitsschätzung

f_s , während sie über eine Reihe von Fenstern gleiten, um Merkmale an verschiedenen Positionen zu erkennen [221]. Die Idee hinter der 1D-Faltung besteht darin, eine Element-für-Element-Multiplikation für jeden 1D-Filterkern mit jedem 1D-Fenster durchzuführen, um eine initiale Merkmalskarte zu erstellen, die zeitempfindlich ist [264].

Sobald der erste Teil des Netzes hochdimensionale Merkmale \hat{f}_s aus dem Fensterbündel extrahiert hat, werden sie in den BLSTMs übergeben. Die hinzugefügte Gruppierungsschicht reduziert die Komplexität des Netzes und konzentriert sich auf die vielversprechendsten \hat{f}_s . Abbildung 5.10 zeigt den BLSTM-Teil der Hybridarchitektur im Detail. Hier fließen Daten von unten nach oben. Das BLSTM verfolgt die Entstehung von Merkmalen f_s im Zeitverlauf. Engpassmerkmale f_t werden unter Verwendung einer Viele-zu-Eins-BLSTM-Architektur extrahiert, d. h. $X_i > Y_i = 1$. Im Gegensatz zu einfachen LSTMs verarbeitet das BLSTM Informationen über die Zukunft und die Vergangenheit, indem es den kurzfristigen und langfristigen Kontext zwischen Ein- und Ausgabe sowohl in Vorwärts- als auch in Rückwärtsrichtung ableitet [221]. Das BLSTM fungiert als zeit- und kontextsensitiver Detektor und Verfolger von hochdimensionalen Bewegungsmerkmalen, extrahiert durch die Faltungsschichten, in der Signaleinbettung acc und gyr . Es werden zwei verborgene LSTM-Schichten (H_f und H_b) für jede BLSTM verwendet, um Vorwärts- und Rückwärtsrichtungen bereitzustellen. Gemäß früheren Arbeiten [264] wird die letzte verborgene Schicht als Engpassschicht (engl. bottleneck layer) verwendet.

Aus einem einzelnen Fenster m lernen das Modell, wie es eine einzelne Geschwindigkeit v'_i schätzt und aufeinanderfolgende voraussagt. Wenn der Anfangszustand H_0 zufällig initialisiert wird, hat das Modell keine Vorkenntnisse über die Bewegung und lernt somit, die Geschwindigkeit aus lokalen Informationen zu schätzen. Während der Trainingsphase lernt das vorgeschlagene Modell, latente Bewegungseigenschaften aus einer aufeinanderfolgenden Folge von Fenstern zu nutzen. Zur Laufzeit wird die aktuelle Geschwindigkeit v'_i eines zu lokalisierenden Objektes in diesem Fenster vorhergesagt. Die letzte Geschwindigkeits schätzung \hat{y}_n wird als Endergebnis v'_n pro Fenster ausgewählt.

5.5.4 Nachbearbeitung

Die Nachbearbeitung verwendet ein LKF, um die Geschwindigkeit v' zu optimieren, die die Hauptverarbeitung bereitstellt. Ein LKF ist ein einfaches Bayessches Filter [222, 203], von dem bekannt ist, dass es Rauschartefakte in zeitlichen Informationen wie der Geschwindigkeit reduziert, sogenannte Glättung. Es besteht aus linearen Zustandsübergangsmodellen und Messmodellen und nimmt an, dass alle Rauschkomponenten ein additives Gaußsches Rauschen c sind, das nicht Null ist. Das Zustandsübergangsmodell beschreibt die Dynamik eines sich bewegenden Objekts, z. B. des IMU-Sensors. Das Messmodell beschreibt die Beziehung zwischen dem verfolgten Zustand, der optimierten Geschwindigkeit v'' , und den erwarteten Messungen, der Eingangsgeschwindigkeit v' . Die Kovarianz des Zustandsübergangsrauschen $c_{v,d}$ setzt die Zuverlässigkeit des Zustands voraus. Basierend auf dieser Verlässlichkeit liefern die Vorhersagen und Messungen Informationen, die das Filter gewichtet. Das Filter verfolgt die Geschwindigkeit v'' des Objekts als Zustand.

Das hier vorgeschlagene Zustandsübergangsmodell geht von einer konstanten Geschwindigkeit zwischen aufeinanderfolgenden Zeitschritten i und $i + 1$ aus:

$$v''_{i+1} = v''_i + c_{v,d}, \quad (5.10)$$

mit der Kovarianz des Zustandsübergangsrauschen $c_{v,d} \sim \mathcal{N}(0, T^2 \sigma_{v,d}^2)$ und dem Zeitintervall der Abtastung $T (= t(i+1) - t(i))$. Da das Filter nur die Geschwindigkeit v'' als 1D-Zustand verarbeitet, kann der Zustandsübergang mit einem Skalar definiert werden und die Zustandsübergangskovarianz auf $T^2 \sigma_{v,d}^2$ gesetzt werden. $\sigma_{v,d}$ wird geschätzt, indem initial die Standardabweichung der Ableitung mit Bezug zur Zeit der Geschwindigkeiten auf den Validierungsdaten berechnet wird. Zur Erinnerung: das DL Modell wird auf dem Trainings- und Validierungsdatensatz trainiert und das LKF wird auf dem Validierungsdatensatz trainiert. In der Regel nimmt die Gültigkeitsannahme des Modells mit der Zeit ab. Ähnlich wie beim Zustandsübergangsmodell geht das Messmodell auch von additivem weißem Gaußschen Rauschen aus $c_{v,m} \sim \mathcal{N}(0, \sigma_{v,m}^2)$:

$$v''_i = v'_i + c_{v,m}. \quad (5.11)$$

Wieder wird das Messmodell mit einem Skalar definiert, aber diesmal die Messkovarianz auf $\sigma_{v,m}^2$ gesetzt. Weiter wird die Standardab-

5 Geschwindigkeitsschätzung

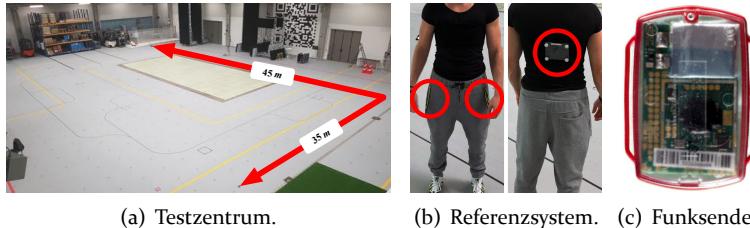


Abbildung 5.11: Hardwarekonfiguration. Links: Seitenansicht der Datenerfassungsumgebung; Mitte: Mobiltelefone in den Taschen eines Probanden und verfolgbares Objekt des Referenzsystems; Rechts: RedFIR-Funksender.

weichungen von v' im Validierungsdatensatz berechnet, um $\sigma_{v,m}$ für jede (Hauptverarbeitungs-)Methode einzeln zu bestimmen, und dann wird die Standardabweichung der Geschwindigkeitsableitung der Validierungsdaten berechnet, um $\sigma_{v,d}$ zu erhalten. Diese Parameter werden direkt aus den Daten abgeleitet und im LKF [222] verwendet, um v'' auf v' in jedem Zeitschritt i vorherzusagen.

5.6 Experiment

Abschnitt 5.6.1 beschreibt zunächst die Hardware- und Softwarekomponenten des Versuchsaufbaus. Anschließend wird der Datenerfassungsprozess beschrieben (Abschnitt 5.6.2) und die verschiedenen Varianten des Datensatzes diskutiert (Abschnitt 5.6.3). Abschnitt 5.6.4 zeigt die Parametrisierung der klassischen, der ML- und der DL-Methoden. Zu beachten ist, dass der Aufbau der Experimente zusammen mit ihren Ergebnissen in Abschnitt 5.7 beschrieben wird.

5.6.1 Hard- und Softwarekomponenten

Messbereich

Die Bewegungsdaten, d. h. acc und gyr , werden am Fraunhofer IIS L.I.N.K. Testzentrum (Lokalisierung, Identifikation, Navigation, Kommunikation) in Nürnberg aufgezeichnet. Das Testzentrum bietet ein einzigartiges Testgelände auf $1.400m^2$, siehe Abbildung 5.11(a) [31]. Die folgenden Mess- und Referenzsysteme wurden verwendet, um die

Trainings-, Validierungs- und Testdaten zu sammeln und zu kennzeichnen und um die Eigenschaften verschiedener Bewegungstypen, d. h. Gehen, Joggen, Laufen und Zufall, zu erfassen.

Referenzsystem

Die Referenzdaten (3D-Positionen) wurden mit 28 Kameras des millimetergenauen optischen Qualisys-Bewegungsverfolgungssystems aufgezeichnet (sphärischer Fehlerwahrscheinlichkeit, $SEP_{95} \leq 5 \text{ mm}$ und $\leq 0.1^\circ$). Die Kameras sind an den Rändern der Oberseite des Testbereichs angebracht und decken ein Volumen von 11.025 m^3 ($45 \times 35 \times 7 \text{ m}$) ab. Die Probanden tragen 4 kleine verfolgbare reflektierende Marker, die an einem elastischen Band befestigt sind (siehe Abbildung 5.11(b)). Alle Kameras hatten permanent ein klares Sichtfeld, als die Position und Ausrichtung jedes Probanden mit konstanten 100 Hz aufgenommen wurden.

System zur Messung der Funklokalisierung

Die Funkdaten werden mit RedFIR [158] aufgezeichnet, ein Breitband-Echtzeit-Lokalisierungssystem (engl. Real-Time Locating System, RTLS), das im 2.4 GHz-Band mit $B_w=80 \text{ MHz}$ arbeitet. 16 Empfängereinheiten, 4 stationäre Referenzsender und bis zu 140 mobile Sendeeinheiten bieten ein Uplink-Ankunftszeitdifferenz (engl. Uplink-Time-Difference-of-Arrival, UTDoA) mit Positionen bei 20 Hz und einer Genauigkeit von $CEP_{95} < \pm 0.2004 \text{ m}$ (kreisförmige Fehlerwahrscheinlichkeit). Abbildung 5.11(c) zeigt eine mobile RedFIR-Sendeeinheit.

IMU-Messgerät

Es werden zwei Samsung Galaxy S7-Telefone mit ihren Sensoren *Acc* und *Gyr* verwendet (STMicroelectronics LSM6DS3 tastet *acc* bei $\pm 16g$ und *gyr* bei $\pm 1000 \text{ dps}$ bei einer quasi-konstanten Abtastrate von 100 Hz ab), um die Beschleunigungen und Winkelraten des Sensors, respektive Probanden, zu messen. Um etwaige Bewegungsunterschiede und -anomalien zwischen dem linken und rechten Bein zu decken, wird ein Telefon, in der Ausrichtung: inverses Porträt, engl. inverse portrait, lose jeweils in die linke und rechte Hosentasche gesteckt, siehe Abbildung 5.11(b). Zu beachten ist, dass *zwei* Telefone verwendet werden, um

5 Geschwindigkeitsschätzung

mehr Daten pro Proband parallel zu erfassen. Jedoch werden Geschwindigkeiten für Daten nur von *einem* der Sensoren zum Trainieren und zum Prognostizieren verwendet. Da alle Methoden auf dem SMV des Sensorsignals ausgewertet werden, sind die Eingangsdaten rotationsinvariant. Daher ist die ursprüngliche Platzierung und Ausrichtung des Telefons, respektive des Sensors, irrelevant.

IMU-Software

Über die C++ Schnittstelle des NDKs der Android-API (Version 6, 2019) wird direkt auf die IMU-Sensordaten der Mobiltelefone zugegriffen [489]. Zusammen mit Zeitstempeln, die global NTP-zeitsynchronisiert sind, werden diese gespeichert. Die Programmierschnittstelle (engl. Application Programming Interface, API) empfängt auch die NTP-zeitsynchronisierten 3DoF- und 6DoF-Statusinformationsdaten und Signaldatenströme von RedFIR und Qualisys über 5 GHz Wi-Fi. Jede Aktivität und Messung (*acc*, *gyr* und *ref*) wird beschriftet und separate Dateien pro Studiendurchlauf und Teilnehmer werden gespeichert. Somit können die Daten zur Positionierung (siehe Abschnitt 4), Geschwindigkeitsschätzung (siehe Abschnitt 5), Orientierungsschätzung (siehe Abschnitt 6) und Posenschätzung (siehe Abschnitt 7) verwendet werden.

Zeitsynchronisation

Alle Systeme und Geräte sind mit demselben globalen NTP-Zeitserver verbunden. Die Referenz- und Messsysteme (Qualysis, RT-LS und IMU) speicherten somit ihre Aufzeichnungen zusammen mit global NTP-zeitsynchronisierten Zeitstempeln. Zu beachten ist, dass die Referenzdaten von 100 Hz mit bis zu 1.000 Hz abgetastet wurden, um mit ihnen die Funkmessungen, variable Abtastrate von durchschnittlich 20 Hz, und die Trägheitsmessungen, konstante Abtastrate von 100 Hz, zu synchronisieren.

Rechensystem

Die Daten wurden mit Mathworks MATLAB analysiert und die DL-Modelle mit dem PyTorch-Framework (Version 1.4) entworfen. Die ML- und DL-Modelle wurden auf einem Cluster verarbeitet und trainiert, auf dem 12 Rechenknoten mit 336 CPU-Kernen (zentrale Rechen- und Steuereinheit eines Computers, engl. Central Processing Unit, CPU),

4 TB RAM und 32 GPUs (NVIDIA V100 32 GB) mit einer Spitzenleistung von 249.6 TFLOPs ausgeführt werden bei doppelter Genauigkeit mit 502.4 TFLOPs und bei einfacher Genauigkeit mit 4 PFLOPs.

5.6.2 Datenerfassung

23 Personen (männlich: 17, weiblich: 6; Durchschnittsalter: 26.7; Körpergröße von 1.46 m bis 1.87 m, SD 0.19 m) führen vier verschiedene Arten von Bewegungsaktivitäten, Gehen, Joggen, Laufen und zufällige Bewegungen, eine natürliche Kombination von allen, innerhalb des Verfolgungsbereichs der Referenzsysteme aus, eine Aktivität nach der anderen. Die Teilnehmer stehen zu Beginn jeder Aktivität an derselben Ausgangsposition für 1 min still. Dies hilft, um die anfänglichen Sensorfehler zu schätzen und um in der Datenaufbereitung die Möglichkeit zu haben, Sensoren zu kalibrieren und die Ergebnisse zu überprüfen und um die anfängliche Pose bei wiederholten Messungen zu kalibrieren und zu identifizieren. Nach Abschluss jeder Aktivität werden die Aufzeichnungen auf Vollständigkeit analysiert. Die Aktivitäten werden manuell beschriftet: Bewegungsklasse gemäß Aktivität. Jede Aktivität dauerte durchschnittlich 7.5 min (SD=±1 min).

Die Teilnehmer führten die Geh-, Jogging- und Rennen-Aktivitäten mit unterschiedlichen Geschwindigkeiten durch, meist jedoch entlang ähnlicher Bewegungsbahnen, siehe Abbildung 5.12. Jeder Teilnehmer führte jedoch die zufälligen Bewegungen auf seine/ihre individuelle und einzigartige Weise aus, d. h. er/sie wechselte zwischen Gehen, Joggen und Rennen mit individuellen Bewegungsgeschwindigkeiten und Trajektorien. Die Teilnehmer wurden gebeten, die gesamte Studie und die Trainingseinheiten ernst zu nehmen. Die Probanden trugen verschiedene Arten von Sporthosen mit weiten oder engen Hosentaschen, um verschiedenes Sensorrauschen und Bewegungsartefakte in den Daten zu erhalten. Beschädigte Datensätze, bspw. durch Signalverlust, wurden sowohl visuell als auch statistisch identifiziert. Kurze Aussetzer (<20 ms) wurden entweder durch erneutes Abtasten und Interpolieren rekonstruiert oder die Aufnahme neu gestartet. Die Synchronisation der Zeitstempel hatte einen Fehler von <1 ns/h aufgrund von Taktverschiebungen zwischen der Referenz und den Messsensoren.

5 Geschwindigkeitsschätzung

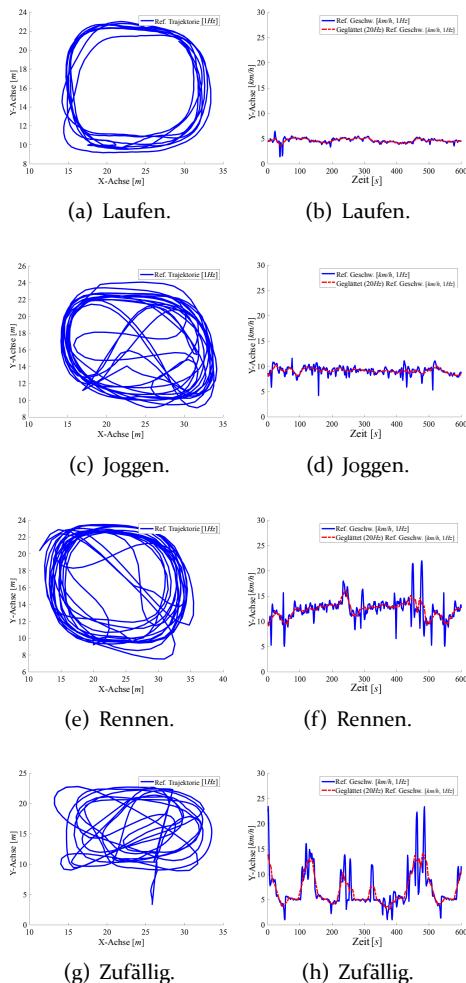


Abbildung 5.12: Beispielhaftes Segmentbündel eines zufällig ausgewählten Probanden. Jeder Datenpunkt, sowohl für Trajektorien als auch für Geschwindigkeiten repräsentiert ein einzelnes Fenster. Die linke Spalte zeigt Referenztrajektorien und die rechte Spalte zeigt Referenzgeschwindigkeiten (bei 1 Hz).

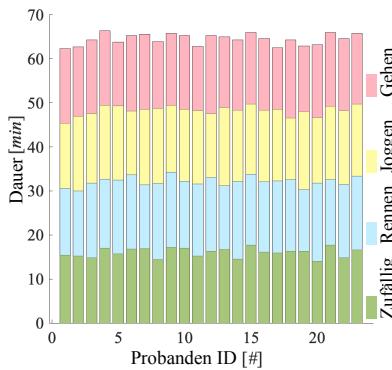


Abbildung 5.13: Statistik des zeitabhängigen Datensatzes mit aufeinanderfolgenden Fenstern mit einer *quasi-identischen* Verteilung.

5.6.3 Datensätze

Insgesamt wurden ungefähr $23h=1.381$ min Bewegungsdaten aufgezeichnet ($=4$ Aktivitäten $\cdot 2$ Sensoren $\cdot 23$ Probanden $\cdot 7$ min, SD 1 min) mit ca. 30 min pro Sensor und Aktivität (SD ± 4.6 min). Abbildung 5.13 zeigt die Verteilung der Aktivitäten und ihre Dauer pro Benutzer, jedoch mit beiden Sensoren (ca. 60 min= $2\cdot 30$ min). Die Studie lieferte Mess- und Referenzdaten bei einer Gesamtstrecke von rund 107 km für alle Aktivitäten (Gehen: ~ 14 km; Joggen: ~ 29 km; Laufen: ~ 38 km; und zufällig: ~ 26 km) bei unterschiedlichen Geschwindigkeiten³. Da es zwei Sensoren pro Benutzer gibt, legten die Sensoren eine Gesamtstrecke von ungefähr 214 km zurück.

In der Studie wird festgestellt, dass Schiebefenstern der Größe 128 Hz und einer Überlappung von 50% ($N_w/2=64$) genügend herausragende Merkmale \hat{f}_s und \hat{f}_t bei geringen Rechenkosten enthalten, um die höchste Genauigkeit der Geschwindigkeitsschätzung zu erreichen. Bereits 1.28 s decken langfristige Beziehungen menschlicher Bewegung ab und machen die hybride DL-Methode anwendbar. Zu beachten ist,

³Geschwindigkeiten: Durchschnitt für Laufen: $1.4 \frac{m}{s}$; min. $0.8 \frac{m}{s}$; max. $2.2 \frac{m}{s}$; SD $0.31 \frac{m}{s}$; Durchschnitt für Joggen: $2.8 \frac{m}{s}$; min. $1.5 \frac{m}{s}$; max. $3.4 \frac{m}{s}$; SD $0.34 \frac{m}{s}$; Durchschnitt für Rennen: $3.6 \frac{m}{s}$; min. $2.9 \frac{m}{s}$; max. $7.9 \frac{m}{s}$; SD $0.67 \frac{m}{s}$; Durchschnitt für Zufällig: $2.5 \frac{m}{s}$; min. $1.2 \frac{m}{s}$; max. $7.8 \frac{m}{s}$; SD $3.19 \frac{m}{s}$.

5 Geschwindigkeitsschätzung

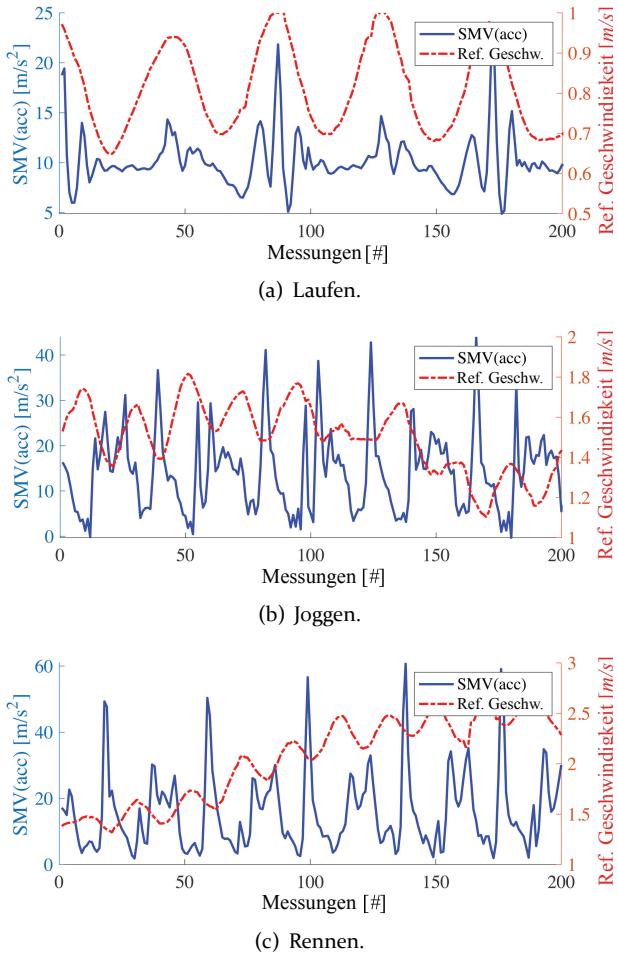


Abbildung 5.14: Rohe acc-Signale: SMV (blau), Referenzgeschwindigkeit (rot).

dass diese Parameter die besten Ergebnisse für die Varianten V_1 , V_2 und V_3 des Datensatzes liefern. Daher beziehen sich die Beschreibung des Datensatzes, die Parametrisierung der Methoden und die Ergebnisse in Abschnitt 5.7 auf diese optimale Einstellung der Parameter.

Insgesamt wurde ein Datenpool gesammelt, der aus 92 Segmentbündeln (23 Themen \cdot 4 Aktivitäten) mit 64.688 aufeinanderfolgenden Fenstern (ohne Überlappung, $N_w=128$, $f_s=100$ Hz ungefähr 350 Fenster pro Benutzer) besteht, pro Aktivität und pro Sensor oder 129.375 überlappende Fenster (50% Überlappung, $N_w/2=128/2=64$) mit entsprechenden Referenzpositionen und Geschwindigkeiten.

Abbildung 5.12 zeigt eine beispielhafte Teilmenge von 10 min Bewegung für einen einzelnen Benutzer. Die Grafiken zeigen Trajektorien (linke Spalte) und Geschwindigkeiten (rechte Spalte) aller Aktivitäten (von oben nach unten). Jeder Datenpunkt repräsentiert ein einzelnes Fenster ohne Überlappung. Abbildung 5.14 zeigt die Beschleunigung (blau) und Geschwindigkeit (rot) beim Gehen, Joggen und Rennen für 200 Messungen. Die Signalkomplexität (Frequenz und Rauschen) nimmt mit höheren Geschwindigkeiten zu, Spitzen werden weniger ausgeprägt, und somit werden Spitzendetektoren ineffizient, siehe Abschnitt 5.7.2. Aus dem Datenpool wurden drei Varianten erstellt.

Datensatz Variante V_1

V_1 bietet Segment-, Fenster- und Merkmalsbündel ($276=3 \cdot 92$), die die gesamten Daten für jedes Thema und jede Aktivität abdecken. Die Bündelgrößen variieren zwischen Benutzern und Aktivitäten, da bspw. einige Probanden mit Geschwindigkeiten joggen, die den Renngeschwindigkeiten anderer entsprechen. Mit Daten von beiden Sensoren enthält V_1 129.375 überlappende ($50\% = N_w/2 = 64$) Fenster oder Merkmale bei $N_w=128$ und $f_s=100$ Hz (ungefähr 700 Fenster pro Benutzer, pro Aktivität und pro Sensor mit Überlappung).

Datensatz Variante V_2

V_2 bietet Segment-, Fenster- und Merkmalsbündel ($276=3 \cdot 92$), die auf die kürzeste Bündelgröße aller Benutzer zugeschnitten sind. Dieser Datensatz enthält daher die gleiche Anzahl von individuellen Bewegungen pro Benutzer und Aktivität. Somit sind die Aktivitäten gleichmäßig verteilt. Aus dem gleichen Grund wie zuvor sind die Geschwindigkei-

5 Geschwindigkeitsschätzung

ten jedoch wieder nicht identisch verteilt. Insgesamt enthält V_2 112.643 überlappende Fenster oder Merkmale ($50\% = N_w / 2 = 64$) bei $N_w = 128$ und $f_s = 100$ Hz (ungefähr 600 Fenster pro Benutzer pro Benutzer) Aktivität und pro Sensor mit Überlappung).

Datensatz Variante V_3

V_3 bietet nur Fenster- und Merkmalsbündel (184=2·92). Alle Fenster aller Probanden und alle ihre Aktivitäten von V_1 werden gemäß ihrer korrespondierenden Referenzgeschwindigkeit v_{ref} sortiert. Dann werden die Fenster mit einer Intervalllänge von 2 km/h in einzelne Gruppen unterteilt, das zu Gruppen mit folgenden Intervallen führt [3.5), [5.7), ..., [11.13] km/h. Für jede dieser Gruppen werden insgesamt 156 Einträge (pro Sensor) für alle Probanden und Aktivitäten gefunden. Gruppen mit weniger als 156 Einträgen werden gelöscht, und Gruppen mit mehr als 156 Einträgen werden auf 156 Einträge beschränkt. Überschüssige Einträge werden zufällig gelöscht. Dies steht im Einklang mit typischen Vorverarbeitungstechniken in der statistischen Analyse und im Tiefenlernen zur Analyse gleichmäßig verteilter Klassen und Kategorien in Daten und ihrer Auswirkungen auf die Verarbeitungsmethoden [64, 468]. Daher enthält V_3 homogene Geschwindigkeitsgruppen für alle Probanden, unabhängig von ihrer Aktivität. Zu beachten ist, dass innerhalb einer Gruppe die Gangart zwischen den Teilnehmern variiert, d. h. es ist unterschiedlich, wie eine Bewegung zu derselben Geschwindigkeit und zu den resultierenden Merkmalen \hat{f}_t führt, die das datengetriebene Modell lernt, bspw. laufen zwei Probanden mit derselben Geschwindigkeit, aber einer hat einen Gangfehler. V_3 enthält insgesamt 35.880 Fenster oder Merkmale ohne Überlappung (5 Geschwindigkeitsgruppen · 156 Bewegungs-/Geschwindigkeitspaare pro Proband · 23 Probanden · 2 Sensoren ohne Überlappung).

Datensatz Limitierungen

Aus Sicht des maschinellen und des tiefen Lernens würde ein *perfekter* Datensatz identisch verteilte Daten, mit exakt äquivalenten statistischen Eigenschaften jeder einzelnen individuellen Kategorie, einbetten, sodass jeder Proband in der gleichen möglichen Variation von Beschleunigungen und Geschwindigkeiten läuft, joggt und rennt mit gleicher Dauer auf der gleichen Trajektorie und mit jeder möglichen Gangart und Geschwindigkeit. Ein solcher Datensatz würde auch alle

möglichen Gang- und Bewegungsverhalten abdecken. Dieser Datensatz würde erfordern, dass Probanden dieselben Bewegungsmuster bereitstellen, die über die gesamte Zeitspanne einer Aktivität zu denselben Geschwindigkeiten führen. Da dies zum Zeitpunkt der Forschungsarbeiten im Rahmen der Dissertation unmöglich war, lieferten die Probanden diese deterministischen Geschwindigkeiten nicht. Z. B. joggten einige mit einer Geschwindigkeit, die andere nur erreichten, wenn sie rannten. Daher mischen sich sowohl die Bewegungsmuster als auch die Geschwindigkeiten innerhalb und zwischen den Aktivitäten. Obwohl dies durch die Bildung von Gruppen mit nahezu gleichmäßig verteilten Geschwindigkeiten näherungsweise kompensiert wird, gibt es noch Raum für Verbesserungen.

5.6.4 Parametrisierung der Geschwindigkeitsschätzer

Die Vergleichsstudie der Dissertation bewertet fünf verschiedene Geschwindigkeitsschätzer, drei aus dem Stand der Technik und die ressourcenschonende CNN-BLSTM- und die genaue Hybridarchitektur. Zu beachten ist, dass im weiteren Verlauf des Kapitels die unterstrichenen Abkürzungen der Methoden verwendet werden, um auf sie zu verweisen, wenn die Ergebnisse in Abschnitt 5.7 diskutiert werden.

Basisverfahren I, klassisch PDR.

Um eine PDR-Methode zu implementieren, die den Stand der Technik darstellt, wird die Methode auf einem etablierten biomechanischen Modell aufgebaut und für unbekannte Benutzer und Bewegungen verallgemeinert. Bei Trägheitsmessungen erkennt diese Schritt-basierte PDR Spitzen in den Signalen der Segmentbündel, betrachtet diese Spitzen als bekannte Gangphase eines Schrittes und schätzt die Schrittlänge von dort aus. Die Schrittlängenschätzung basiert auf dem bekannten Modell von Tian et al. [190]. Ihr Modell basiert auf einem groben geometrischen Verständnis, wie sich ein Sensor bewegt, wenn eine Person einen Schritt macht. Ihr Schrittlängenschätzungsmodell:

$$L = K \cdot h \cdot \sqrt{fq} \quad (5.12)$$

wurde nach bestem Gewissen implementiert, wobei h die Größe, m.a.W. Körperhöhe mit Schuhen, der Person ist, K ein Kalibrierungsparameter ist und fq die Schrittfrequenz ist. Bestehende

5 Geschwindigkeitsschätzung

PDR-Implementierungen unterscheiden sich im Allgemeinen darin, wie sie K und f_q bestimmen.

Die Berechnung von f_q basiert normalerweise auf den Ideen von Weinberg et al. [428, 467], die Spitzen auf den minimalen und maximalen vertikalen Beschleunigungswerten auf der Körperachse erfassen. In der Literatur auch als Gier-, Azimut-(engl. azimuth-) oder Kurs(engl. heading)-Achse bekannt. Anstatt nur einen der vorhandenen Schrittdetektoren zu verwenden, wurde im Rahmen der hier vorgeschlagenen Methode ein Schrittdetektor entwickelt, der am besten zu Sport- und VR-Anwendungen passt: verschiedene Benutzer, verschiedene Arten von Bewegungen und vor allem Sensoren in Taschen und am Kopf. Der resultierende Detektor erzielt bessere Ergebnisse als vorhandene. Um den idealen Schrittdetektor zu finden, wurden verschiedene Varianten untersucht, die entweder mit den Signalen acc oder gyr oder beiden arbeiten. Die besten Ergebnisse wurden mit dem SMV von acc erzielt, da dieser Detektor eine Rotationsinvarianz ermöglicht. Eine wichtige Erkenntnis war (a) die Verwendung eines Butterworth-Tiefpassfilters mit einer Grenzfrequenz (engl. cut-off frequency) von 15 Hz, um hochfrequentes Rauschen vor dem Erkennen der Spitzen zu entfernen, und (b) das Ignorieren von Spitzen im Signal unterhalb eines Schwellenwerts von 17.23 m/s^2 während des Laufens, 24.17 m/s^2 während des Joggens und 32.61 m/s^2 während des Rennens.

Auch die Koeffizienten K sind individuell für die Bewegung und den Probanden und müssen daher ebenfalls gesondert betrachtet werden. Da die genaue Beziehung zwischen Schrittlänge und Schrittzeiten für eine bestimmte Person s und eine bestimmte Aktivität a spezifisch ist, wie z. B. Gehen und Rennen, da sich jeder anders bewegt, verwenden Weinberg et al. [467] ein probanden- und aktivitätsspezifisches ($K=$) K_{sa} , das sie in einer vorhergehenden Kalibrierungsphase messen. Um die Obergrenze der Leistungsfähigkeit, bspw. die Genauigkeit der Geschwindigkeitsschätzung, dessen zu bestimmen, was eine subjektaktivitätsspezifische PDR erreichen kann, wurde auch K_{sa} unter Verwendung von Referenzdaten aus dem submillimetergenauen Referenzsystem genau kalibriert, das auch die Referenzhöhe h für jeden Teilnehmer bereitstellte. Die ersten Schritte einer Person (ungefähr 1 min) wurden

individuell für jede Aktivität betrachtet, um das Verhältnis zwischen der geschätzten Entfernung:

$$d' = v' \cdot dt, \quad (5.13)$$

mit der Zeitspanne von zwei Schritten dt und des Referenzabstands d_{ref} zwischen zwei Schritten zu bestimmen:

$$K_{sa} = d'/d_{ref}. \quad (5.14)$$

In einem realen Anwendungsfall gibt es jedoch kein genaues Referenzsystem und es ist unmöglich, K_{sa} -Werte pro Proband und pro Aktivität zu kalibrieren. Stattdessen führen unbekannte Probanden unbekannte Aktivitäten aus. Da dies die Genauigkeit der Geschwindigkeitsschätzung von PDR Methoden einschränkt, wird die Schrittängenschätzung verallgemeinert, indem ein einzelner allgemeiner Kalibrierungsparameter K_g abgeleitet wird, der die Bewegungseigenschaften mehrerer Benutzer und Aktivitäten einbettet. Dazu wurden alle $80K_{sa}$ -Parameter (=20 Probanden · 4 Aktivitäten) gemittelt:

$$K_g = \frac{1}{n} \sum_1^n K_{sa(i)} = 1.2771. \quad (5.15)$$

3 Probanden wurden aus dem Ergebnis von Gleichung 5.15 weggelassen, um die Generalisierbarkeit der Methoden zu testen. Die Verwendung eines festen K_g ohne vorherige Kalibrierung anstelle fein abgestimmter K_{sa} -Werte ermöglicht PDR für reale Anwendungsfälle einzusetzen und ermöglicht auch einen fairen Vergleich mit den anderen vier Schätzern, die ebenfalls nur mit unbekannten Testpersonen und unbekannten Aktivitäten evaluiert werden, m.a.W. die ebenfalls nicht subjektaktivitätsspezifisch optimiert werden.

Der Nachteil der klassischen PDR Verfahrens des Stands der Technik ist der Genauigkeitsverlust. In den Experimenten werden alle feinkörnigen K_{sa} -Koeffizienten berechnet. Während K_{sa} -Koeffizienten verwendet werden, erhöhte sich die Genauigkeit der PDR-Geschwindigkeitsschätzung um 19%, was jedoch die Gesamtergebnisse des Vergleichs mit den anderen vier Schätzern in Abschnitt 5.7 überhaupt nicht ändert. Die klassischen PDR-Verfahren liefern in allen Experimenten die schlechtesten Ergebnisse. Daher ist es sowohl sicherer (ein K_{sa} eines Menschen mit sehr großem h liefert völlig falsche Werte

5 Geschwindigkeitsschätzung

als ein K_{sa} eines Menschen mit sehr kleinem h) und alltagstauglich als auch fair in der Auswertung und im Vergleich mit anderen Verfahren mit K_g Zahlen zu arbeiten.

Basisverfahren II, Maschinelles Lernen mit Gaußschen Prozessen ML-GP

Das zweite Basisverfahren der Vergleichsstudie ist ein ML-basierter Schätzer, der ebenfalls bestrebt ist, den Stand der Technik am besten widerzuspiegeln. Um die beste ML-Methode für das vorliegende überwachte Geschwindigkeitsregressionsproblem zu finden, wurde eine umfangreiche Studie durchgeführt, die eine Reihe bekannter ML-basierter Methoden sowie eine große Anzahl handgefertigter statistischer und Frequenzbereichsmerkmale umfasste, von denen bekannt ist, dass sie die besten Ergebnisse für Zeitreihendaten erzielen in ähnlichen Anwendungsgebieten [45]. Die in einem Raster optimierte Studie durchsuchte die Parameter der Methoden und bewertete verschiedene Kombinationen von Merkmalen, um die eine Methode und den einen Merkmalssatz zu finden, die die höchste Genauigkeit und Robustheit des Geschwindigkeitsschätzers ergeben.

Zunächst werden die in der Studie verwendeten Merkmale erörtert [231]. Die 6 Zeitbereichsmerkmale sind von Bishop et al. [45] inspiriert und setzen sich aus absoluter Energie, d. h. der absoluten Energie der Zeitreihe, bzw. die Summe der Quadratwerte, Maximum, Minimum, Mittelwert, Median und Varianz. Die 7 Frequenzbereichsmerkmale sind von Vathsangam et al. [444] inspiriert und setzen sich aus der spektralen Bandbreite, der spektralen Ebenheit , d. h. das Verhältnis des geometrischen Mittels zum arithmetischen Mittel des Betragsspektrum des Signals, dem spektralen Rollout , d. h. Frequenz, unter der ein bestimmter Prozentsatz der gesamten spektralen Energie liegt, dem spektralen Schwerpunkt , d. h. der spektrale Schwerpunkt (Mittelwert), der Varianz, der Versatz und der Kurtosis des absoluten Fourier-Transformationsspektrums zusammen.

Während alle Zeitbereichsmerkmale einfach zu berechnen sind, muss für die Frequenzbereichsmerkmale eine Kurzzeit Fourier-Transformation (engl. Short Term Fourier-Transformation, STFT) auf jedes Fenster angewendet werden und die Merkmale aus der STFT-ähnlichen Darstellung der Zeit-Frequenz-Verteilung extrahiert

werden. Die Rastersuche untersuchte auch eine Dimensionsreduktion (Hauptkomponentenanalyse, engl. Principal Component Analysis, PCA [524]) aller 13 Merkmale, um sie auf Merkmalsbündel zu reduzieren, die am meisten zur Genauigkeit der Geschwindigkeitsschätzung beitragen. In den Datensätzen V₁, V₂ und V₃ ergab dies 84 Merkmalskombinationen für jedes Fenster (42 Merkmalsbündel = 6 statistische · 7-Frequenzbereichsmerkmale sowie die beiden größten Hauptkomponenten jedes dieser Merkmalsbündel, inspiriert von Bishop et al. [45]).

Bei der Rastersuche wurden auch verschiedene Regressionsmethoden (Reg.) für alle möglichen Kombinationen der oben genannten Merkmale berücksichtigt. Dazu wurden die von Bishop et al. [45] vorgeschlagenen Methoden nach dem Stand der Technik ausgewählt: lineare Reg., logistische Reg., Klassifizierungs- und Regressionsbaum (CART) und Stützvektor-Reg. (SVR). Inspiriert von der Arbeit von Rasmussen et al. [354] (der die Robustheit von ML-Methoden bewertete) wurden auch GP [444] untersucht. Mithilfe einer Rastersuche nach diesen 5 Regressionsmethoden und den 84 Merkmalskombinationen wurden die optimalen Hyperparameter abgeleitet.⁴

Unter den Schätzern liefert GP die höchste Genauigkeit mit einem Matern52-Kernel, jedoch mit etwas höheren Rechenkosten für das Training und die Laufzeit. Interessanterweise erreicht Matern52 die höchste Genauigkeit nur mit den beiden Hauptkomponenten, die die stärksten Eigenschaften aller 42 Merkmale darstellen. Um mit der hohen Dimensionalität der Daten fertig zu werden, wurde ein spärlicher (engl. sparse) GP verwendet. Für den Vergleich in Abschnitt 5.7 wurde diese beste Konfiguration von GP (fette Zahlen) als ML-Methode der Wahl verwendet. Die Ergebnisse der Rastersuche sagen aus, dass ML-GP die

⁴ML-GP Rastersuchparameter: Logistische Reg.: Strafe ∈ [L₁, L₂, elastisches Netz], Toleranz ∈ [0.0001, **0.001**, :1.0], C ∈ [0.1, **1.0**, 10], Löser ∈ [saga, newton-cg]; SVR Linear: C ∈ [1, **10**, **100**, 1000]; SVR Polynom: C ∈ [1, **10**, **100**, 1000], degree=3 ∈ [2::6]; SVR Radialbasisfunktion: γ ∈ [**0.001**, 0.0001], C ∈ [1, **10**, **100**, 1000]; DT: max. Tiefe=97 ∈ [:1::50], max. Merkmale=27 ∈ [:1::30], max. Blattknoten=15 ∈ [:1::20]; GP: Kernel ∈ [RBF, **Matern52**, Rotationsquadrat (RQ)], RQ α ∈ [0.001, 0.01, 0.1, **1.0**, **2.0**, :10], RQ Längenskala ∈ [0.1, **1.0**, :10], RBF Längenskala ∈ [0.1, 1.0, **3.0**, :10], M52-(Matern52-GP-Kernel-)Längenskala ∈ [0.1, **1.0**, :10], M52 ν ∈ [0.1, 1.0, 1.5, **2.0**, 2.5, :10]; Für jeweils 84 Merkmalskombinationen und Kombinationen von f_s (50, **100**, 200, 400) und N_w (64, **128**, 256, 512). Die Parameter stimmen mit den Ergebnissen von Rasmussen et al. [354] überein. Die Parameter, die die höchste Genauigkeit erzielen werden mit **fetten** Zahlen dargestellt.

5 Geschwindigkeitsschätzung

bestmögliche Konfiguration für alle renommierten ML-basierten Geschwindigkeitsschätzer bietet.

Tiefes Lernen I, RoNIN

Yan et al. [490] veröffentlichten RoNIN, den aktuellen Stand der Technik der Geschwindigkeitsschätzung für homogenes Gehen in öffentlich verfügbaren Datensätzen. RoNIN ändert und erweitert ResNet [176], siehe Abschnitt 5.5.3. Es ersetzt alle Resteinheiten vom Typ II durch Resteinheiten vom Typ I, ersetzt die Softmax- und Klassifizierungsschichten durch eine vollständig verbundene Schicht und eine (halbmittelquadratische) Regressionsschicht und passt die Eingabedimension an einen 6D-Eingabevektor mit einer Länge von 200 Werten an, der 1.200 Rohmessungen (= 6·200) für 3D-*acc* und -*gyr* mit einer Abtastrate von 100 Hz abdeckt.

Das ursprüngliche RoNIN Verfahren wurde mit seiner Standardkonfiguration auf die hier vorgeschlagenen Datensätze angewendet. Um den bestmöglichen DL-basierten Schätzer vom RoNIN-Typ zu erhalten, wurde eine Rastersuche zwischen möglichen Variationen und Konfigurationen, mit Eingabekombinationen von 1D- und 2D-SMVs (von *acc* und *gyr*), und unterschiedlichen Fenstergrößen der Eingabevektoren durchgeführt, mit Variationen von 64 bis 1024 Hz, und eine Kombination von Resteinheiten vom Typ I und Typ II. Mit einer Eingabefenstergröße von 256 Messungen an 2D-SMVs (von *acc* und *gyr*) und mit den weiteren idealen Konfigurationsparametern⁵ liefert die DL-basierte Methode vom Typ RoNIN für die hier vorgeschlagenen Datensätze die besten Ergebnisse der Rastersuche. Deshalb wird diese Konfiguration in Abschnitt 5.7 im Vergleich mit anderen Verfahren verwendet.

⁵RoNIN-Rastersuchparameter: Löser ∈[SGD, Adam, rmsprop]; $\beta_1, \beta_2=0.01$; Momen-tum=0.9; anfängliche Lernrate (*LR*)=0.05 ∈[1.0:0.1:0.00001]; *LR* Verwerfungsperiode ∈[0, 10, 50, 100] Epochen; *LR* Verwerfungsrate =0.9; Stapelgröße ∈[128, 256, 512, 1024, 2048]; ResNet $F_s=d \times N$ mit $d \in [1, 2]$ Dimensionen und $N=7 \in [3, 4, \dots, 9, 10]$; ResNet $S=1 \times N$ mit $N \in [1, 2, \dots, 9, 10]$; Mischen ∈[Nein, pro Epoch]; Gradienten-beschneidung = max(Eingabe); $dp \in [10, 20, 50, 75]\%$; Für jede Kombination von f_s (50, 100, 200, 400) und N_w (64, 128, 256, 512). Die Parameter stimmen mit den Ergebnissen von Startsev et al. [412] überein. Die Parameter, die die höchste Genauigkeit erzielen werden mit **fetten** Zahlen dargestellt.

Tiefes Lernen II, C/RNN

In einem vorläufigen Experiment werden alle RNN-, LSTM- und BLSTM-Blöcke identisch hinsichtlich der Anzahl der verborgenen Schichten und der Anzahl der LSTM-Zellen, untersucht, um die optimale Architektur zu finden. Zur Bewertung werden verschiedene Architekturen wie RNN, LSTM, BLSTM und CNN-LSTM untersucht. Diese Experimente ergeben, dass zwei Faltungsschichten und ein BLSTM-Block mit 128 LSTM-Zellen die höchste Genauigkeit bei kurzen Inferenzzeiten liefern. In den weiteren Abschnitten dieser Arbeit repräsentiert C/RNN von nun an die Kombination von CNN und BLSTM. Und eine Faltung mit 128 Filtervektoren $f_v=1 \times 3$ gleicht am besten zwischen Genauigkeit und Inferenzzeit aus. Die Eingabesequenzen werden nicht aufgefüllt, da die Eingabefenster immer eine konstante Länge haben. Es wird zur Vermeidung von Überanpassung und zur Modellkalibrierung die Verwerfungsrate auf 50% gesetzt. Es werden verschiedene Abtastraten (50, 100 Hz) und Fensterlängen (64, 128) für 1D- oder 2D-SMVs von w bewertet. Die Sequenzlänge s (d. h. $s=w$) bei $N_w=128$ enthält maximal 2.56 s (128/50 Hz) Bewegungsdaten. Jeder Parametersatz wird für 33 Epochen trainiert, wobei ein vorzeitiger Stop basierend auf der Leistung des Validierungssatzes durchgeführt wird, um eine Überanpassung zu verhindern.

Der Vollständigkeit halber werden hier kurz die Schlüsselaspekte der C/RNN Methode skizziert: 1D-Trägheitsdaten-Eingabesequenz, mit $m=128$ Messungen pro Eingabefenster, einer ersten Faltungsschicht (128 Filterkerne, jeweils mit der Größe 1×3), gefolgt von einer BN und einer ReLU, einer zweiten Faltungsschicht (128 Filterkerne der Größe 1×3), gefolgt von einer BN- und einer ReLU-Schicht, einer Abflachungsschicht, eine BLSTM-Schicht (=2·LSTM-Schichten), eine Verwerfungs schicht, eine vollständig verbundene Schicht (liefert eine Ausgabe mit der Dimension 1×1) und eine endgültige Regressionsschicht, die den Verlust mit einem halb-MS berechnet. Die Architektur wurde auch für den hier vorgeschlagenen Datensatz in einer Rastersuche optimiert⁶,

⁶C/RNN-Rastersuchparameter: Löser: **Adam**; $\beta_1, \beta_2=0.01$; Momentum=0.9; B/LSTM Schichten=2; LSTM Zellen pro Schicht=128; LR=0.01; LR Verwerfungsperiode=10 Epochen; LR Verwerfungsrate pro Epoche=0.9; Stapelgröße=512; Mischen ∈[Nein,pro Epoche]; Gradientenbeschränkung=max(Eingabe); $dp \in [10, 20, 50, 75]\%$; CNN $F_s=d \times N$ mit $d=2$ Dimensionen und $N=3$; CNN $S=1 \times N$ mit $N=2$; Für jede Kombination von f_s (50, 100, 200, 400) und N_w (64, 128, 256, 512). Die Parameter, die die höchste Genauigkeit erzielen werden mit **fetten** Zahlen dargestellt.

um den ressourcenschonenden Konkurrenten von der vorgeschlagenen Hybridarchitektur für Abschnitt 5.7 zu erstellen.

Tiefes Lernen III, Hybrid

Um die optimale Architektur und ihre Parameter für die Hybridmethode zu finden, wurde auch eine Rastersuche durchgeführt⁷. Dabei wurden auch verschiedene Architekturen wie LSTM, BLSTM, ResNet und Kombinationen davon durchsucht. Es wurden alle LSTM- und BLSTM-Blöcke identisch mit Bezug zur Anzahl der verdeckten Zwischenschichten und zur Anzahl der Zellen untersucht. ResNet-18 [176] wurde an die Eingabe- und Ausgabedimensionen angepasst, die je nach Verfahren die höchste Genauigkeit liefert. Es wurden Verwerfungsschichten zwischen oder nach den BLSTM-Schichten getestet. Verwerfungsschichten nach BLSTM-Schichten liefern bessere Ergebnisse, mit einer idealen Verwerfungsrate von 50%. Die verschiedenen Abtastraten (50, **100**, 200 und 400 Hz) und Fensterlängen (64, **128**, 256 und 512 Hz) wurden auf 1D- oder 2D-SMVs ausgewertet (beste Konfiguration in **Fettdruck**). Die Eingangssequenzlänge von $N_w=128$ enthält maximal 1.28 s (128/100 Hz) Bewegungsdaten. Jeder Parametersatz wurde für 100 Epochen trainiert, wobei das frühzeitige Anhalten auf der Leistung, bzw. Genauigkeit, des Validierungssatzes basiert, um eine Überanpassung zu verhindern. Die resultierende beste Kombination von ResNet-18 und zwei BLSTM-Blöcken (jeweils mit 128 LSTM-Zellen), siehe Abschnitt 5.5.3, liefert die beste Genauigkeit bei kurzen Inferenzzeiten. Die in Abbildung 5.9 verwendeten Parameter wurden auch in der Rastersuche gefunden und lieferten die genauesten Geschwindigkeitsschätzungen.

Als letzte Phase hat die Hybridarchitektur ein LKF. Das LKF wird basierend auf den Vorhersagen v' der Hybridmethode wie folgt parametrisiert. Die Kovarianz des LKF des Messrauschens wird anhand der

⁷ Hybrid Rastersuchparameter: Löser ∈ [SGD, Adam, rmsprop]; $\beta_1, \beta_2=0.01$, Momentum=0.9; B/LSTM Schichten=2 [1, 2, 3, 4], Zellen pro Schicht=128 [16, 32, 64, 128, 256], Anfängliche Lernrate (LR)=0.01 ∈ [1.0:0.1:0.00001]; LR Drop Periode ∈ [0, **10**, 50, 100] Epochen; LR Dropout Rate=0.9; Stapelgröße ∈ [128, 256, 512, **1024**, 2048]; ResNet $F_s=d \times N$ mit $d \in [1, 2]$ Dimensionen und $N=7 \in [3, 4, \dots, 9, 10]$; ResNet $S=1 \times N$ mit $N \in [1, 2, \dots, 9, 0]$; Mischen ∈ [Nein, pro Epoche]; Gradientenbeschneidung=max(Eingabe); $dp \in [10, 20, 50, 75]\%$; Für jede Kombination von f_s (50, **100**, 200, 400) und N_w (64, 128, **256**, 512). Die Parameter, die die höchste Genauigkeit erzielten werden mit **fetten** Zahlen dargestellt.

Vorhersagen v' trainiert und die Kovarianz des Zustandsübergangsrauschens wird anhand der Vorhersagen v' optimiert, bis das LKF eine robuste und plausible Leistung für v' liefert und somit optimierte Geschwindigkeiten v'' auf unbekannten Testdaten vorhersagen kann.

Der Vollständigkeit und Fairness halber wird LKF auch auf die v' -Schätzungen aller anderen Methoden angewendet, um die Auswirkungen eines LKF auf die Genauigkeit jeder Methode zu bewerten. Detaillierte Informationen zu den Auswirkungen eines LKF auf die Ergebnisse der einzelnen Methoden werden in Abschnitt 5.7.2 beschrieben.

5.7 Auswertung und Diskussion

In diesem Abschnitt werden die Ergebnisse diskutiert. Abschnitt 5.7.1 bereitet die Datensätze für verschiedene Auswertungsszenarien vor. In Abschnitt 5.7.2 wird die Genauigkeit der fünf Schätzer für Validierungs- und Testdatensätze bewertet. In Abschnitt 7.7 in Kapitel 7 wird die beste Methode hinsichtlich ihrer Genauigkeit bei der Rekonstruktion der Posen und Trajektorie bewertet. Abschnitt 5.7.3 vergleicht den Rechenaufwand aller Methoden und Abschnitt 5.7.4 diskutiert ihre Generalisierbarkeit. Abschnitt 5.7.5 diskutiert, wie sich unterschiedliche Fenstergrößen auf die datengetriebenen Methoden auswirkt.

Zu beachten ist, dass nachfolgend für die Methoden die folgenden Namen verwendet werden: PDR (klassisches schrittbasierter PDR-Referenzverfahren), ML-GP (ML-Referenzverfahren), RoNIN (DL-Referenzverfahren), C/RNN [128] und Hybrid [131].

5.7.1 Datensatzvorbereitung

Im Rahmen der Forschungsarbeit an der Hybridmethode wurden keine öffentlich verfügbaren Datensätze, wie z. B. der OxIOD-Datensatz [69], der RIDI-Datensatz [489] oder der RoNIN-Datensatz [490], verwendet, da diese nur langsame Bewegungen abdecken (Durchschnittsgeschwindigkeit für OxIOD: 3.1 km/h; RIDI: 4.3 km/h; RoNIN: 3.8 km/h) und weil deren Trajektorien (0.3 m in 10 min) und deren Orientierungen ($20^\circ/min$) drifteten [490]. Diese Datensätze stellen im Gegensatz zu dem hier vorgeschlagenen Datensatz (in Abschnitt 5.6.3) weder typische zufällige Bewegungen mit abrupten Änderungen der Bewegung und Ausrichtung dar, noch decken sie

5 Geschwindigkeitsschätzung

Variationen in Trajektorieformen und Bewegungsraten ab.

Die Auswertung in diesem Kapitel vergleicht die Methoden in drei Dimensionen: Genauigkeit der Geschwindigkeitsschätzung, Rechenaufwand und Generalisierbarkeit unbekannter Daten. Kapitel 7 vergleicht die Genauigkeit der Trajektorienrekonstruktion. Daher benötigen diese Auswertungen verschiedene Teilmengen von Daten, die vorbereitet werden müssen (siehe unten), bevor die Daten in Abschnitt 5.7.2 und folgenden zur Auswertung verwendet werden können. Zunächst wird der Pool von 23 Probanden in drei Gruppen aufgeteilt. 22 Benutzer stellen Daten von denselben Trägheitssensoren bereit. Ein Benutzer stellt Daten von zwei speziellen Trägheitssensoren bereit, diese Daten dieser unbekannten Geräte werden in der Generalisierbarkeitsauswertung (siehe Abschnitt 5.7.4) verwendet. Von den 22 Benutzern werden Daten aus zufällig ausgewählten 20 Benutzern in der Genauigkeitsauswertung verwendet (siehe Abschnitt 5.7.2) und in der Auswertung des Rechenaufwands (siehe Abschnitt 5.7.3). Die Daten der verbleibenden 2 Benutzer (sogenannte Ausgelassene Probanden) werden für die Trajektorienrekonstruktion und auch für die Generalisierbarkeitsauswertung (siehe Kapitel 7 Abschnitt 7.7 und 7.8) verwendet.

Genauigkeitsauswertung

Die Genauigkeitsauswertung in Abschnitt 5.7.2 arbeitet mit Daten der 20 zufällig ausgewählten Probanden. Um die Genauigkeit der Geschwindigkeitsschätzung explizit zu untersuchen, werden alle fünf Methoden für *bekannte* Trainingsdaten (70% = 14 Probanden), *quasi unbekannte* Validierungsdaten (10% = 2 Probanden) und *unbekannte* Testdaten (20% = 4 Probanden) ausgewertet. Die ausgelassenen Probanden waren zu keinem Zeitpunkt Teil der Genauigkeitsauswertung. Die Daten der 20 Probanden wurden mit der optimalen Parameterkonfiguration ($N_w=128$ und $f_s=100$ Hz) vorverarbeitet, wie in den Abschnitten 5.5.1 und 5.6.3 beschrieben, um die Segment-, Fenster- oder Merkmalbündel (und ihre Varianten) zu erhalten, die jeweils in 70/10/20 aufgeteilt sind. Die 10% Validierungsdaten werden verwendet, um eine Überanpassung der datengetriebenen Methoden zu verhindern.⁸ Die 20%-Testdaten

⁸Überanpassung: Während des Trainings bewertet eine datengetriebene Methode ihre Genauigkeit am Ende jeder Trainingsepoke anhand der Validierungsdaten. Das Training wird beendet, sobald eine Methode anfängt, verschlechternde Ergebnisse in den Validierungsdaten anzuzeigen [410].

Tabelle 5.1: Statistik des Datensatzes der Geschwindigkeitsschätzung.

Name	Prob. [#]	Total Segment [#]	Fenster/ Merkmal	Training Segment	Fenster/ Merkmal	Valid. Segment	Fenster/ Merkmal	Test Segment	Dauer [min]	Distanz [km]	v_{ref} [m/s] \varnothing min max
Genauigkeit											
V1	20	160	112.500	112	78.750	16	11.250	32	22.500	1.203	185.40 2.5 0.8 7.9
V2	20	160	97.950	112	68.565	16	9.795	32	19.590	1.045	161.42 2.6 0.8 7.8
V3	20	-	31.200	-	21.840	-	3.120	-	6.240	666	71.9 1.8 0.8 3.6
Trajektorie											
Ausgelassene Prob.	2	16	9.795	-	-	-	-	16	9.795	105	16.14 2.5 0.7 3.7
Rechenaufwand											
V1	20	160	112.500	112	78.750	16	11.250	32	22.500	1.203	185.40 2.5 0.8 7.9
Generalisierbarkeit											
Ausgelassene Prob.	2	16	9.795	-	-	-	-	16	9.795	105	16.14 2.5 0.7 3.7
Unbekannte Geräte	1	8	4.898	-	-	-	-	8	4.898	52	8.07 2.6 1.3 3.5
Interpolation (V3)	20	-	31.200	-	24.960	-	2.059	-	4.181	666	71.9 1.8 0.8 3.6
Extrapolation (V3)	20	-	31.200	-	24.960	-	2.059	-	4.181	666	71.9 1.8 0.8 3.6

werden verwendet, um die Genauigkeit des Trainings der Methoden zu messen. Detaillierte Statistiken und Größen zu Datensätzen sind im oberen Teil von Tabelle 5.1 zu finden.

Zu beachten ist, dass die für beide Sensoren (linke und rechte Hosentasche) zurückgelegte Gesamtstrecke und die Gesamtzahl der Fenster (214 km und 129.375) in Abschnitt 5.6.3 alle 23 Probanden widerspiegeln. Tabelle 5.1 zeigt Werte für die Anzahl der Probanden, die für eine Teilmenge der Daten ausgewählt wurden. Z. B. hat V1 160 Segmente oder 112.500 Fenster- oder Merkmalbündel, da die Variante Daten von 20 Benutzern·2 Sensoren·30min·60s·100 Hz Abtastrate/64 Hz pro 50% überlappende Fenstergröße ($N_w/2=128/2=64$) für 4 Aktivitäten enthält.

Für die Auswertungen werden immer die leistungsstärksten Schätzerkonfigurationen verwendet. Für PDR wurde dies durch Training auf dem V2 Datensatz erreicht. Die datengetriebenen Methoden zeigten die beste Leistung, wenn sie auf V3 trainiert wurden.

Auswertung des Rechenaufwands

Um den Rechenaufwand aller fünf Methoden zu untersuchen, werden ihre Trainings- und Inferenzzeiten gemessen, d. h. wie lange sie brauchen, um Ergebnisse aus einer bestimmten Anzahl von Eingabesequenzen vorherzusagen. Es wird dieselbe Teilmenge von V1 verwendet, die in der Genauigkeitsauswertung verwendet wird, siehe die (red-

5 Geschwindigkeitsschätzung

undante) dritte Zeilengruppe in Tabelle 5.1. Die Trainingszeiten werden am Trainingssatz und die Inferenzzzeiten am Testsatz gemessen. Für Hybrid wird auch die Trainingszeiten des Nachbearbeitungsschritts am Validierungssatz gemessen. Sowohl die Trainings- als auch die Inferenzzzeiten hängen stark von den Parametern des Verfahrens ab, z. B. Breite, Tiefe und Zelltyp eines neuronalen Netzes.

Generalisierbarkeitsauswertung

In dieser Auswertung gibt es vier Experimente. Siehe die letzte Zeilengruppe in Tabelle 5.1. Um die Generalisierbarkeit der fünf Methoden für unbekannte Daten zu untersuchen, wird ihre Leistung bei (1) den unbekannten Daten der ausgelassenen Probanden und (2) des Benutzers mit zwei speziellen Sensoren untersucht.

Um auch die Generalisierbarkeit der vier datengetriebenen Methoden für unbekannte *Geschwindigkeiten* (3 + 4) zu untersuchen, müssen diese Verfahren von Grund auf neu trainiert werden. Hierfür werden die V₃-Daten der 20 Probanden, ohne die ausgelassenen Probanden und den unbekannten Benutzer der Spezialgeräte, verwendet. Die Gründe für diese Wahl sind die Vergleichbarkeit mit den anderen Auswertungen und die Tatsache, dass die datengetriebenen Methoden auf V₃ besser abschneiden als auf V₁ und V₂.

(3) Es wird untersucht, ob die Methoden zwischen zwei Geschwindigkeitsbereichen interpolieren können. Dazu werden die Methoden mit Fenstern (oder Merkmalen) mit langsamem ($v_{ref} \in [3.7] \text{ km/h}$) und schnellen ($v_{ref} \in (9.13] \text{ km/h}$) Geschwindigkeiten trainiert. Zu beachten ist, dass V₃ diese Geschwindigkeiten in vier seiner fünf 2 km/h-Gruppierungen hält. Um die Interpolationsfähigkeiten zu messen, wird die verbleibende Gruppierung mit normalen Geschwindigkeiten ($v_{ref} \in [7, 9] \text{ km/h}$) verwendet und diese Daten in Validierungs- (33% = 2.059 Fenster/Merkmale) und Testdatensätze (67% = 4.181 Fenster/Merkmale) aufgeteilt.

(4) Um die Extrapolationsfähigkeiten der Methoden zu testen, enthalten die Trainingsdaten jetzt langsame bis normale Geschwindigkeiten ($v_{ref} \in [3.11] \text{ km/h}$), d. h. die ersten vier Gruppierungen. Die verbleibende Gruppierung, die schnellen Geschwindigkeiten ($v_{ref} \in [9.13] \text{ km/h}$), bilden die Auswertungsdaten. Diese Daten werden erneut in Validierungs- (33% = 2.059 Fenster/Merkmale) und Testdatensätze (67% = 4.181 Fenster/Merkmale) aufgeteilt.

Zu beachten ist, dass der Einfachheit halber in (3 + 4) die 5 Gruppierungen von V₃, mit jeweils 6.240 Fenstern von 20 Benutzern 2 Sensoren 156 Fenstern 1 pro Gruppierung (Gruppengröße von 2 km/h), wiederverwendet und neu angeordnet werden, siehe Abschnitt 5.6.3 für Details. Natürlich kann die Verteilung von v_{ref} innerhalb jeder Gruppierung geringfügig variieren, aber die Anzahl der Fenster/Merkmale pro Gruppierung ist gleich.

5.7.2 Genauigkeit

Nach der Einführung der Metriken zur Bewertung der Genauigkeit der Geschwindigkeitsschätzung wird die Genauigkeit aller Methoden für die Datensatzvarianten V₁ bis V₃ diskutiert. Schließlich werden die Auswirkungen des LKF bewertet.

Metriken

Die Genauigkeit eines Geschwindigkeitsschätzers ergibt sich aus (1) dem Fehler der Geschwindigkeit und (2) dem Fehler der zurückgelegten Strecke. Für Erstere werden vier Fehlervariablen und zwei Fehlerfunktionen verwendet: Der mittlere absolute Fehler (engl. mean absolute error, MAE), der mittlere quadratische Fehler (engl. mean square error, MSE), das mittleres Abweichungsquadrat (engl. Root Mean Square Error, RMSE) und die zirkuläre Fehlerwahrscheinlichkeit (engl. Circular Error Probability, CEP). Zwei Funktionen helfen, die Fehler als Diagramme darzustellen: Die kumulative Verteilungsfunktion (engl. Cumulative Distribution Function, CDF) und die Wahrscheinlichkeitsdichtefunktion (engl. Probability Density Function, PDF). Um den Fehler der zurückgelegten Strecke (2) zu messen, werden zwei Fehlervariablen verwendet: Der Entfernungsfehler pro Meter (engl. Distance Error Per Meter, DEPM) und die Entfernungsfehlersumme (engl. Distance Error Total, DET). Es wird auch die kumulative DET (über die Zeit) gezeichnet. Alle Fehler werden auf der geschätzten Geschwindigkeit v' , der optimierten Geschwindigkeit v'' und der Referenzgeschwindigkeit v_{ref} , respektive über die Zeit für die Abstandsfehler, bestimmt. Während MAE, MSE, RMSE und CEP sowie DEPM und DET in Tabellen aufgelistet werden, werden die Fehlerfunktionen CDF und PDF sowie den Gesamtentfernungsfehler DET in Abbildungen visualisiert.

5 Geschwindigkeitsschätzung

Diese Metriken und Grafiken wurden aus folgenden Gründen ausgewählt: MAE gibt einen guten und leicht lesbaren Überblick über die Genauigkeit eines Schätzers. Je niedriger der Wert ist, desto besser ist die Leistung eines Modells. MAE berücksichtigt jedoch weder die Richtung (negative Vorzeichen) noch Ausreißer (unerwartete Werte, die zu hoch oder zu niedrig sind). Ausreißer tragen jedoch stärker zum MSE bei, da die Werte quadriert werden. CEP wird mit einer Wahrscheinlichkeit indiziert, z. B. CEP_{95} . Der Fehler gibt an, wie viele Fehler innerhalb des angegebenen Perzentils liegen.

Verteilungsfunktionen, CDF-Diagramme, bilden den Schätzfehler auf eine Wahrscheinlichkeit in $[0.0; 1.0]$ ab. Es ist die Wahrscheinlichkeit, dass alle Schätzungen einer Methode kleiner oder gleich diesem Fehler sind. In den CDF-Diagrammen in Abbildung 5.15 weisen Methoden mit kleinen Fehlern Kurven auf, die sich der oberen linken Ecke nähern, während Methoden mit größeren Fehlern eher rechts unten liegen. Je steiler eine Kurve ist, desto wahrscheinlicher ist es, dass die Methode nur kleine Fehler macht, und desto besser ist ihre Leistung. Im Gegensatz dazu beschreiben die PDF-Diagramme (siehe Abbildung 5.16) die Wahrscheinlichkeit, einen bestimmten Fehlerwert von v' zu erhalten. Je höher und schmäler die Amplitude eines PDFs ist und je zentrierter diese um Null (auf der x-Achse) liegt, desto weniger Fehler macht ein Schätzer und desto robuster ist er.

DEPM und DET befassen sich mit der zurückgelegten Strecke. DEPM ist der absolute Fehler zwischen dem Referenzabstand d_{ref} und dem geschätzten Abstand d_{est} pro Meter. DET ist der absolute Fehler zwischen dem *totalen* Referenzabstand $\sum_1^n d_{ref(i)}$ und dem geschätzten Gesamtabstand $\sum_1^n d_{est(i)}$ eines vollständigen Datensatzes. Der DET wird über die Zeit visualisiert, siehe z. B. Abbildung 5.17, die die Referenz und die geschätzten absoluten Entfernungen zeigt, die über die Zeit zurückgelegt wurden. Je näher die Linie einer Methode an der Referenz liegt, desto besser ist die Leistung der Methode.

Standardmäßig verwendet Hybrid ein LKF als letztes Prozessglied. Da im Rahmen der Dissertation herausgefunden wird, dass das Hinzufügen eines LKF auch die Ergebnisse der anderen Schätzer verbessert, werden im Interesse eines fairen Vergleichs auch, LKF-freie Bewertungszahlen und Diagramme für Hybrid in Tabelle 5.2 und

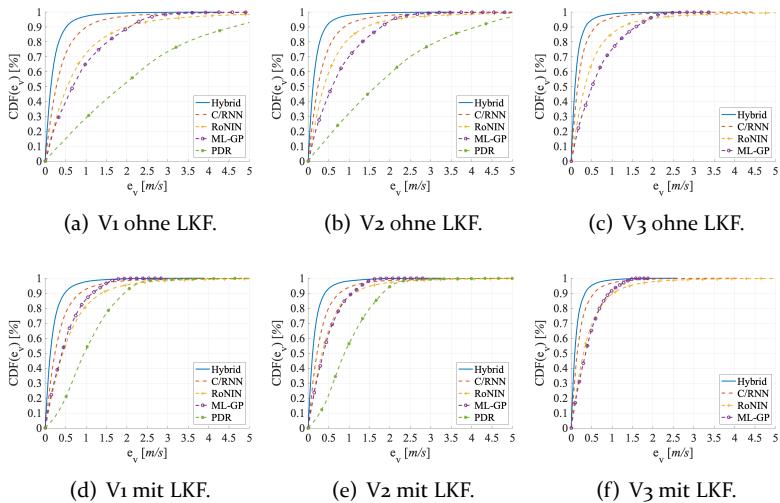


Abbildung 5.15: CDFs der Methoden auf den Testdatensätze V1-V3, sowohl ohne als auch mit LKF.

Abbildungen 5.15 bis 5.17 aufgeführt.

Im Folgenden werden die Ergebnisse des Genauigkeitsbenchmarks für die drei Datensätze diskutiert, indem zunächst ein kurzer Überblick über die Geschwindigkeitsschätzungsfehler der Methoden, ihre Robustheit und ihre Entfernungsschätzungsfehler gegeben wird. Dann wird die Leistung der Methoden in Bezug auf f_s und f_t verglichen. Anschließend werden auch andere interessante Ergebnisse diskutiert, bevor zusammengefasst wird.

Auswertung der Methoden zu V1

Zu beachten ist, dass V1 Aktivitäten enthält, deren Dauer zwischen den Benutzern variiert. Daher müssen sich die Schätzer mit redundanten Informationen befassen, da weder die Aktivitäten noch die Geschwindigkeiten gleichmäßig verteilt sind. Die Tabelle 5.2 enthält die Ergebnisse in ihrer oberen Zeilengruppe. Die grauen Linien können vorerst ignoriert werden. Es gibt zwei allgemeine Erkenntnisse aus den Messungen: (a) Alle Schätzer schneiden bei der Validierung besser ab

5 Geschwindigkeitsschätzung

Tabelle 5.2: Genauigkeit (höchste **fett) der Geschwindigkeits schätzer auf den Datensätzen V1 - V3 in m/s, Distanzfehler in m.**

Datensatz mit 20 Prob.	MAE	MSE	PDR K_0	CEP $_{95}$	DEPM	DET	MAE	MSE	RMSE	ML-GP	DET	MAE	MSE	RMSE	RONIN	C-RNN	Hybrid	DET	MAE	MSE	RMSE	DET	MAE	MSE	RMSE	CEP $_{95}$	DEPM	DET	MAE	MSE	RMSE	DET	MAE	MSE	RMSE	CEP $_{95}$	DEPM	DET
V1 Valid.	2.01	6.31	2.51	4.88	0.348	977.00	0.82	1.17	1.08	2.16	0.077	21.05	0.84	1.94	1.39	2.76	0.01	37.21	0.44	0.52	0.72	1.44	0.018	18.71	0.23	0.14	0.37	0.72	0.022	9.46								
V1 Test	2.16	7.37	2.71	5.25	0.374	203.80	0.80	1.38	1.17	2.37	0.088	16.95	0.88	2.15	1.46	2.86	0.013	49.47	0.47	0.59	0.77	1.49	0.019	39.59	0.25	0.16	0.40	0.77	0.022	15.65								
V1 Test mit LKF	1.06	1.55	1.35	2.15	0.435	1451.61	0.51	0.44	0.66	1.41	0.065	369.80	0.64	0.93	0.57	1.91	0.178	101.70	0.38	0.39	0.62	1.22	0.055	38.53	0.22	0.13	0.36	0.70	0.061	5.32								
V2 Valid.	1.84	5.38	2.32	4.51	0.337	87.31	0.71	0.99	0.95	1.91	0.064	4.80	0.61	1.04	1.02	1.98	0.004	10.90	0.40	0.43	0.66	1.27	0.022	10.69	0.20	0.11	0.33	0.65	0.024	3.19								
V2 Test	1.93	5.82	2.41	4.60	0.328	806.20	0.77	1.04	1.02	2.05	0.074	76.96	0.64	1.15	1.07	2.07	0.005	206.69	0.42	0.48	0.69	1.33	0.022	66.70	0.22	0.12	0.35	0.67	0.023	8.38								
V2 Test mit LKF	0.99	1.35	1.16	2.01	0.410	1010.47	0.47	0.57	0.61	1.30	0.054	267.01	0.51	0.64	0.80	1.56	0.127	62.35	0.34	0.33	0.57	1.11	0.043	36.98	0.19	0.10	0.32	0.61	0.004	4.93								
V3 Valid.	-	-	-	-	-	-	0.03	0.70	0.88	1.76	0.036	26.77	0.50	0.71	0.84	1.65	0.030	31.81	0.25	0.17	0.41	0.77	0.005	14.95	0.35	0.66	0.24	0.44	0.013	8.62								
V3 Test	-	-	-	-	-	-	0.67	0.78	0.88	1.87	0.047	39.84	0.55	0.85	0.92	1.86	0.042	45.08	0.28	0.21	0.46	0.89	0.010	21.50	0.17	0.07	0.27	0.53	0.006	12.66								
V3 Test mit LKF	-	-	-	-	-	-	0.43	0.39	0.55	1.16	0.076	113.83	0.47	0.52	0.72	1.45	0.135	22.52	0.24	0.17	0.41	0.80	0.026	14.80	0.16	0.07	0.26	0.50	0.003	3.75								

als beim Testdatensatz (RMSE=-3% im Durchschnitt). Zu beachten ist, dass Hybrid standardmäßig das LKF verwendet (siehe graue Linien in Tabelle 5.2), aber für einen fairen Vergleich wird das Filter derzeit nicht in den Bewertungen verwendet. (b) Beim Vergleich der Schätzer untereinander gibt es sowohl im Test als auch im Validierungssatz dieselbe allgemeine Reihenfolge zwischen den Schätzern (für alle Fehlermetriken). Auf dem Testdatensatz schneidet Hybrid am besten ab (MAE=0.25 m/s), gefolgt von C/RNN (MAE=0.47 m/s). RoNIN (MAE=0.88 m/s) und ML-GP (MAE=0.89 m/s) sind gleichwertig. Dies steht im Einklang mit der CDF-Abbildung 5.15(a), die zeigt, dass alle vier datengetriebenen Modelle in 70% aller Fälle Schätzungen mit einem Fehler unter 1.25 m liefern. PDR (mit K_g) liefert den schlechtesten MAE (=2.16 m/s), obwohl K_g 70% der Bewegungsinformationen von V1 einbettet. Die Reihenfolge der CEP_{95} Ergebnisse der Methoden stimmt mit der MAE-Reihenfolge überein.

Angesichts der Robustheit der Methoden gegenüber kleinen und großen Ausreißern zeigen sowohl die MSE- als auch die RMSE-Zahlen einen ähnlichen Trend wie MAE und CEP_{95} . Hier hat Hybrid (am besten) weniger große Ausreißer (RMSE=0.40 m/s), gefolgt von C/RNN (RMSE=0.77 m/s), während RoNIN (RMSE=1.46 m/s) und ML-GP (RMSE=1.17 m/s) schwerwiegendere Probleme haben. PDR hat die größten Ausreißer (RMSE=2.71 m/s). Sie können durch falsch erkannte oder nicht erkannte Schritte verursacht werden. Interessanterweise scheint ML-GP (MSE=1.38 m/s) Ausreißer etwas besser zu behandeln als RoNIN (MSE=2.15 m/s). Dies kann daran liegen, dass RoNIN fälschlicherweise räumliche Merkmale f_s aus verschiedenen Eingangsmessungen einem einzelnen Fingerabdruck (Referenzwert) zuordnet. Die PDF-Abbildung 5.16(a) unterstützt diese Ergebnisse und unterstreicht die Robustheit von C/RNN und Hybrid, da sie engere und höhere Amplituden (um Null) bilden als die anderen Schätzer.

Die Entfernungsfehler (DET und DEPM) zeigen, dass PDR ernsthafte Probleme hat ($DET=2.1$ km), gefolgt von RoNIN ($DET=460.47$ m), die Distanzen genau zu bestimmen, während die anderen drei Methoden kaum oder gar keine Anomalien zeigen. In der DET-Abbildung 5.17(a) sind sowohl PDR als auch RoNIN weiter von der Referenz entfernt als die anderen drei Methoden. Interessanterweise macht PDR zwar große DET- und DEPM- ($=0.37$ m) Fehler, aber für RoNIN korreliert sein hoher DET-Wert nicht mit seinem niedrigen DEPM-Wert. Offensichtlich

5 Geschwindigkeitsschätzung

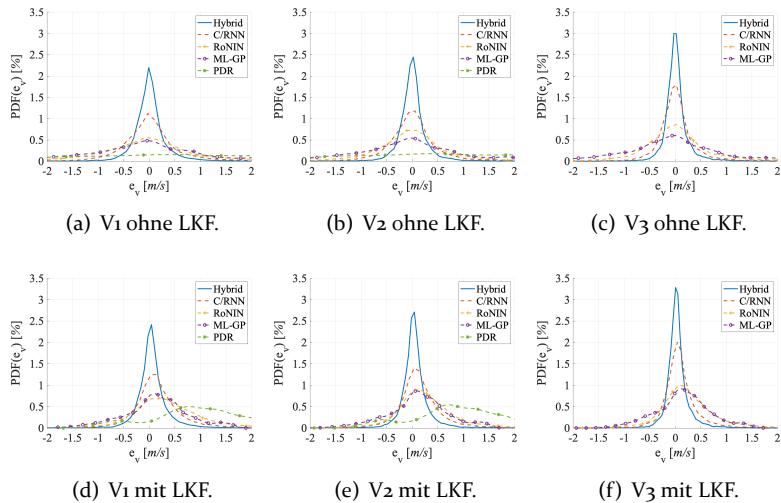


Abbildung 5.16: PDFs der Methoden auf den Testdatensätze V1-V3, sowohl ohne als auch mit LKF.

korreliert DEPM nicht unbedingt direkt mit DEPM, stattdessen deuten die großen RMSE-Werte (PDR: 2.71 m/s; RoNIN: 1.46 m/s) darauf hin, dass es Ausreißer gibt, die die hohen DET-Werte verursachen können.

Die Auswirkungen von f_s und f_t auf die Genauigkeit zeigen, dass ML-GP RoNIN übertrifft, da es Gaußsche Verteilungen zum Speichern von f_s erstellt, die möglicherweise unnötige Informationen verworfen. Ein Vergleich der ML-GP-Kurve mit der RoNIN-Kurve bei $e_v > 2$ m/s in der CDF-Abbildung 5.15(a) untermauert die These und zeigt niedrigere DET-Werte von ML-GP. Während sich RoNIN im Verlauf des Trainings immer mehr auf die redundanten Daten konzentriert, scheint ML-GP sie intelligent zu gruppieren. Im Gegensatz dazu scheinen sich die Speicherzellen von C/RNN und Hybrid daran zu erinnern, welches Wissen in f_t gespeichert und welches davon vergessen werden soll. Die Verfolgung des Auftretens von f_s im Laufe der Zeit scheint zur Genauigkeit von f_t beizutragen. C/RNN ist jedoch in allen Metriken signifikant schlechter als Hybrid (C/RNN: $CEP_{95}=1.49$ m/s gegenüber Hybrid: $CEP_{95}=0.77$ m/s). Dies kann darauf hinweisen, dass der viel tiefere Merkmalsextraktor (ResNet) von Hybrid eine bessere Leistung

aufweist als der flache von C/RNN (2-Faltungsschichten), da der Rest ihrer Architekturen ähnlich ist. Die RMSE-Werte sind auch ein früher Hinweis darauf, dass Zeit und Kontext f_t einen signifikant positiven Einfluss auf die Genauigkeit zu haben scheinen: C/RNN und Hybrid, die f_s und f_t ausnutzen, führen zu viel geringeren Fehlern (RMSE=0.77 m/s und 0.40 m/s) als ML-GP und RoNIN, die nur f_s ausnutzen (RMSE=1.17 m/s und 1.46 m/s). Dies steht im Einklang mit der CDF-Abbildung 5.15(a), die zeigt, dass in 90% aller Fälle die datengetriebenen Methoden, die f_t berücksichtigen, Fehler unter 1 m liefern, während die anderen Methoden doppelt so große Fehler liefern.

Eine Vorstudie der PDR auf den Daten von V1 zeigte, dass selbst bei subjekt- und aktivitätsspezifischen Koeffizienten K_{sa} die Fehler hoch bleiben (MAE=1.97 m/s, RMSE=2.41 m/s und DEPM=0.31 m). Während PDR mit K_g einen hohen Distanzfehler (DET=2.103 m) hat, verbessert sich PDR mit K_{sa} nur minimal (DET=1.73 m). Nur ein K_{sa} zusammen mit einer Feinabstimmung der Schrittdetektion, mit einem Schwellenwert, der für den Spitzendetektor für ein bestimmtes Testsegment optimiert wurde, ergibt geringere Fehler (MAE=1.13 m/s, RMSE=1.87 m/s und DEPM=0.27 m/s). Aber selbst mit einer aufwendigen und unpraktischen Optimierung, die eine Aktivität im Sensorstrom vorklassifiziert und ein geeignetes subjektaktivitätsspezifisches K_{sa} auswählt, ist die PDR schlechter als die vier datengetriebenen Schätzer.

Zusammenfassung: PDR fällt zurück und liefert unpraktisch große Fehler über alle Fehlermetriken hinweg. Die redundanten Informationen in V1 sind ein erhebliches Handicap für die Methoden ML-GP und RoNIN, die nur f_s verarbeiten. Stattdessen scheinen die Methoden C/RNN und Hybrid, die f_s und f_t verfolgen, diese Nachteile zu überwinden und zeigen die kleinsten Fehler über alle Metriken hinweg.

Auswertung der Methoden zu V2

Die Aktivitäten in V2 werden so beschnitten, dass sie dieselbe Dauer haben. Die Tabelle 5.2 enthält die Ergebnisse in ihrer mittleren Zeilengruppe. Die allgemeine Kernaussage der Auswertung besteht darin, dass beim Wechsel von V1 nach V2 die Beziehungen der Fehlermetriken zwischen den fünf Schätzern sowohl für die Testdaten als auch für die Validierungsdaten gleich bleiben. Der Wechsel von V1 zu V2 verringert jedoch alle Fehler in allen Metriken für alle fünf Schätzer.

5 Geschwindigkeitsschätzung

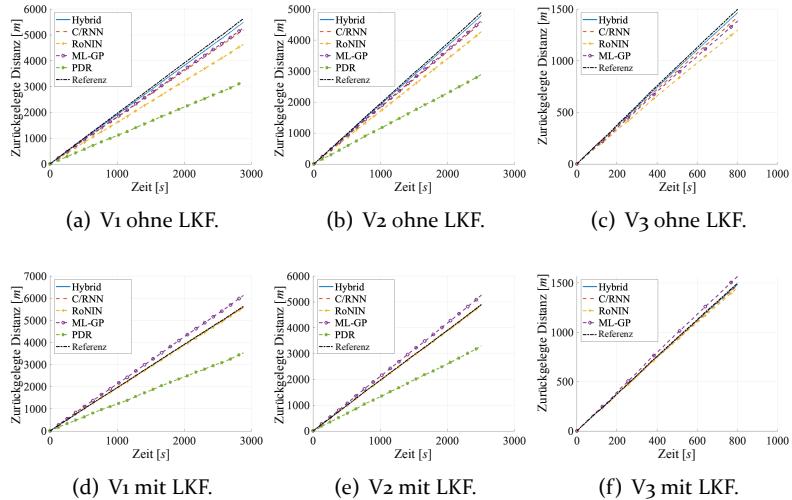


Abbildung 5.17: DETs im Zeitverlauf der Methoden auf den Testdatensätze V₁-V₃, sowohl ohne als auch mit LKF.

Z. B. verringern sich die MAE- und CEP_{95} -Fehler im Durchschnitt um: PDR: -13%, ML-GP: -17%, RoNIN: -49%, C/RNN: -6% und Hybrid: -3%.

Auf dem Testdatensatz profitiert RoNIN mehr als ML-GP beim Wechsel von V₁ zu V₂ (MAE=-0.24 m/s gegenüber -0.21 m/s und CEP_{95} : -0.79 m/s gegenüber -0.32 m/s). Eine logische Schlussfolgerung ist, dass RoNIN nicht länger gezwungen ist, redundante Daten f_s zu speichern oder sich nicht länger auf diese überpasst, während ML-GP weniger signifikant von seiner Funktionsstruktur profitiert. Für C/RNN und Hybrid gibt es nur geringfügige Verbesserungen der Genauigkeit (MAE=ungefähr -0.04 m/s und CEP_{95} : ungefähr -0.12 m/s). Dies könnte daran liegen, dass sie bereits nur die wichtigsten Merkmale verfolgen. Ein Vergleich der Kurven in den CDF-Abbildungen 5.15(a) und 5.15(b) zeigt, dass sie alle nach links oben verschoben sind (niedrigere Fehler liefern). Alle vier datengetriebenen Methoden zeigen jetzt Fehler, die in 70% aller Fälle unter 1 m/s liegen, verglichen mit 1.25 m in 70% in V₁. Somit ergibt sich für alle Methoden eine höhere Gesamtgenauigkeit. Wie bei V₁ ist PDR (mit K_g) immer noch schlechter als die vier datengetriebenen Methoden, obwohl beim

Übergang von V₁ zu V₂ ein erheblicher Gewinn zu verzeichnen ist ($MAE = -0.23 \text{ m/s}$ und $CEP_{95} = -0.56 \text{ m/s}$). Dies wird begründet, wenn der Entfernungsfehler diskutiert wird.

Die kleinen und großen Ausreißer der Methoden zeigen, dass der Wechsel von V₁ zu V₂ die Anzahl der großen Ausreißer erheblich verringert, da sowohl MSE als auch RMSE über alle fünf Schätzer hinweg um bis zu 46% (MSE von RoNIN) abnehmen. Ein genauerer Blick zeigt jedoch, dass Hybrid zwar die wenigsten Ausreißer aufweist ($MSE = 0.12 \text{ m/s}$; $RMSE = 0.32 \text{ m/s}$), gefolgt von C/RNN ($MSE = 0.48 \text{ m/s}$; $RMSE = 0.69 \text{ m/s}$) aber die anderen Schätzer haben immer noch ernsthafte Probleme (ML-GP: $MSE = 1.04 \text{ m/s}$; $RMSE = 1.02 \text{ m/s}$; RoNIN: $MSE = 1.15 \text{ m/s}$; $RMSE = 1.07 \text{ m/s}$). Dies entspricht der schmaleren Glockenform aller Kurven mit Ausnahme der PDR-Kurve in der PDF-Abbildung 5.16(b). Die kleineren MSE- und RMSE-Werte in V₂ als in V₁ können darauf hinweisen, dass sich alle Verfahren mehr auf das Entrauschen, d. h. das Herausfiltern kleiner und großer Ausreißer, als auf das Lernen redundanter Informationen konzentrieren und deshalb höhere Genauigkeiten erzielen können.

Die Abstandsfehler wirken sich auf die Robustheit aus. Bei PDR schrumpft der DET erheblich von 2.1 km auf 1.8 km. Dies liegt wahrscheinlich daran, dass die Distanz zwischen den Datensätzen abnimmt (von 5.632 m auf 4.903 m), ebenso wie die Anzahl der möglicherweise falsch erkannten Schritte abnimmt. DET schrumpft auch für RoNIN um beachtliche 250 m, was der RMSE-Reduzierung um etwa 40 m entspricht. Folglich ist die RoNIN-Kurve näher an der Referenz in der DET-Abbildung 5.17(b). ML-GP, C/RNN und Hybrid weisen in allen Metriken ebenfalls geringere Fehler auf, profitieren jedoch nicht so stark wie PDR und RoNIN. In der PDF-Abbildung 5.16(b) haben alle vier datengetriebenen Methoden höhere Spitzen bei Null als mit V₁ in der PDF-Abbildung 5.16(a). Dies impliziert eine geringere Varianz der Schätzer und damit eine höhere Robustheit gegenüber Ausreißern. Der Vergleich der PDF-Abbildungen 5.16(a) und 5.16(b) zeigt einen merklichen Anstieg der Amplituden von RoNIN, C/RNN und Hybrid, die ebenfalls eine engere Glockenform bilden, die eine stärkere Robustheit der Methoden gegenüber Ausreißern impliziert.

Auf V₂ schneidet ML-GP ($MAE = 0.77 \text{ m/s}$) schlechter ab als RoNIN ($MAE = 0.64 \text{ m/s}$), während sie auf V₁ gleichauf waren. Obwohl beide

5 Geschwindigkeitsschätzung

große Ausreißer ähnlich behandeln ($\text{RMSE} \sim 1.0 \text{ m/s}$), ist RoNINs Umgang mit kleinen Ausreißern jetzt näher an ML-GP. Ihre MSEs unterscheidet sich bei V_2 um 0.11 m/s gegenüber 0.77 m/s bei V_1 . Dies und eine Abnahme der DEPM-Werte (ML-GP: 0.014 m gegenüber RoNIN: 0.008 m) führt dazu, dass die ML-GP-Kurve näher an der Referenz liegt als die RoNIN-Kurve, siehe DET-Abbildung 5.17(b). Dies weist darauf hin, dass RoNIN jetzt weniger Schwierigkeiten hat, um wichtige Informationen zu trennen und zu speichern.

Während bei V_2 kein signifikanter Vorteil der f_t -Verfolgungsmethoden (C/RNN und Hybrid) zu erkennen ist, profitieren die Methoden, die f_s verfolgen, am meisten von einem Datensatz mit weniger Redundanz.

Zusammenfassung: Ein Vergleich von V_2 mit V_1 zeigt, dass das Lösen redundanter Informationen aus den Trainingsdaten die Genauigkeit aller datengetriebener Methoden erheblich verbessert. Daher wird diese Art von Trainingsdaten empfohlen.

Auswertung der Methoden zu V_3

Zu beachten ist, dass PDR nicht auf V_3 evaluiert werden kann, da V_3 keine Segmentbündel enthält. In V_3 sind die Geschwindigkeiten für alle Benutzer homogen auf 3 bis 13 km/h verteilt. Es gibt weniger Redundanz als in V_2 . Die Tabelle 5.2 enthält die Ergebnisse in den unteren Zeilen. Die zwei allgemeinen Kernaussagen der Auswertung sind: (1) Beim Übergang von V_1 nach V_3 oder von V_2 nach V_3 bleiben die allgemeinen Beziehungen zwischen den fünf Schätzern sowohl mit den Testdaten als auch mit den Validierungsdaten unverändert. (2) Bei allen vier Schätzern verringern sich alle Fehlermetriken, wenn von V_1 nach V_3 gewechselt wird. Die Reduzierung ist stärker als beim Übergang von V_1 nach V_2 (V_1 nach V_2 : $\text{MAE}=-19\%$, $\text{RMSE}=-17\%$ gegenüber V_1 nach V_3 : $\text{MAE}=-50\%$, $\text{RMSE}=-49\%$). Durch den Wechsel von V_2 zu V_3 werden auch alle Fehler für alle vier Schätzer verringert. Bspw. verringern sich die MAE- und CEP_{95} -Fehler im Durchschnitt um: ML-GP: -12% , RoNIN: -16% , C/RNN: -50% und Hybrid: -20% .

Auf dem Testsatz, während ML-GP ($\text{MAE}=-16\%$ und $CEP_{95}: -16\%$) und RoNIN ($\text{MAE}=-37\%$ und $CEP_{95}: -38\%$) am meisten vom Wechsel von V_1 zu V_2 profitieren, zeigen jetzt C/RNN ($\text{MAE}=-50\%$ und $CEP_{95}: -49\%$) und Hybrid ($\text{MAE}=-19\%$ und $CEP_{95}: -22\%$) die stärksten Verbesserungen.

serungen auf einem viel feinkörnigeren Niveau (Absolutfehler deutlich unter 0.20 m), bei einem Wechsel von V₂ zu V₃. Hybrid (am besten) zeigt den kleinsten Fehler in V₃ (MAE=0.17 m/s; CEP₉₅=0.53 m/s), gefolgt von C/RNN (MAE=0.28 m/s; CEP₉₅=0.89 m/s). RoNIN hat auf V₂ bereits etwas besser abgeschnitten als ML-GP. Auf V₃ kann es seinen Vorsprung gegenüber RoNIN ausbauen (MAE=0.55 m/s gegenüber 0.67 m/s; CEP₉₅=1.80 m/s gegenüber 1.87 m/s). Natürlich schneiden alle vier Methoden in der Genauigkeitsauswertung mit V₃ am besten ab. Der Grund ist, dass V₃ eine statistisch gute Verteilung von Daten und Informationen aufweist. Es ist bekannt, dass eine solche Informationsverteilung zu quasi optimalen Ergebnissen der Schätzer [468] führt. Dies steht auch im Einklang mit der CDF-Abbildung 5.15(c), aus der hervorgeht, dass in 70% aller Fälle die Fehler der vier datengetriebenen Methoden unter 0.75 m liegen. Dies ist nicht nur eine bemerkenswerte Verbesserung um etwa 40%, sondern es ist auch einfach, diese Verbesserung zu erreichen, indem lediglich redundante Informationen aus den Trainingsdaten entfernt werden. Bei V₃ konzentrieren sich alle Methoden wahrscheinlich darauf, die Rohdaten zu entrauschen, die besten f_s - und f_t -Merkmale zu extrahieren und einen klaren Überblick darüber zu haben, was wichtig ist, um es sich zu merken und was nicht.

Der Umgang der Methoden mit kleinen und großen Ausreißern in V₃ zeigt den gleichen Trend wie bei MAE und CEP₉₅. Bei allen vier Methoden nehmen sowohl die MSE- als auch die RMSE-Werte beim Übergang von V₂ zu V₃ erheblich ab. Dies ist in der PDF-Abbildung 5.16(c) zu sehen, in der alle Spitzen höher und alle Amplituden enger sind als in den PDF-Abbildungen 5.16(a) und 5.16(b) von V₁ und V₂. C/RNN profitiert am meisten beim Wechsel von V₂ zu V₃ (MSE=-56%; RMSE=-30%), gefolgt von Hybrid (MSE=-30%, RMSE=-19%) und Ronin (MSE=-26%; RMSE=-14%). C/RNN und Hybrid haben auch die wenigsten Ausreißer (MSE=0.21 m gegenüber 0.07 m und RMSE=0.46 m gegenüber 0.27 m). ML-GP profitiert am wenigsten vom Wechsel von V₂ zu V₃ (MSE=-25%; RMSE=-14%). Bei V₃ scheint das Ergebnis das geringste Rauschen zu enthalten. Alle Methoden scheinen die Daten effizient zu entrauschen und müssen sich nicht mehr auf redundante Daten konzentrieren.

Die Abstandsfehler und ihre Auswirkungen auf die Robustheit profitieren von der Umstellung auf V₃ auf die gleiche Weise wie die Ausreißer. Wie erwartet profitiert C/RNN aufgrund seiner kleinen MSE- und RMSE-Werte am meisten (DEPM: -55%, DET: -68%). Aus

5 Geschwindigkeitsschätzung

diesem Grund gibt es den gleichen Effekt für ML-GP und Hybrid. RoNIN spielt die Rolle des Außenseiters. Sein DEPM wächst um unerklärliche 740%, während sein DET aufgrund des besseren Umgangs mit Ausreißern ebenfalls schrumpft (von 207 m auf 145 m). Im Vergleich zu V₁ und V₂ erzielen alle vier Methoden ihre besten DET-Werte für V₃. Da C/RNN und Hybrid bei einer Gesamtentfernung von 1,5 km die niedrigsten DET-Werte liefern, kommen ihre Kurven der Referenz am nächsten, siehe DET-Abbildung 5.17(c). Aufgrund seiner großen DEPM hält RoNIN größeren Abstand (DET=145 m) von der Referenz.

Ähnlich wie mit dem V₂-Datensatz, erzielen sowohl C/RNN als auch Hybrid (die f_s und f_t verwenden) über alle Metriken hinweg viel kleinere Fehler als ML-GP und RoNIN (die nur f_s verwenden). Daher scheint die Fähigkeit eines BLSTM in Kombination mit einer Verwerfungsschicht (C/RNN und Hybrid), f_t in einem Fenster zu verfolgen, einen positiven Einfluss auf die Genauigkeit zu haben.

Zusammenfassung: Alle datengetriebenen Methoden profitieren von einem Training mit einem Datensatz, der nahezu frei von redundanten Informationen ist. Der Aufwand für eine angemessene Aufbereitung der Daten lohnt sich, da im Durchschnitt alle Fehlermetriken um 40% schrumpfen. Diese Trainingskonfiguration ist empfehlenswert.

Mehrwert der Nachbearbeitung mit einem LKF

Im Folgenden wird diskutiert, welcher Bruchteil der Genauigkeit von Hybrid auf die endgültige LKF-Nachbearbeitung zurückzuführen ist und gezeigt, dass ein LKF auch die Genauigkeit aller Methoden verbessert. Es werden alle drei Datensätze V₁ bis V₃ betrachtet. Die Ergebnisse befinden sich in den grauen Linien der Tabelle 5.2. Es gibt keine Ergebnisse des LKF für die Validierungsdatensätze, da die Validierungsdaten verwendet werden, um die LKF mit den Vorhersagen jeder Methode zu trainieren.

Es gibt drei wichtige Ergebnisse. Erstens erhöht die Nachbearbeitung mit einem LKF die Geschwindigkeits- und die Entfernungsschätzgenauigkeit aller Methoden für alle Datensätze im Durchschnitt um: V₁: 145%, V₂: 70% und V₃: 60%, siehe die schwarzen Linien mit den entsprechenden grauen Linien in Tabelle 5.2. Zweitens profitieren Methoden, die nur mit f_s (PDR, ML-GP und RoNIN) funktionieren,

stärker von der LKF als C/RNN und Hybrid. In V₁ beträgt die durchschnittliche Verbesserung von MAE, MSE, RMSE und CEP_{95} für PDR: -185%, ML-GP: -108%, RoNIN: -67% gegenüber C/RNN: -30% und Hybrid: -14%. Drittens senkt ein hinzugefügtes LKF den DET für alle Methoden (außer ML-GP) um etwa 50%. Es gibt einen ähnlichen Trend für V₂ und V₃, jedoch weniger ausgeprägt.

Sowohl mit V₁ als auch mit V₂ hilft ein hinzugefügter LKF der PDR am meisten (MAE=-1 m; CEP_{95} : -3 m). Die PDR-Kurve ändert sich am meisten zwischen den CDF-Abbildungen 5.15(d) zu 5.15(e) von V₁ und V₂. Der Grund für den starken Effekt könnte die Übereinstimmung zwischen dem Bewegungsmodell des PDR und dem Zustandsübergangsmodell des LKF sein, das zwischen erkannten und nicht erkannten benachbarten Schritten interpoliert und die Vorhersagen glättet. Das hinzugefügte LKF zeigt eine bessere Entrauschen und Interpolation für die Methoden, die mit f_s (ML-GP und RoNIN) arbeiten, als für die beiden f_t -Verfolgungsmethoden (C/RNN und Hybrid). Die CDF-Abbildungen 5.15(d) bis 5.15(f) V₁ bis V₃ unterstreichen dies: Alle Kurven werden nach links oben verschoben (ML-GP übertrifft sogar RoNIN), sodass mit einem angewendeten LKF die Fehler in 70% aller V₃-Fälle unter 0.5 m liegen, was einer Verbesserung von 125% gegenüber den V₁-Ergebnissen ohne LKF entspricht.

In Bezug auf kleine und große Ausreißer zeigt sich der gleiche Trend von MSE und RMSE wie bei MAE und CEP_{95} . Während das LKF Ausreißer für alle Schätzer entfernt, profitiert PDR am meisten (MSE=-5.8 m/s, RMSE=-1.25 m/s), gefolgt von ML-GP (MSE=-0.94 m/s, RMSE=-0.51 m/s) und RoNIN (MSE=-1.22 m/s, RMSE=-0.49 m/s). Das LKF lernt, wie die Schätzer das Rauschen intern charakterisieren, die Validierungsdaten verwenden, um die Schwachstellen dieser Charakterisierungen zu verfolgen und das Entrauschen und damit die MSE- und RMSE-Werte zu verbessern, siehe PDF-Abbildungen 5.16(d) bis 5.16(f), die auch zeigen, dass der Effekt mit V₃ am stärksten ist. Obwohl PDR vom LKF profitiert, behält es seine unpraktisch hohe Unsicherheit bei, da seine Kurve nicht einmal nahe am Zentrum (um Null) liegt. Hybrid liefert die höchste Genauigkeit und zeigt die höchste Robustheit gegenüber kleinen und großen Ausreißern. Auch wenn die Entrauschungsfähigkeiten von Hybrid ohne LKF die anderen Methoden bereits übertreffen, kann die hinzugefügten LKF sie dennoch

5 Geschwindigkeitsschätzung

ein wenig steigern.

Betrachtet man die Abstandsfehler, so zeigen die DET-Abbildungen 5.17(d) bis 5.17(f), dass mit einem LKF alle Kurven außer ML-GP der Referenz viel näher kommen. Die Auswirkung des LKF auf die DEPM- und DET-Werte der Methoden etwas verschwommen: Bei drei Methoden stuft das LKF sogar die durchschnittliche DEPM herab (PDR: +17%, RoNIN: +86%, C/RNN: +59%), möglicherweise weil einige Ausreißer nicht erkannt und verfolgt werden können. Obwohl das LKF den DEPM-Wert von ML-GP verringert, weist ML-GP die schlechtesten DET-Werte über alle Datensätze auf (durchschnittlicher DET: 163 m). Dies steht im Einklang mit der geringeren Reduzierung der RMSE-Werte durch ML-GP. Die anderen vier Methoden profitieren vom LKF und zeigen verbesserte DET-Werte (PDR: -724 m; RoNIN: -209 m; C/RNN: -43 m und Hybrid: -8 m).

Unabhängig davon, ob Methoden mit f_s oder mit f_t funktionieren, profitieren alle von einem LKF. Das LKF hat einen stärkeren Einfluss auf die größeren Fehler der f_s -Methoden, möglicherweise weil die f_t -Methoden (C/RNN und Hybrid) bereits die Zeit berücksichtigen, wenn sie ein Zustandsraummodell lernen, das die LKF ebenfalls bereitstellt, sodass die f_t -Methoden lediglich von einem leichten Glättungseffekt des LKF profitieren.

Zusammenfassung: Das Hinzufügen eines LKF (das schnell zu berechnen ist, siehe Abschnitt 5.7.3) zu einem der fünf Schätzer verbessert dessen Genauigkeit und wird dringend empfohlen. Je größer die Fehler (Varianz der Schätzungen) einer Methode sind, desto vorteilhafter ist das LKF. Da die Modelle sowieso trainiert werden müssen, kann auch das LKF problemlos zusätzlich trainiert werden, zumal in der Regel Validierungsdaten zur Verfügung stehen.

5.7.3 Rechenaufwand

Zur Erörterung des Rechenaufwands der fünf Schätzer für Datensatz V1 (Einzelheiten siehe Teil 3 von Tabelle 5.1) wird der Aufbau skizziert, das Bewertungskonzept vorgestellt und die Ergebnisse diskutiert.

Tabelle 5.3: Genauigkeit (höchste **fett**) der Geschwindigkeitsschätzer auf den ausgelassenen Probanden in m/s, Distanzfehler in mm.

Datensatz	Ausgelassenen Probanden			DET			DEPM			CFP ₉₅			Hybrid			CFP ₉₆			DET			DEPM								
	MAE	MSE	RONIN	MAE	MSE	CNN	MAE	MSE	CNN	MAE	MSE	DET	MAE	MSE	DET	MAE	MSE	DET	MAE	MSE	DET	MAE	MSE	DET						
Gehen	1.15	1.82	2.15	0.54	8.51	0.70	0.68	0.85	1.36	0.993	138.81	0.67	0.62	0.78	1.33	0.004	5.67	0.32	0.15	0.38	0.06	0.015	2.56	0.17	0.04	0.20	0.32	0.003	2.26	
Laufen	1.27	2.18	2.38	0.49	41.02	0.66	0.60	0.90	1.48	0.048	138.59	0.73	0.73	0.86	1.46	0.003	9.73	0.35	0.17	0.42	0.72	0.009	2.10	0.19	0.05	0.21	0.35	0.003	1.68	
Recken	1.35	2.50	2.54	0.92	87.74	0.83	0.76	0.95	0.97	1.62	0.937	140.97	0.80	0.87	0.94	1.00	0.001	44.91	0.39	0.21	0.46	0.77	0.006	3.52	0.21	0.05	0.25	0.38	0.002	2.33
Zufällig	1.38	2.56	1.60	2.56	0.37	105.69	0.97	0.93	0.95	0.93	0.954	153.55	0.83	0.82	0.93	1.58	0.026	75.11	0.41	0.23	0.48	0.79	0.009	21.46	0.21	0.06	0.25	0.43	0.004	11.47

5 Geschwindigkeitsschätzung

Tabelle 5.4: Rechenaufwand (beste **fett**) der Geschwindigkeitsschätzer.

Methode	t_{train} [h]		t_{live} [s]		t_{live} [Hz]	
	CPU	GPU	CPU	GPU	CPU	GPU
PDR	0.2	-	0.00073	-	1370	-
ML-GP	23.6	-	0.00203	-	493	-
RoNIN	11.2	8.1	0.01041	0.0147	96	68
C/RNN	6.9	4.9	0.00610	0.0069	164	145
Hybrid	15.6	10.5	0.01785	0.0277	56	36

Konfiguration

Für die Auswertungen für den Rechenaufwand werden handelsübliche Hardwarekomponenten verwendet. Es gibt zwei Konfigurationen: (1) einen Kern einer einzelnen CPU (Intel Core i7 4× 3.6 GHz), (2) diesen Computer, der mit einer einzelnen GPU (engl. Graphics Processing Unit, dt. Grafikprozessor) ausgestattet ist (Nvidia GTX1080Ti 11 GB). Für PDR, ML-GP und Hybrids LKF gibt es keine GPU-Implementierungen. Daher wird für PDR und ML-GP nur mit Konfiguration (1) gemessen. Für die anderen Systeme gibt es CPU-Messungen, Konfiguration (1), und auch Messungen, die mit Konfiguration (2) vorgenommen wurden. Hier koordiniert der CPU-Kern für RoNIN und C/RNN nur die GPU-Implementierungen. Für Hybrid läuft auch der LKF. Die Laufzeiten des vollständigen *Trainings* (alle Fenster/Merkmale) und der *Inferenz* (unten wird der Begriff *live* verwendet) für eine einzelne Eingabesequenz werden separat gemessen. Das Messverfahren kümmert sich nicht explizit um die Zeit, um die Daten vorzuverarbeiten oder sie von der Festplatte in den RAM oder in die GPU zu laden oder zu versenden (14 s), da die Trainingsphase deutlich länger dauert (~20 min). Bei den *live*-Messungen sind einige Vorverarbeitungszeiten in realen Anwendungen von Bedeutung: Sammeln von Daten von den Sensoren, (optionale) Synchronisation, (optionale) Stichprobewiederholung, (optionales) Interpolieren, (optionales) Schneiden von Fenstern, (optionales) Extrahieren von Merkmalen und schließlich die Vorhersagefunktion füttern. Diese Zeiten sind implizit in den Messungen enthalten. Auch hier wird die Zeit ($0.1 \text{ ms} = 14 \text{ s}/112.500 \text{ Fenster/Merkmale}$) vernachlässigt, die zum Laden oder Versenden von Daten erforderlich ist.

Bewertungskonzept

Für den Datensatz V1 listet Tabelle 5.4 t_{train} , für alle Fenster/Merkmale oder Segmente, und t_{live} , die durchschnittliche Zeit der Vorhersage der Geschwindigkeit für ein einzelnes Fenster/Merkmalbündel. Da Hybrid ein LKF verwendet, beträgt seine gesamte Trainingszeit:

$$t_{train} = t_{train}^{pure} + t_{train}^{LKF}, \quad (5.16)$$

wobei t_{train}^{pure} die Zeit von ResNet und BLSTM für den gesamten Trainingssatz (für alle 112.500 Fenster) abdeckt und t_{train}^{LKF} zeigt an, wie lange das trainierte Hybridmodell benötigt, um v' für den vollständigen Validierungssatz (11.250 Fenster) vorherzusagen plus wie lange es dauert, um das LKF auf diese Vorhersagen zu trainieren v' ($11.250 \times v'$). Die Live-Zeit des Hybrids beträgt:

$$t_{live} = t_{live}^{pure} + t_{live}^{LKF}, \quad (5.17)$$

wobei t_{live}^{pure} im Durchschnitt angibt, wie lange das trainierte pure Hybrid (ohne LKF) benötigt, um v' für jedes der 22.500 Fenster des Testdatensatzes vorherzusagen, und t_{live}^{LKF} mittelt, wie lange das trainierte LKF benötigt, um v'' vorherzusagen.

Diskussion

Im Folgenden werden die Trainingsergebnisse, gefolgt von den Live-Ergebnissen diskutiert. Für das Training sind die GPU-Zeiten um einen linearen Faktor schneller als die CPU-Zeiten, siehe Tabelle 5.4. Für die Live-Zeit sind die GPU-Zeiten langsamer als die CPU-Zeiten, da die GPU ihre Rechenleistung nicht in einem einzigen Fenster nutzen kann. Im Folgenden werden deshalb nur die CPU-Ergebnisse erläutert.

Das Training der PDR-Methode ist bei Weitem am schnellste ($t_{train}=0.2h$). Aufgrund ihrer Gradientenoptimierung dauern die anderen (datengetriebenen) Methoden mehr als $34\times$ länger. ML-GP trainiert am langsamsten ($t_{train}=23.6h$), was Liu et al. [277] bestätigen. RoNIN dauert lange ($t_{train}=11.2h$), da es auf ResNet aufbaut, das aufgrund seiner vielen Parameter [335] als rechenintensiv bekannt ist. Da C/RNN ein flaches CNN verwendet und daher deutlich weniger Parameter als ResNet verwendet, ist es schneller ($t_{train}=6.9h$). Für Hybrid verlangsamen die zusätzlichen BLSTM-Schichten ($\approx 3.4h$) und

5 Geschwindigkeitsschätzung

das LKF ($t_{train}^{LKF}=0.98h$) die Trainingszeiten ($t_{train}=15.6h$). Das LKF fügt dem reinen Hybrid-Training $t_{train}(pur)=14.6h$ für die CPU-Variante ($t_{train}(rein)=9.8h$ für die GPU-Variante) eine zusätzliche Zeit von $t_{train}(LKF)=0.98h$ für die CPU-Variante ($t_{train}(LKF)=0.71h$ für die GPU-Variante) hinzu. Vergleicht man Hybrid mit RoNIN, kann man ableiten, dass die BLSTM-Erweiterungen von Hybrid ungefähr 3.4h der gesamten Trainingszeit verbrauchen ($=15.6h - 0.98h - 11.2h$).

In Bezug auf die Live-Zeiten in Hz ist PDR wiederum die mit Abstand schnellste Methode ($t_{live}=1370$ Hz). Diesmal ist ML-GP etwa $3\times$ schneller als die anderen datengetriebenen Methoden. Sowohl für C/RNN als auch für Hybrid verlangsamen die BLSTM-Schichten die Vorhersage um etwa 40 Hz. Das LKF ist leicht-gewichtig und fügt nur etwa 0.002 s pro Fenster hinzu. Daher ist es eindeutig erschwinglich, jedem der Schätzer ein LKF hinzuzufügen.

Zusammenfassung. Das PDR Framework verarbeitet Daten so vor, dass ungefähr alle 2 Hz ($1\text{ s}/N_w/2=1/128/64=0.5$ s) ein neues Messbündel (Fenster, Merkmale oder Bruchteil eines Segments) verfügbar ist. Selbst auf einer CPU erreicht Hybrid seine genauesten Geschwindigkeitsschätzungen mit 56 Hz und ist daher schnell genug für die Sensor-signale in einer Vielzahl von Anwendungsfällen.

5.7.4 Generalisierbarkeit

Nachdem im Folgenden die Metriken eingeführt werden, mit denen die Generalisierbarkeit der Geschwindigkeitsschätzung bewertet wird, werden vier Experimente diskutiert: die unbekannten Daten der ausgelassenen Probanden, der Benutzer mit zwei speziellen Sensoren und die unbekannten *Geschwindigkeiten* zum Testen der Fähigkeit der Methoden, zwischen Geschwindigkeitsbereichen zu interpolieren und zu extrapoliieren (siehe die letzten beiden Zeilengruppen in Tabelle 5.1).

Metriken

Für die vier Experimente listet Tabelle 5.5 die sechs *Geschwindigkeitsfehlermetriken* auf, die aus der Genauigkeitsauswertung bekannt sind. Die Fehler werden auch als: CDF-, PDF- und DT-Diagramme in Abbildung 5.18 visualisiert. Vor den Experimenten *Ausgelassene Probanden* und *Unbekannte Geräte* wurden alle Methoden bestmöglich

5.7 Auswertung und Diskussion

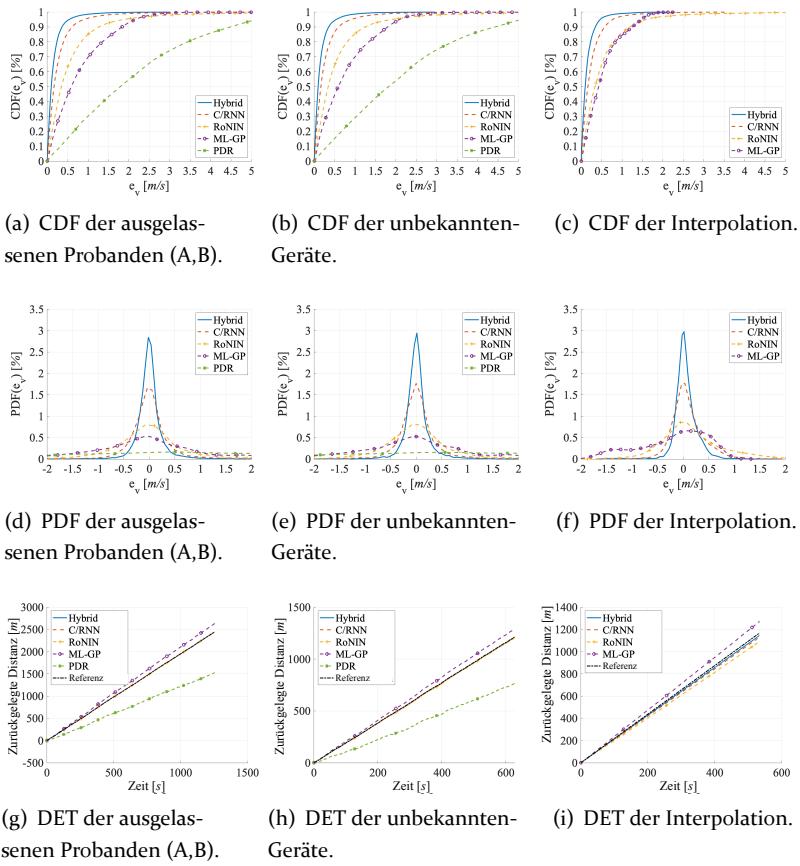


Abbildung 5.18: CDF, PDF, und DET der Generalisierbarkeitsauswertungen der Datensätze: ausgelassenen Prob. (A,B), unbekannte Geräte und Interpolation.

trainiert (PDR auf V₂, andere auf V₃). Die Methoden werden in ihrer ursprünglichen Form bewertet, nur Hybrid verwendet ein LKF. Vor den Experimenten *Interpolation* und *Extrapolation* müssen die vier datengetriebenen Methoden von Grund auf neu trainiert werden. Hierfür werden die V₃-Daten der 20 Probanden verwendet, ohne die ausgelassenen Probanden und den Benutzer der Spezialgeräte.

5 Geschwindigkeitsschätzung

Tabelle 5.5: Genauigkeit (höchste **fett**) der Generalisierbarkeit der Geschwindigkeitsschätzer in m/s; Distanzfehler in m.

Datensatz	MAE	MSE	PDR \bar{K}_d	CEP ₉₅	DET	MAE	MSE	RMSE	ML-GP	CEP ₉₅	DET	MAE	MSE	RMSE	RONIN	CEP ₉₅	DET	MAE	MSE	RMSE	C/RNN	CEP ₉₅	DET	MAE	MSE	RMSE	Hybrid	CEP ₉₅	DEPM	DET		
Ausgelausener Proband A	2.10	6.94	2.63	5.09	0.376	460.19	0.76	1.03	1.01	2.09	0.082	99.75	0.57	0.90	0.95	1.82	0.035	3.97	0.29	0.33	0.48	0.91	0.027	2.00	0.38	0.08	0.29	0.55	0.026	1.26		
Ausgelausener Proband B	2.06	6.64	2.58	5.03	0.362	442.94	0.78	1.07	1.03	2.10	0.071	86.58	0.57	0.90	0.95	1.85	0.061	6.87	0.30	0.23	0.48	0.91	0.022	3.38	0.38	0.08	0.29	0.54	0.023	2.07		
Ausgelausene Probanden A+B	2.10	6.95	2.64	5.08	0.374	914.55	0.78	1.08	1.04	2.14	0.077	188.13	0.58	0.92	0.96	1.85	0.029	4.43	0.30	0.23	0.48	0.93	0.024	2.33	0.39	0.09	0.29	0.56	0.024	1.54		
Unbekannte Geräte	2.32	7.21	2.89	5.43	0.56	506.44	0.83	1.21	1.16	2.38	0.077	101.42	0.73	0.98	1.06	2.12	0.093	12.44	0.32	0.35	0.52	1.02	0.032	5.21	0.21	0.12	0.32	0.67	0.028	3.44		
Interpolation	-	-	-	-	-	0.54	0.49	0.70	1.47	0.105	61.31	0.49	0.61	0.78	1.49	0.060	34.84	0.24	0.15	0.38	0.70	0.005	16.86	0.15	0.05	0.23	0.42	0.007	10.12			
Valid.	-	-	-	-	-	0.53	0.47	0.68	1.46	0.089	104.44	0.52	0.66	0.82	1.58	0.064	75.19	0.26	0.16	0.40	0.76	0.007	36.41	0.16	0.06	0.24	0.44	0.006	21.872			
Interpolation Test	-	-	-	-	-																											

Ausgelassene Probanden

Die beiden ausgelassenen Probanden bewegen sich ziemlich ähnlich: Proband A bewegt sich durchschnittlich mit 0.74 m/s (min.: 2.5 m/s, max.: 3.9 m/s, SD: 0.41 m/s). Proband B gab an, nicht so oft Sport zu machen wie Proband A, und bewegt sich mit durchschnittlich 0.66 m/s etwas langsamer (min.: 2.4 m/s, max.: 3.6 m/s, SD: 0.38 m/s). Die Schätzer für die ausgelassenen Probanden werden sowohl einzeln als auch in Kombination bewertet.

Die wichtigsten Erkenntnisse sind: (1) Alle Schätzer generalisieren gut (kleine Fehlerwerte), zumindest wenn sich unbekannte Benutzer mit Geschwindigkeiten bewegen, die von den Trainingsdaten abgedeckt werden. Sie generalisieren auch robust (ähnliche Ergebnisse in Zeilen für A, B und A + B in Tabelle 5.5). (2) Trotz der leicht unterschiedlichen Geschwindigkeit der beiden Probanden ist die Generalisierung ähnlich. Alle Schätzer machen nur kleine Fehler über alle Metriken hinweg. Die Fehler sind ähnlich wie die des Testsatzes von V₃, bzw. V₂ für PDR, in der Genauigkeitsauswertung, siehe Tabelle 5.2.

Ähnlich wie bei der Genauigkeitsauswertung generalisiert PDR am schlechtesten (größten Fehler), Hybrid am Besten (kleinsten Fehler). Alle Methoden behandeln Proband B über alle Fehlermetriken hinweg etwas besser als A, wahrscheinlich weil die Trainingsdaten die Geschwindigkeit von A nur teilweise abdecken. Die Ergebnisse für die Kombination der beiden Probanden A + B in Tabelle 5.5 ähneln denen für die einzelnen Ausgelassenen, aber die Schätzer schneiden über alle Fehlermetriken hinweg etwas schlechter ab (etwa -8%) als auf dem Testdatensatz von V₃ (bzw. V₂ für PDR). Die Ergebnisse sind immer noch beeindruckend, aber die leichte Abweichung kann durch fehlende Bewegungsmuster und Geschwindigkeiten in den Trainingsdaten verursacht werden. Das CDF-Diagramm in Abbildung 5.18(a) bestätigt diese Ergebnisse und zeigt, dass alle Methoden außer PDR in 70% aller Fälle Geschwindigkeiten mit Fehlern unter 1 m ergeben, die mit den Ergebnissen der Genauigkeitsauswertung vergleichbar sind. PDR kämpft mit den Probanden A + B, da PDR's Parametrisierung nicht die Kinetik der Bewegungen der Probanden abdeckt.

Hybrid und C/RNN sind die robustesten Schätzer, da sie die höchsten Spitzen mit den engsten Amplituden in Abbildung 5.18(d) zeigen.

5 Geschwindigkeitsschätzung

Ihre MSE- und RMSE-Werte sind jedoch schlechter (um -11%) als ihre Ergebnisse in der Genauigkeitsauswertung, siehe Tabelle 5.2. Dies lässt sich erklären mit den unbekannten und sehr individuellen Bewegungsmustern zwischen Fenstern, die sich im Laufe der Zeit ändern. Sogar die f_t -Methoden haben leichte Probleme, wenn sie mit den schnelleren Bewegungen von Proband A arbeiten, die nicht von den Trainingsdaten abgedeckt werden. PDR, ML-GP und RoNIN leiden noch mehr darunter.

Interessanterweise sind die DET-Werte für PDR (DET=914 m) und ML-GP (DET=188 m) fast die Summe der einzelnen DET-Werte für A und B, während die DEPM-Werte in allen drei Zeilen von Tabelle 5.5 gleich sind. Dies ist auch in Abbildung 5.8(g) zu sehen, in der PDR die Referenz unterschreitet und ML-GP die Referenz überschreitet. Es ist davon auszugehen, dass die Akkumulation und Unterschätzung der Geschwindigkeit durch PDR auf unentdeckte Schritte von Proband A zurückzuführen ist, da K_g möglicherweise nicht mit seinen/ihren schnelleren Bewegungsmustern übereinstimmt. Stattdessen können die Akkumulation von DET-Werten und das Überschießen der Geschwindigkeiten durch ML-GP darauf hinweisen, dass die Geschwindigkeitsgruppierungen der Trainingsdaten viele schnellere Geschwindigkeiten abdecken. Diese Neigung zu höheren Geschwindigkeiten kann die etwas höheren Fehlerwerte erklären. Stattdessen zeigen die DL-basierten Methoden (DET < 5 m im Durchschnitt) keine Akkumulation von DET-Werten. Sie alle nähern sich der Referenz, wahrscheinlich weil sie implizit lernen, mit solch komplexeren Merkmalsabhängigkeiten direkt aus den Daten umzugehen.

Zusammenfassung: ML-GP, RoNIN, C/RNN und Hybrid schneiden bei den beiden ausgelassenen Probanden in Bezug auf Genauigkeit und Robustheit gegenüber Ausreißern etwas schlechter ab als bei V3. C/RNN und Hybrid, die die f_t Merkmale nutzen, übertreffen ML-GP und RoNIN, die nur die f_s Merkmale verwenden. Da sie alle mit den unbekannten Bewegungsmustern und den höheren Geschwindigkeiten zu kämpfen haben, ist es empfehlenswert, alle möglichen Geschwindigkeitsgruppierungen in den Trainingsdaten abzudecken, aber zumindest die minimalen und die maximalen.

Unbekannte Geräte

In diesem Experiment bewegen sich zwei spezielle Sensoren (je ein Sensor wird in der linken und rechten Hosentasche eines unbekannten Benutzers platziert) mit einer Durchschnittsgeschwindigkeit von 2.6 m/s (min.: 1.3 m/s, max. : 3.5 m/s, SD: 0.28 m/s). Tabelle 5.5 enthält die Fehler in der zweiten Zeilengruppe. Die wichtigsten Erkenntnisse sind, dass alle Schätzer gut generalisieren (die Fehlerwerte sind etwas schlechter als der Testsatz von V₃, bzw. V₂ für PDR, der Genauigkeitsauswertung, siehe Tabelle 5.2), wenn sich bekannte Sensoren mit Geschwindigkeiten bewegen, die durch die Trainingsdaten abgedeckt werden. Aber sie haben etwas mehr Probleme mit unbekannten Sensoren als mit ausgelassenen Probanden.

Alle fünf Schätzer generalisieren etwas schlechter (PDR am schlechtesten, Hybrid am besten) über alle Fehlermetriken für die unbekannten Geräte als für die ausgelassenen Probanden. Und ihre Genauigkeit auf den unbekannten Geräten ist ebenfalls etwas schlechter als auf V₃ (bzw. V₂ für PDR), aber vergleichbar mit ihren Ergebnissen auf V₁, siehe Tabelle 5.2. Dies ist darauf zurückzuführen, dass die Methoden nicht nur mit unbekannten Bewegungsmustern, sondern auch mit unbekannten Skalierungen der Sensormessungen oder unbekanntem Kurz- und Langzeittrauschen zu kämpfen haben. Ähnlich wie bei den Ergebnissen der ausgelassenen Probanden in Abbildung 5.18(a) zeigt das CDF-Diagramm in Abbildung 5.18(b), dass alle Methoden außer PDR Geschwindigkeiten mit Fehlern liefern, die in 70% aller Fälle unter 1 m liegen. Dies liegt daran, dass die Sensoren eine andere Skalierung verwenden, die von den Trainingsdaten nicht abgedeckt wird.

Die Robustheit aller Methoden gegenüber Ausreißern ist für die unbekannten Geräte und für die ausgelassenen Probanden ähnlich, siehe Abbildungen 5.18(e) und 5.18(d). Das Entrauschen von Hybrid und C/RNN ist sowohl gegen Ausreißer als auch gegen Schwankungen in der Skalierung der Sensorwerte robuster. Alle Methoden zeigen schlechtere MSE- und RMSE-Werte (um -15%) als in der Genauigkeitsauswertung, siehe Tabelle 5.2. Da diese Ergebnisse mit den Ergebnissen für die ausgelassenen Probanden übereinstimmen, können sowohl die sehr individuellen Bewegungsmuster des unbekannten Benutzers, der die beiden Sensoren trägt, als auch die unbekannte Bewegungsempfindlichkeit der unbekannten Sensoren die höheren MSE- und

5 Geschwindigkeitsschätzung

RMSE-Fehler verursachen. Da unterschiedliche Skalierungen der Sensorwerte direkt in unterschiedliche Amplituden umgewandelt werden und die Methoden diese als unterschiedliche Geschwindigkeiten interpretieren, gibt es größere Ausreißer.

Die Abstandsfehler von PDR (DET=506 m) und ML-GP (DET=101 m) sind am höchsten, siehe Abbildung 5.18(h). Das stimmt auch mit ihren höheren MSE- und RMSE-Werte überein. Dies ist plausibel, da ihre schlechten MSE- und RMSE-Werte, die möglicherweise auf eine andere Skalierung der Sensordaten zurückzuführen sind, direkt in höhere DET-Werte umgewandelt werden. Wiederum überschreitet PDR und ML-GP die Referenz, siehe Abbildung 5.18(h). Der Grund dafür, dass PDR möglicherweise viele Schritte nicht erkennt, ist, dass die unterschiedliche Skalierung nicht mehr mit den Schwellenwerten des Spitzendetektors übereinstimmt, der auf anderen Sensoren mit einer anderen Skalierung trainiert wurde. Stattdessen werden die DL-basierten Methoden (DET < 12 m) nahezu perfekt auf die Referenz abgebildet. Ihre Abweichung vom Genauigkeitsbenchmark kann auch auf die unbekannte Abbildung von Bewegungsmustern sowie auf die unterschiedlichen Skalierungen und Messbereiche von Sensordaten zurückgeführt werden. Wiederum profitieren C/RNN und Hybrid (am besten) von der Verwendung von f_t , da sie von Zeit zu Zeit lernen können, Geschwindigkeiten aus verschiedenen Perspektiven zu interpretieren, und daher generell weniger Fehler verursachen.

Zusammenfassung: Alle Methoden arbeiten mit speziellen Sensoren nur geringfügig schlechter als mit unbekannten Probanden. Es gibt keine Hinweise darauf, dass sie mit anderen Sensoren mit noch stärkeren Variationen in der Skalierung noch gut funktionieren. Daher ist es empfehlenswert, den gesamten Bereich der Skalierungen in den Trainingsdaten abzudecken und auszubalancieren.

Interpolation

Die vier Schätzer werden mit Fenstern (oder Merkmalen) von langsamem ($v_{ref} \in [3.7] \text{ km/h}$) und schnellen ($v_{ref} \in (9,13] \text{ km/h}$) Geschwindigkeiten trainiert und werden anhand normaler Geschwindigkeiten ($v_{ref} \in [7.9] \text{ km/h}$) bewertet. Die Ergebnisse sind in den Zeilen am Ende der Tabelle angegeben 5.5. Zu beachten ist, dass PDR

nicht auf V_3 ausgewertet werden kann.

Die wichtigsten Erkenntnisse sind, dass alle vier datengetriebenen Methoden in der Lage sind, zwischen zwei unterschiedlichen Geschwindigkeitsbereichen zu interpolieren und Geschwindigkeits schätzungen mit einer ähnlichen oder sogar höheren Genauigkeit als beim Testsatz von V_3 bereitzustellen (siehe Tabelle 5.2). Die vier Schätzer verhalten sich bei der Validierung und den Testdatensätzen über alle Fehlermetriken hinweg ähnlich.

Während ML-GP (am schlechtesten) und RoNIN ähnlich abschneiden ($MAE=0.5$, $CEP_{95}=1.5$ [m/s]), schneiden C/RNN und Hybrid (am besten) etwas besser ab ($MAE=0.26$, $CEP_{95}=0.76$ gegenüber $MAE=0.16$, $CEP_{95}=0.44$ [m/s]). Abbildung 5.18(c) zeigt, dass in 70% aller Fälle alle vier Methoden Fehler unter 0.7 m/s aufweisen. Die Grafik zeigt auch, dass ML-GP bei Wahrscheinlichkeiten von über 90% RoNIN deutlich übertrifft (kleinere Fehler liefert). Eine Erklärung ist, dass ML-GP lernt, seine internen Verteilungen so auszurichten, dass sie den unbekannten Geschwindigkeitsbereich sehr gut abdecken, während RoNIN seinen Lernmechanismus für räumliche Fingerabdrücke, f_s , nicht so stark nutzen kann. Die gute Leistung von C/RNN und Hybrid steht im Einklang mit allen anderen Ergebnissen, bei denen die Nutzung von f_t fast immer die Verwendung von nur f_s übertrifft. Beide Schätzer profitieren von der Verwendung eines optimalen Trainingssatzes, der auch die Grenzgeschwindigkeiten abdeckt. Die vier Methoden interpolieren ungefähr +3% besser als sie in der Genauigkeitsauswertung schätzen (Tabelle 5.2). Hier hilft es, dass der unbekannte Geschwindigkeitsbereich vergleichsweise klein ist, sodass nur Interpolationen mit kurzer Reichweite erforderlich sind.

Die Robustheit aller Methoden ähnelt der Genauigkeitsauswertung, da sich die Trainings- und Testdaten nicht ändern (dieselben 20 Benutzer), obwohl hier andere Teile weggelassen wurden.

Sowohl die DEPM- als auch die DET-Werte skalieren linear mit den MSE- und der RMSE-Werten. Wiederum ist ML-GP die einzige Methode, die die Referenz überschreitet, siehe Abbildung 5.18(i), da Geschwindigkeitsbereiche innerhalb jeder Geschwindigkeitsgruppierung leicht positiv unausgeglichen sind, m.a.W. es existieren etwas mehr schnelle als langsame Geschwindigkeiten. Während RoNIN mit

5 Geschwindigkeitsschätzung

solchen Ungleichgewichten besser umzugehen scheint als ML-GP, unterschreitet es die Referenz und führt zu geringeren Fehlern über alle Metriken hinweg. C/RNN und Hybrid interpolieren etwas besser ($\text{DET}=36 \text{ m}$ gegenüber $\text{DET}=21 \text{ m}$) als die Methoden, die nur f_s verfolgen. Jedoch ist es schwierig zu begründen, wie sich ihre Zeit- und Kontextsensitivität auf die Interpolation auswirken. Vermutlich zwingen die f_t -Merkmale die C/RNN- und Hybrid-Modelle dazu, sich implizit mehr auf die Konnektivität und die gegenseitigen Abhängigkeiten von Geschwindigkeiten zu konzentrieren und führen so zu besseren Ergebnissen.

Zusammenfassung: Alle vier Methoden eignen sich sehr gut für Nahbereichsinterpolationsaufgaben. Obwohl diese Erkenntnisse vielversprechend sind, da weniger Trainingsdaten benötigt werden, wird Expertenwissen über die Skalierungseffekte und Geschwindigkeitsbereiche in den Daten benötigt. Zukünftige Forschung ist erforderlich, um die Interpolation über größere Entfernung, bzw. größere Abstände zwischen Geschwindigkeitsgruppierungen, besser zu verstehen.

Extrapolation der Geschwindigkeit

In diesem Experiment werden die vier Schätzer mit Fenstern (oder Merkmalen) langsamer bis normaler Geschwindigkeiten ($v_{ref} \in [3..11] \text{ km/h}$) trainiert und ihre Leistung mit schnellen Geschwindigkeiten ($v_{ref} \in [11..13] \text{ km/h}$) bewertet. Zu beachten ist, dass PDR mit V₃ nicht ausgewertet werden kann. Es gibt weder Zahlen in Tabelle 5.5 noch Diagramme, da die datengetriebenen Methoden nicht extrapoliieren können, um plausible Ergebnisse zu liefern.

Alle Schätzer zeigen ähnliche und hohe Fehler ($5\text{-}10 \times$ höher im Vergleich zur Genauigkeitsauswertung), die zufällig schwanken, sowohl bei den Validierungs- als auch bei den Testdaten. Es gibt keine klare Beziehung zwischen der Eingabe und der Extrapolation. Da datengetriebene Methoden lernen, die Validierungsdaten anzupassen, können sie nur dann gut mit den Testdaten arbeiten, wenn beide Datensätze aus demselben Datenpool stammen. Da in diesem Experiment alle Methoden sowohl für die Validierung als auch für die Testsätze fehlschlagen, bedeutet dies, dass sie wirklich nicht extrapoliieren können. Je näher die Geschwindigkeiten an den Trainingsdaten liegen, d. h. näher an 11 km/h, desto höher wird die Genauigkeit. Sie nimmt ab, wenn mit

Daten getestet wird, die die von den Trainingsdaten abgedeckten Maximgeschwindigkeiten überschreiten. Daher ist die Verwendung der datengetriebenen Methoden für die Extrapolation nicht empfehlenswert, zumindest nicht für Geschwindigkeiten, die nicht in den Trainingsdaten enthalten sind.

Extrapolation zufälliger Bewegungsmuster

Da keine der Methoden bei speziell ausgewählten Geschwindigkeiten ausreichend gut extrapoliert, wurde getestet, ob und wie genau sie zufällige Bewegungsmuster extrapolieren können. Zu diesem Zweck wurden die Methoden nur mit Daten von Gehen, Joggen und Rennen trainiert und die Aktivität *Zufällig* wurde ausgelassen. Da die Ergebnisse keine Anomalien zeigen, wird nur der zurückgelegte Entfernungsfehler diskutiert.

Die PDR-Methode überschreitet die Referenz erneut signifikant um 759 m, während ML-GP (-200 m), RoNIN (-142 m), C/RNN (-87 m) und Hybrid (-16 m) die Referenz unterschreiten. Eine mögliche Erklärung ist, dass die Daten von Gehen, Joggen und Rennen bereits Bewegungsmuster von Benutzern enthalten, die sich mit unterschiedlichen Bewegungen mit abrupten Richtungsänderungen bewegten und daher alle Methoden implizit Merkmale der Aktivität *Zufällig* lernen. Daher ist die Verwendung der datengetriebenen Methoden zur Extrapolation nur empfehlenswert, wenn die Bewegungsmuster in den Trainingsdaten enthalten sind. Kapitel 7 berichtet vergleichbare Erkenntnisse.

5.7.5 Effekte der Hyperparameter

Typischerweise integrieren Geschwindigkeitsschätzer kontinuierlich Fehler über die Zeit. Dieser Zyklus wird von der Hybridarchitektur unterbrochen, indem quasi unabhängige Fenster über das Signal [189, 70] geschoben werden. Die Fenstergröße N_w und die Fensterüberlappung sind jedoch Parameter, die das Modell sowohl hinsichtlich der Beschleunigung als auch der Winkelrate beeinflussen. Die Fenstergröße beeinflusst auch die Lernrate. Brownlee et al. [55] behaupten, dass die maximale Eingabesequenz im Allgemeinen $N_w=400$ (bei 100 Hz) beträgt, damit LSTM-Zellen ausfallsicher arbeiten. Das Hybridmodell scheint Langzeitabhängigkeiten f_t von Messungen in Fenstern bereits aus 128 Messpunkten (respektive 1.28 s) zu lernen und lieferte die

5 Geschwindigkeitsschätzung

höchste Genauigkeit aller getesteter Methoden. Der Vollständigkeit halber werden hier die Leistungsschwankungen skizziert, die durch die Änderung von N_w verursacht werden.

Während die *Inferenzzeiten* t_{live} nicht von N_w und der Überlappung beeinflusst werden ($SD \pm 0.002$ s), sind Effekte für die *Trainingszeiten* t_{train} ersichtlich. Eine Fenstergröße $N_w=512 > 128$ führt zu redundanten Informationen, verlangsamt den Trainingsprozess für datengetriebene Methoden ($t_{train} + 7.3h$ bei 100 Hz) und senkt den wertvollen Informationsgehalt (f_s) für ML-GP (da die Merkmale verschwimmen) und RoNIN (Testsatz von V1: RMSE 1.01 m/s und 1.16 m/s), wirkt sich auch negativ auf die Leistung von C/RNN und Hybrid aus (Testsatz von V1: RMSE 0.74 m/s und 0.61 m/s) und erhöht den Rechenaufwand, selbst für PDR-Methoden. $N_w=64 < 128$ erhöht die Rechenleistung ($t_{train} - 4.1h$ bei 100 Hz), liefert jedoch bemerkenswert weniger genaue Schätzungen für den Testsatz von V1 (RMSE [m/s]: ML-GP 1.94, RoNIN 1.92, C/RNN 1.89, Hybrid 1.78, vergleiche Zeile V1 der Tabelle 5.2). Dennoch kann auch ein niedriges N_w immer noch eine Option für die Mobiltelefon-basierte Lokalisierung sein, wenn ein geringerer Energieverbrauch wichtig ist.

Auswirkung der Dimension der Eingabesequenz auf die Genauigkeit der Schätzer: Experimente haben gezeigt, dass alle vier datengetriebenen Methoden sowohl bei 1D- als auch bei 2D-SMVs ähnlich funktionieren (RMSE SD ≤ 0.06 m/s), 1D-SMV jedoch Rechenkosten spart ($t_{train} - 2.9h$ bei 100 Hz). Es ist daher ratsam, die Verwendung von Mehrkanalsignalen gegenüber Einkanalsignalen immer anwendungsspezifisch abzuwägen: Genauigkeitszuwachs gegenüber Robustheit und Genauigkeit gegenüber Rechenaufwand.

5.8 Zusammenfassung

Modernste IMU-Ansätze zur Fußgänger Koppelnavigation aus dem Stand der Technik können bereits in einigen alltäglichen Situationen die Geschwindigkeit menschlicher Bewegungen abschätzen. Sie alle weisen jedoch Schätzfehler auf, wenn die Bewegung dynamisch ist, z. B. bei VR- oder Sportanwendungen, wenn abrupte und unvorhersehbare Bewegungsänderungen auftreten oder wenn IMU-Sensoren nicht starr am Körper angebracht sind. Dieses Kapitel hat einen neuartigen

Hybridschätzer präsentiert. Die Architektur von Hybrid kombiniert tiefes Lernen mit einem Bayesschen Filter, um die Geschwindigkeit einer Person in rotationsinvarianten Signalströmen zu schätzen. Die Kombination aus faltenden neuronalen Netzen (ResNet-i8) und BLSTM-Netzen extrahiert räumliche Merkmale aus den Sensorsignalen und verfolgt deren zeitliche Beziehungen, um Geschwindigkeiten zu schätzen. Ein LKF optimiert diese Schätzungen weiter.

Ein hochgenaues Referenzsystem wird verwendet, um Geschwindigkeitsdaten von Athleten zu sammeln und aus den Daten auf überwachte Weise zu lernen, wie die Geschwindigkeit einer gehenden Person, eines Joggers oder eines Sprinters genau geschätzt werden kann. Die resultierenden Geschwindigkeitsschätzungen und die aktuelle Orientierung (siehe Kapitel 6) des Athleten können dann kombiniert werden, um seine aktuelle Pose zu bestimmen (siehe Kapitel 7). Das Kapitel vergleicht die Methoden des Stands der Technik wie PDR, ML-GP (Gaußsche Prozesse, die mit handgefertigten Merkmalen arbeiten), RoNIN (ResNet), C/RNN (CNN + BLSTM) und Hybrid auch quantitativ. PDR liefert die schlechteste Genauigkeit, selbst wenn subjektaktivitätsspezifische Kalibrierungs- oder Vorklassifizierungsmechanismen verwendet werden. Während der Rechenaufwand für PDR am niedrigsten ist, hat es Schwierigkeiten, zu generalisieren. Die vier datengetriebenen Ansätze sind toleranter gegenüber der Ausrichtung und Platzierung des Sensors als PDR und lernen, einen Geschwindigkeitsvektor in dynamischen Szenarien auf eine genaue Referenzgeschwindigkeit abzubilden. ML-GP zeigt eine mittlere Genauigkeit, höhere Kosten aufgrund handgefertigter Funktionen und instabile Generalisierbarkeitseffekte. Stattdessen weist RoNIN eine höhere Genauigkeit, geringere Kosten und eine etwas bessere und stabilere Generalisierbarkeit auf, da Ende-zu-Ende direkt aus Daten gelernt wird. Im Gegensatz dazu liefern C/RNN und Hybrid aufgrund des BLSTM die höchste Genauigkeit bei langsamen Inferenzzeiten. Die Bearbeitungszeit ist aufgrund der lang anhaltenden Rastersuche und des längeren Trainings ebenfalls hoch. Hybrid erreicht einen Geschwindigkeitsfehler von ≤ 0.16 m/s und einen Fehler über die zurückgelegte Strecke von ≤ 3 m/km. Ein hinzugefügtes LKF verbessert die Ergebnisse um etwa 10%. Hybrid generalisiert auch besser als alle anderen Methoden. BLSTM profitiert am besten von der Extraktion lokaler Merkmale f_s und die Behandlung sequentieller langfristiger kontextbezogener Abhängigkeiten f_t . Alle Methoden leiden jedoch

5 Geschwindigkeitsschätzung

unter der Anhäufung relativer Fehler. Selbst die datengetriebenen Methoden akkumulieren einen merklichen Fehler von 24 m/km in einem unbekannten Testdatensatz.

Hybrid ist auf realistischere Probleme anwendbar als frühere Ansätze, da (a) seine Leistung stabil bleibt, selbst wenn die eingegebenen IMU-Messungen von unbekannten Sensoren oder unbekannten Benutzern stammen und wenn sich die Trainings- und Testdaten nicht in derselben Domäne befinden; (b) Hybrid kann mit herausfordernden Bewegungen und unterschiedlichen Gehgewohnheiten unbekannter Benutzer effizient umgehen, und (c) Hybrid ist robust gegen Ausreißer, hauptsächlich aufgrund des Trainings, das alle Bewegungen und Geschwindigkeiten in einem Modell kombiniert. Natürlich kann es immer noch reale Situationen geben, die Hybrid noch nicht abdeckt. Ein initiales Training mit synthetischen Daten und ein Nachtrainieren mit realen Daten könnten dazu beitragen, die Grenzen der Generalisierbarkeit künstlicher Intelligenz zu überschreiten.

6 Orientierungsschätzung

Einordnung

Abbildung 6.1 zeigt die Zuordnung der Orientierungsschätzung in die übergeordneten Verarbeitungskette der Dissertation. Die Orientierungsschätzung liefert kalibrierte Orientierungen aus Richtungsvektoren der Positionsschätzungen eines Funklokalisierungssystems (Kapitel 4) und verrauschten Trägheitsmessungen von einem am Kopf platzierten Trägheitssensor. Dieses Kapitel baut auf bereits veröffentlichten Ergebnissen [125, 126, 124, 122] auf. Der Abschnitt 6.3 basiert auf den Ergebnissen zweier veröffentlichter Studien zur Fehleranalyse moderner kommerzieller Trägheitssensoren in Anwendungen der virtuellen Realität [125, 122]. Die Methode basiert auf zwei aufeinanderfolgenden veröffentlichten Studien, die die Anwendbarkeit der Methode in der virtuellen Realität untersuchen, wenn es unterschiedliche Rotationssysteme und dynamische Bewegungen gibt [124, 126]. Im Rahmen dieser Forschungsarbeiten wurden drei Patente eingereicht und teilweise veröffentlicht [119, 123, 121].

Kurzzusammenfassung

Mit Bewegungsfreiheit und Mehrbenutzerfunktionen besteht die Notwendigkeit, virtuelle Realität (engl. Virtual Reality, VR) für große Räume zu ermöglichen. Die Kosten für eine genaue kamerabasierte Lokalisierung des Stands der Technik steigen jedoch mit der Größe

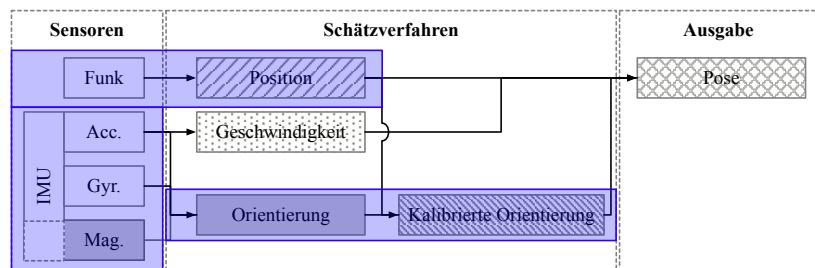


Abbildung 6.1: Einordnung der Orientierungsschätzung in die Verarbeitungskette (blaue Kästchen umschließen die beteiligten Komponenten).

6 Orientierungsschätzung

der Räumlichkeiten und der Anzahl der Benutzer. Verfahren, die keine vollständige Pose eines Benutzer verfolgen müssen, sogenannte Keine-Pose-Verfolgung (engl. No-Pose-Tracking, NP-Tracking) sind billiger, können jedoch die Gier-Orientierung des Kopfes des Benutzers auf lange Sicht nicht genau und zuverlässig abschätzen. SLAM-basierte von Innen nach Außen Verfahren (engl. inside-out) sind für interaktive Mehrbenutzer-VR nicht zuverlässig genug.

Die hier vorgeschlagene neuartige Orientierungsschätzung des Gierwinkels kombiniert einen einzelnen Trägheitssensor, der exemplarisch für eine VR-Anwendung am Kopf des Benutzers angebracht wird, um die unterschiedlichen Bewegungsausrichtungen von Körper und Kopf zusammenzuführen, mit der Positionsschätzung des Körpers bzw. Kopfes aus Kapitel 4. Das Verfahren nutzt aus, dass Menschen dazu neigen, in ihre Blickrichtung zu gehen, und optional, dass VR-Nutzer einen gewissen Orientierungsfehler tolerieren. Die Methode klassifiziert die Kopf- und Körperbewegung und schätzt den Orientierungsfehler, um eine kostengünstige, langzeitstabile Kopforientierung bei der NP-Verfolgung auf beliebigen Flächen (hier exemplarisch 100 m × 100 m) zu ermöglichen. Das Verfahren kann folglich dazu genutzt werden, Bewegungsmuster der menschlichen Bewegung, hier des Kopfes, zu denen die absolute Weltausrichtung des Sensors bekannt ist, bspw. die Ausrichtung des Ganges über die ermittelte Lauf- bzw. Positionsrichtung zu detektieren, um diese initial und nachhaltig zu kalibrieren, d. h. etwaige auflaufende Fehler zu korrigieren. Zu beachten ist, dass dieses Verfahren auch die Fehler bekannter Orientierungsschätzer, z. B. Madgwick [287], KF [239] und Oculus [253] korrigieren kann, auch wenn die Trägheits- und Funksensoren nicht am Kopf angebracht sind. Da die vorgeschlagene Klassifizierung auch auf andere Bewegungsmuster, andere Körperteile übertragen werden kann [478]. Zur besseren Übersicht werden in diesem Kapitel nur die dynamischen Beziehungen zwischen Kopf- und Körperrotationen am Beispiel einer VR-Anwendung untersucht und Ergebnisse berichtet.

Die Evaluation zeigt, dass das Verfahren die Orientierung in einer realen VR-Anwendung sehr genau und zuverlässig (absoluter Fehler in 98% aller Klassifikationsfälle <5°) schätzen kann. Kapitel 7 kalibriert die Posenschätzung mit den Orientierungen.

Das Kapitel ist folgendermaßen strukturiert: Abschnitt 6.1 führt die Orientierungsschätzung ein. Abschnitt 6.2 beschreibt die Herausforderungen der Orientierungsschätzung formal. Abschnitt 6.3 analysiert die Entstehung von Fehlern der Orientierungsschätzung. Abschnitt 6.4 diskutiert vorausgegangene Arbeiten aus dem Stand der Technik. Aus der Diskussion leitet Abschnitt 6.5 die Beiträge der hier erforschten Orientierungsschätzung ab. Abschnitt 6.6 beschreibt die Verarbeitungskette der Methode und deren Architektur. Abschnitt 6.7 beschreibt das Experiment, das zur Auswertung der Methode verwendet wird. Abschnitt 6.8 präsentiert die Ergebnisse der Auswertung. Abschnitt 6.9 diskutiert die Ergebnisse und Limitierungen der Methode. Abschnitt 6.10 fasst die wesentlichen Inhalte des Kapitels zusammen und gibt einen Ausblick.

6.1 Einführung

VR treibt Innovationen bei Anwendungen für Themenparks, Museen, Architektur, Schulungen, Simulationen usw. voran. Sie alle können von der Interaktion mit mehreren Benutzern, von Bereichen über $20m \times 20m$ und von natürlichen Bewegungen ohne Simulatorkrankheit profitieren, aber die aktuellen SLAM-basierten Verfahren zur Posenschätzung erreichen eine präzise und driftfreie Verfolgung nur unter einschränkenden Bedingungen, z. B. kleine Räume, statische Szenen, m.a.W. keine oder nur wenige sich bewegende Objekte und homogener Ausleuchtung [57, 133]. Darüber hinaus verwenden die heutigen hochmodernen VR-Systeme für kleine Bereiche unter $20m \times 20m$ eine kamerabasierte Bewegungsverfolgung von hoher Qualität. Mit der Größe des Bereichs ist eine höhere Kameraauflösung nötig, somit steigen die Kosten und die Genauigkeit nimmt stark ab. Der Mehrbenutzerbetrieb benötigt mehr Kameras, um Verdeckungen zu vermeiden.

Konzeptionell sind Keine-Pose-Verfolgungssysteme (NP), die nur einzelne Positionen anstelle der vollständigen Pose (Position und Orientierung) verfolgen, billiger und können mit größeren Verfolgungsbereichen und mehr Benutzern arbeiten. Im Gegensatz zur kamerabasierten Verfolgung der vollständigen Pose bieten sie nur Positionen, die aufgrund mangelnder Genauigkeit und fehlender Absolutorientierung nicht genutzt werden können, um die Ganzkörperpose (Rumpf und Kopf) abzuleiten. Die absolute Koporientierung muss

6 Orientierungsschätzung

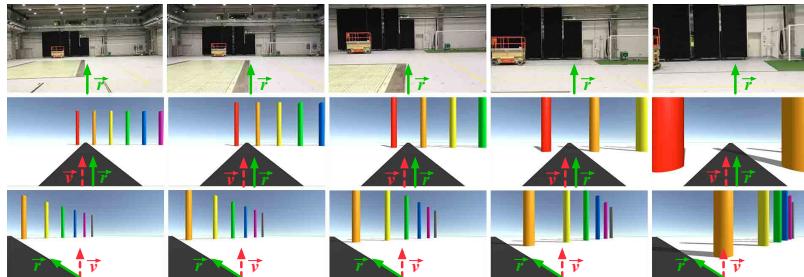


Abbildung 6.2: Reale (obere) und virtuelle Welt (mittlere und untere Reihe). Reale Kopf- / Blickrichtung \vec{r} , virtuelle Blickrichtung \vec{v} . Zweite Reihe: ohne Orientierungsfehler. Untere Reihe: Orientierungsfehler von 45° nach rechts (Fehler um die vertikale Körperachse, d. h. Gierachse). Obwohl der Benutzer in der dritten Reihe immer noch in die gleiche reale Blickrichtung \vec{r} blickt und geht wie in der oberen und mittleren Reihe, wird seine virtuelle Ansicht um 45° nach rechts verschoben. Anstatt virtuell geradeaus in Richtung der blauen Säule zu gehen, geht er seitwärts und nähert sich der orangefarbenen Säule in der virtuellen Welt. Der Benutzer kann entweder den Kopf um 45° drehen oder zur Seite gehen, um \vec{r} auf \vec{v} oder beides einzustellen. Wenn die virtuelle Ansicht \vec{v} von der realen Ansicht \vec{r} abweicht, nehmen die Auswirkungen der Simulatorkrankheit mit dem Versatz zwischen \vec{v} und \vec{r} zu und zerstören das VR Erlebnis.

daher separat geschätzt werden.¹

Gegenwärtige kostengünstige Kopf-montierte Bildschirmeinheiten (engl. Head-Mounted-Display, HMD), mit ihren Trägheitsmessseinheiten (IMU) wie Beschleunigungsmessern, Gyroskopen und Magnetometern, können verwendet werden, um die Kopforientierung auch bei geringerer Latenz und besserem Eintauchen als bei kamera-basierten Systemen zu schätzen. Obwohl es möglich ist, die richtige absolute Nick- und Rollorientierung mit Beschleunigungsmesser- und Gyroskopsensoren zu schätzen, ist in der Praxis eine IMU-basierte Schätzung des Gier-Winkels, d. h. des Drehwinkels um die vertikale Körperachse sehr ungenau. Erstens sind Magnetometer in vielen Innen- und Magnetumgebungen unzuverlässig und liefern eine falsche absolute Gier-Orientierung [44]. Zweitens führt die auf relativen IMU-Daten basierende Koppelnavigation zu einem Sensorfehler und (nach

¹Es ist zu beachten, dass der Kopf in eine andere Richtung als die Bewegungsrichtung ausgerichtet sein kann.

einer Weile) zu einer falschen Gier-Orientierungsschätzung [142]. Drittens liefern Filter nach dem Stand der Technik keine zuverlässigen Bewegungsrichtungsschätzungen für verrauschte, kostengünstige Sensoren, da sie entweder rauschfreie genaue Sensormodelle oder militärische Sensoren benötigen [170]. Eine naive Orientierungs-/ Versatzberechnung summiert verrauschte Beschleunigungen zweimal und führt zu schnell zunehmenden Fehlkalkulationen. Viertens machen es Kopfdrehungen neben dem Sensorrauschen schwierig, die linearen und Schwerkraftkomponenten der Beschleunigung zuverlässig zu trennen [141]. Fünftens ist es praktisch unmöglich, die Parameter des KF, d. h. die Kalman-Verstärkung (engl. Kalman gain) oder das Prozess- und Messrauschen so einzustellen, dass sie die Dynamik von Sensordrift, die Bewegung des menschlichen Kopfes und die daraus resultierenden Nichtlinearitäten korrekt beschreiben [526]. Sechstens können selbst Methoden, die ein KF nach dem Stand der Technik auf der Grundlage bekannter Sensorzustände stabilisieren, z. B. schuhmontierte Sensoren [142], kontextbasierte Handsensoren oder Sensoren am Rumpf eines Menschen (z. B. der Hosentasche), den absoluten Gier-Winkel vom Kopf nicht abschätzen [413].

Eine fehlerhafte Orientierungsschätzung führt zu einer Nichtübereinstimmung der realen Welt und der VR-Anzeige. Die obere Reihe in Abbildung 6.2 zeigt die Ansicht eines Benutzers, der mit seinem Kopf in Bewegungsrichtung geradeaus geht. In der virtuellen Realität (mittlere Reihe von Abbildung 6.2) sollte diese Bewegung zwischen der roten und der orangefarbenen Säule enden. Unter Drift (die untere Reihe zeigt einen Versatz des Gier-Winkels von 45°) führt dieselbe Bewegung jedoch zu einer Verschiebung von rechts nach links da eine falsche Kopf- bzw. Blickrichtung \vec{v} zum darstellen (engl. rendern) der VR-Bilder verwendet wird. Für den Benutzer passt dann die Bewegungsrichtung nicht zur VR-Ansicht. Dies führt zu Simulatorkrankheit, die ähnliche Auswirkungen wie die Reisekrankheit hat.

Die Schlüsselidee des Verfahrens zur Orientierungsschätzung, das in dieser Dissertation vorgestellt wird, besteht darin, die ungenaue Positionsverfolgung (horizontaler Fehler von ± 0.20 m, siehe Kapitel 4) mit relativen IMU-Daten zu kombinieren, um eine langzeitstabile Gier-Orientierung zu erreichen, während sich der Benutzer natürlich bewegt und dabei auch seinen/ihren Kopf frei drehen darf. Unter der Annahme, dass Menschen meist in ihre Blickrichtung gehen, werden Merk-

6 Orientierungsschätzung

male aus Sensorsignalen extrahiert, die Beziehung zwischen realer Bewegungsrichtung und realer Kopforientierung (mit überwachtem maschinellem Lernen) klassifiziert und diese mit absoluten Positionsinformationen kombiniert. Dieses Verfahren ergibt eine Schätzung der absoluten Gier-Orientierung, mit der ein Versatz zwischen realer und geschätzter Orientierung an die virtuelle Sicht eines Benutzers angepasst wird. Das Verfahren kann mit jedem Verfahren zur Orientierungsschätzung kombiniert und die Schätzfehler kalibriert werden. Im Rahmen der Dissertation wurden die Orientierungsschätzungen der bekannten Methode von Lavalle et al. [253] korrigiert. Natürlich können auch Bewegungsmuster der Hände, Füße oder des Rumpfes in Bezug auf die Bewegungsrichtung unter Verwendung des hier vorgeschlagenen Verfahrens klassifiziert werden. Bspw. wurden in zwei Vorstudien [53, 207] die Bewegungsmuster der Hände verwendet, um die Ausrichtung des Körpers genau zu bestimmen (96,3%).

6.2 Problembeschreibung

Dieser Abschnitt beschreibt die Herausforderungen der trägheitsbasierten Orientierungsschätzung. Zunächst wird das Problem motiviert (Abschnitt 6.2.1). Dann wird es formalisiert (Abschnitt 6.2.2). Anschließend wird eine Lösung vorgeschlagen (Abschnitt 6.2.3).

6.2.1 Hintergrund

Relative Bewegungsschätzungssensoren verursachen auf lange Sicht aufgrund der Sensordrift zwangsläufig eine falsche Orientierung. Oft sind jedoch relative Sensoren die einzige Option, da absolute Sensoren (wie Magnetometer) in der Praxis nicht zuverlässig funktionieren. Während Beschleunigungsmesser eine absolute Schätzung von Nick und Roll ermöglichen, können sowohl Gyroskop als auch Beschleunigungsmesser den Gier-Winkel nicht absolut schätzen. Neuere SLAM-Technologien wie Tango [474] (implementieren einen Hardware-Tiefensor mit einem maximalen Bereich von 5 m) und ARKit [57] (die Augmented Reality-Programmierschnittstelle für Apple-Geräte implementiert einen Merkmalsabgleich-(engl. Feature-matching-)Algorithmus, der von den Szenenbedingungen abhängt) funktionieren ebenfalls nicht, da sie fehlschlagen, wenn mehrere Benutzer die optischen Sensoren verdecken oder wenn der Verfolgungsbereich wenig geometrische oder strukturelle Eigenschaften aufweist,

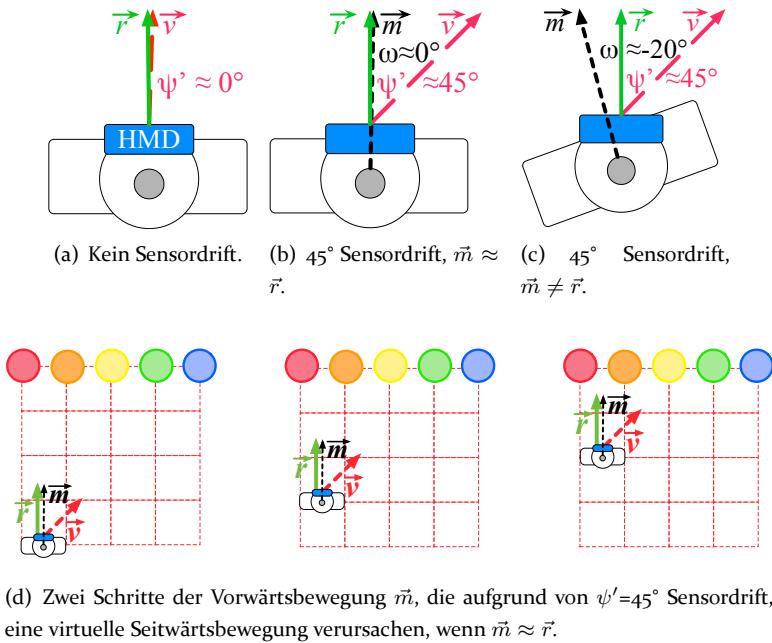


Abbildung 6.3: Sensordrift und seine Auswirkung. Vogelperspektive auf einen Benutzer, der ein HMD trägt. Versatz ψ' zwischen der realen Kopfrichtung \vec{r} und der virtuellen Richtung \vec{v} ; Versatz ω zwischen \vec{r} und Bewegungsrichtung \vec{m} ; farbige Säulen aus Abbildung 6.2 in (c).

z. B. bei schlechten Lichtverhältnissen. Da VR-Lokalisierungssysteme physische Hindernisse auf großen Flächen vermeiden, um Benutzer vor Kollisionen zu schützen, ist ihre geometrische und strukturelle Komplexität zu gering, um genaue Posen zu liefern [133].

6.2.2 Formalisierung

Abbildung 6.3 zeigt verschiedene Driftszenarien aus der Vogelperspektive auf einen Benutzer mit einem HMD (in blau). In Abbildung 6.3(a) gibt es fast keine Drift ($\psi' \approx 0^\circ$). Der reale Kopf-Ausrichtung \vec{r} des Benutzers befindet sich in der Nähe seiner virtuellen Kopf-Ausrichtung \vec{v} d. h. der Winkel zwischen \vec{r} und \vec{v} ist Null. Bewe-

6 Orientierungsschätzung

gungen fühlen sich natürlich an, wenn das VR-Bild mit der richtigen Kopf-Körper-Pose, d. h. mit einem korrekten absoluten Gier-Winkel visualisiert wird. In Abbildung 6.3(b) unterscheidet sich der Winkel zwischen \vec{v} und \vec{r} um $\psi' \approx 45^\circ$. Wenn sich der Benutzer in Richtung \vec{m} bewegt, erkennt er/sie dies als unnatürliche Übersetzung des visualisierten Bildes in Richtung \vec{v} . Abbildung 6.3(d) ist eine verkleinerte Ansicht dieser Situation mit dem Benutzer unten und den farbigen Säulen aus Abbildung 6.2 am oberen Ende des Rasters. Der Benutzer, der sich in zwei Schritten in Richtung \vec{m} vorwärts bewegt, versucht, auf die blaue Säule zuzugehen, da die VR-Anzeige anzeigt, dass sie geradeaus ist. In Wirklichkeit ist der Kopf/Körper des Benutzers jedoch in Richtung \vec{m} ausgerichtet. Was Simulatorkrankheit verursacht, ist, dass, wenn sich der Benutzer in der Realität geradeaus bewegt (in Richtung \vec{m}), um die blaue Säule zu erreichen, die VR-Ansicht eine Seitwärtsbewegung zeigt, siehe auch die untere Reihe von Abbildung 6.2 [414]. In Schritt 3 von Abbildung 6.3(d) sieht der Benutzer aufgrund der Sensordrift die gelbe Säule geradeaus in Richtung \vec{v} . Die Nichtübereinstimmung der realen und virtuellen Ausrichtung führt zu einer unnatürlichen Wahrnehmung der visuell-propriozeptiven Sinne, was im schlimmsten Fall zur Simulatorkrankheit führt.

Mit einem kleinen ψ' verspürt der Benutzer keine Unannehmlichkeiten [125, 133], siehe die in Abschnitt 6.4 erwähnte Arbeit zum umgeleiteten Gehen. Bei einem größeren ψ' fühlt sich der Benutzer, als würde er bei Annäherung an das Ziel seitlich weggezogen, da die Drift der relativen Entfernung jedes Objekt betrifft, das der Benutzer in der VR sieht. Leider kennt ein VR-System bei einer unbekannten Kopfrichtung (\vec{r}) ψ' nicht und kann daher \vec{v} nicht näher an \vec{r} ausrichten. In der Realität kompensieren Benutzer die Drift in Abbildung 6.3(d). Um den Abstand zwischen der roten und der orangefarbenen Säule zu erreichen, richten sie \vec{m} normalerweise näher an \vec{v} aus, d. h. sie drehen den Kopf. Bei größeren ψ' richten die Benutzer nicht nur ihre Köpfe, sondern auch ihre Körperorientierung aus, um \vec{m} an \vec{v} auszurichten. Methoden, die die IMU-Signale in Kombination mit den Bewegungen der Benutzer in VR berücksichtigen, müssen die Kompensationsstrategien der Benutzer verstehen [125], da diese die IMU-Signale und die Fortbewegung der Benutzer beeinflussen und natürlich auch die Lokalisierungsgenauigkeit, nämlich immer dann, wenn sich Benutzer untypisch bewegen, bzw. wenn dieses Bewegungsverhalten für eine Lokalisierungsmethode undefiniert ist.

6.2.3 Lösungsvorschlag

Der hier vorgeschlagenen Ansatz lässt sich wie folgt vereinfacht darstellen: Angenommen eine feinkörnige absolute Positionsverfolgung der Benutzer ist sowohl in Bezug auf Koordinaten als auch in Bezug auf Zeitstempel verfügbar. Dann kann die absolute Position eines Benutzers über die Zeit aufgezeichnet werden und einen Trajektorienvektor \vec{m} aus Paaren aufeinanderfolgender absoluter Positionen extrahiert werden. Unter der Annahme, dass Benutzer während der Vorwärtsbewegungen auch vorwärts blicken, m.a.W. der Kopf blickt in die Bewegungsrichtung des Körpers, d. h. $\vec{m} = \vec{r}$, dann kann ein VR-System \vec{r} ableiten und \vec{v} anpassen, und Sensordrift beseitigen, die Simulatorkrankheit verursachen. Natürlich ist der Kopf in der Realität nicht immer in die Bewegungsrichtung des Rumpfes ausgerichtet, siehe Abbildung 6.3(c). Wenn der Benutzer um $\omega = -20^\circ$ nach rechts schaut, würde die oben beschriebene Anpassung von \vec{v} immer noch eine Drift von 25° ergeben. Ähnliche Probleme treten auf, wenn \vec{m} von der tatsächlichen Trajektorie des Benutzers abweicht.

Um das Bewegungsverhalten und die Kompensationsstrategien von bspw. VR-Benutzern zu verstehen und deren Auswirkungen auf die Entwicklung und Fortpflanzung der Orientierungsfehler zu verstehen, wird im (nächsten) Abschnitt 6.3 eine ausführliche Fehleranalyse durchgeführt. Auf Basis der Erkenntnisse aus Abschnitt 6.3 kann dann die Methode zur (Re-)Kalibrierung der Orientierung und damit zur Fehlerkorrektur entwickelt werden.

6.3 Fehleranalyse

Abschnitt 6.3.1 motiviert die Analyse der Orientierungsfehler. Dann beschreibt Abschnitt 6.3.2 die Messinfrastruktur der Fehleranalyse. Anschließend beschreibt Abschnitt 6.3.3 die Struktur der Akkumulationsstudie für Orientierungsfehler. Der Abschnitt 6.3.3 bewertet die Fehler der Orientierung unter Laborbedingungen. Und Abschnitt 6.3.3 bewertet die Orientierungsfehler in einer realen VR-Anwendung.

6.3.1 Einführung

Moderne Lokalisierungssysteme, sowohl Kamera- als auch IMU-basierte Systeme verfolgen nur einzelne Positionen, i.d.R. der Füße

6 Orientierungsschätzung

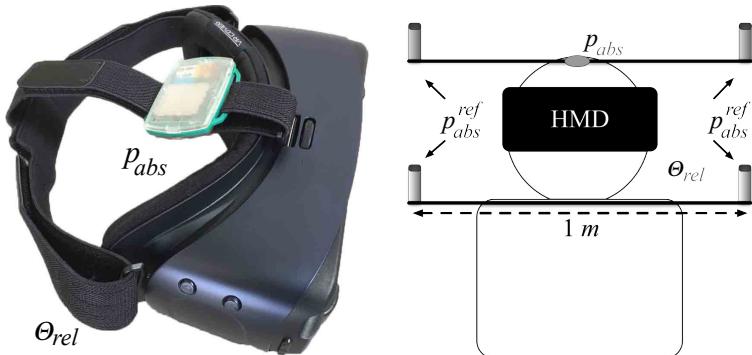
oder des Körpers und können nicht zum Ableiten der Kopf- und Körperpose verwendet werden, da ihre Genauigkeit nicht ausreicht. Außerdem leiden alle bekannten Systeme unter sich anhäufenden Orientierungsfehlern [129, 125, 133]. Abbildung 6.2 beschreibt einen typischen Orientierungsfehler in einer VR-Anwendung.

Head-Mounted Display (HMD)-Einheiten sind häufig mit Trägheitsmesseinheiten wie Beschleunigungsmessern, Gyroskopen und Magnetometern ausgestattet, mit denen auch die Pose des Kopfes geschätzt werden kann. In der Praxis sind IMU-basierte Schätzungen jedoch ungenau. Sowohl Beschleunigungsmesser (absolute Neigung und Roll) als auch Gyroskope (relative Neigung, Rolle und Gieren) können die absolute Kopforientierung (mit Gieren) nicht abschätzen. Während Magnetometer absolute Gier-Winkel liefern, sind sie empfindlich gegenüber Magnetfeldschwankungen und liefern oft eine falsche absolute Kopforientierung, z. B. in Innenräumen [172].

In VR-Anwendungen ist es notwendig, sowohl den Kopf als auch den Körper individuell zu lokalisieren, um eine optimale Visualisierung der virtuellen Welt zu ermöglichen. Außerdem ist es notwendig, den Orientierungsfehler und -drift so gering wie möglich zu halten, da VR-Benutzer unter Orientierungsdrift dazu neigen, implizit ihre Köpfe und Körper zur Seite zu drehen, das VR-Erlebnis dadurch negativ beeinträchtigt wird, sogar zu Symptomen der Simulatorkrankheit führen kann [129, 125] und so das VR-Erlebnis zerstört wird.

Methoden, die die IMU-Signale analysieren und die Sensordrift bestimmen, um die falsche Kopforientierung zu kompensieren, benötigen auch Informationen über die Rotation des Körpers in Bezug auf den Kopf, um eine gute Leistung zu erzielen. Dieses Wissen hilft bspw. VR-Systemen, wie z. B. der virtuellen Museumsanwendung von Roth et al. [362], um Orientierungsdrift in natürlichen Mehrbenutzeranwendungen zu bewältigen.

Das in dieser Dissertation vorgeschlagene Lokalisierungssystem verarbeitet Daten eines Trägheitssensors am Kopf oder Körper, bspw. der Hosentasche des Benutzers, um die Position und Orientierung des Kopfes oder Körpers zu bestimmen. Das System schätzt bspw. die Kopforientierung mit Trägheitssensoren und analysiert die Körperbewegung, um Drift zu kompensieren. Um die Sensordrift zu bestimmen und die



- (a) HMD-Trägheitssensor für die relative Ausrichtung θ_{rel} und Funksensor für die absolute Position p_{abs} .
- (b) 2-2 iGPS-Referenzsensoren p_{abs}^{ref} zur Messung der absoluten Ausrichtung von Kopf und Körper.

Abbildung 6.4: Hardweraufbau für die Experimente.

Ausrichtung der VR-Anzeige explizit anzupassen, müssen daher sowohl die Kopf- als auch die Körperausrichtung des Benutzers verstanden und berücksichtigt werden. Daher werden in diesem Abschnitt die Auswirkungen der Drift der Kopforientierung um die Gierachse auf die absolute Kopf- und Körperorientierung des Benutzers beim natürlichen Gehen in der VR und auf einen Sensor in einer Laborumgebung untersucht. Außerdem wird untersucht, wie viel Drift sich im Laufe der Zeit während typischer Bewegungen ansammelt.

6.3.2 Messinfrastruktur

Um herauszufinden, wie groß die Orientierungsfehler sind, die die Verfahren des Stands der Technik innerhalb weniger Minuten ansammeln, wurde die folgende Messinfrastruktur verwendet.

Auf einer Tracking-Fläche von ca. 40×35 m verwenden alle Experimente ein Samsung Galaxy Note 4-Mobiltelefon (Android 6.0.1, Qualcomm Snapdragon 805-CPU und 3 GB RAM²) mit einem 6DOF IMU-Sensor von InvenSense (MPU-6500) an einem Samsung GearVR HMD (Version SM-R320, mit einem 6DOF Bosch BMI055 IMU-

²Speicher mit wahlfreiem/direktem Zugriff, engl. Random-Access Memory, RAM.

6 Orientierungsschätzung



Abbildung 6.5: Mechanisch gelagerter 2-achsiger Rotationstisch mit Direktantrieb zur Bewegungssimulation.

Sensor), siehe Abbildung 6.4(a). Das HMD bietet in Kombination mit dem Unity / Oculus-Framework eine relative Kopforientierung bei etwa $60 \text{ Hz}(\theta_{rel})$. Die absoluten Positionen werden mit zwei verschiedenen absoluten Tracking-Systemen erfasst. (1) Ein Hochfrequenz-NP-Tracking-System (p_{abs}) verfolgt mehr als 100 Benutzer mit ca. 20 Hz in einem Bereich von bis zu $100 \text{ m} \times 100 \text{ m}$ mit einem $CEP_{95} < 0.224 \text{ m}$ pro Benutzer [158], siehe Abbildung 6.4(a). Ähnliche NP-Tracking-Systeme sind Open Source (einschließlich Hardware) und können zur Reproduktion der Experimente verwendet werden [434]. (2) Ein optisches NIKON IGPS-System (p_{abs}^{ref}) bestimmt die absolute Körperhaltung genau ($< 10 \text{ mm}$) bei ca. 30 Hz. Es werden vier Tracking-Ziele verwendet, von denen zwei sowohl am Kopf als auch an den Schultern eines Benutzers angebracht sind (siehe Abbildung 6.4(b)).

Um die Benutzer in der VR zu positionieren, wird p_{abs} verwendet und um ihre absoluten Kopf- / Schulterorientierungen und ihre Positionen zu messen, wird p_{abs}^{ref} verwendet. p_{abs}^{ref} wird als Referenz und zum Kalibrieren der relativen Kopforientierung von θ_{rel} für jeden Benutzer verwendet. Die Kalibrierung wird für jeden Benutzer und vor jedem Experiment neu durchgeführt, bevor der Benutzer zu laufen beginnt. Während des Gehens werden die relativen Orientierungen von θ_{rel} verwendet, um die Drehung der VR-Bilder zu visualisieren.

6.3.3 Drift-Akkumulationsstudie

Um den Einfluss der Kopf-Körper-Bewegung auf die Gierdrift zu verstehen, wurden zwei Studien durchgeführt. Erstens wurde die Sens-

ordrift unter Rotationen *ohne* plötzliche und starke Änderungen synthetischer VR-Bewegung, d. h. nur mit linearen Bewegungen, die im Bereich der typischen Kopfdrehgeschwindigkeiten in VR liegen, untersucht. Zweitens wurde die Sensordrift unter Rotationen *mit* plötzlichen und starken Geschwindigkeitsänderungen, d. h. mit nichtlinearen Bewegungen, typischer Kopfdrehungen in realen VR-Bewegungen untersucht. Um die Akkumulation der Drift über die Zeit zu bewerten, wird mit frisch kalibrierten HMD-Bewegungssensoren ($\vec{v}=\vec{r}$) begonnen. Die Referenz ist p_{abs}^{ref} . Da für dieses Experiment nur die Kopforientierung von Bedeutung ist, wurden die Schultersensoren nicht verwendet.

Synthetische VR-Bewegung

Es wurde die Orientierungsdrift des Samsung GearVR Systems gemessen, das auf einer Drehplatte sitzt (Aerotech ARMS260, mit einer maximalen Auflösung von $1500^{\circ}/s$ bei einer Auflösung von $< 0.0007^{\circ}$, vergleiche Abbildung 6.5) und das sich für 30 Minuten bei $32^{\circ}/s$ oder bei $41^{\circ}/s$ dreht. Diese sind typische Bereiche der Kopfdrehgeschwindigkeiten in VR, die im Rahmen der Dissertation gefunden wurden, indem 11 Probanden (durchschnittliches Alter 28.14 Jahre, min.: 19, max.: 52; durchschnittliche Höhe 1.72 m, min.: 1.51, max.: 1.78; ohne Behinderung; alle sind Abteilungsmitglieder: Studenten, Ingenieure, Wissenschaftler; normal oder auf normales Sehen korrigiert; 4 Brillen tragen; nur 1 hatte keine Erfahrung mit dem Gehen in VR Umgebungen; 7 Männer, 4 Frauen), natürlich und entspannt gingen.

Für die verschiedenen synthetischen Winkelrotationsgeschwindigkeiten zeigt Tabelle 6.1 auf der linken Seite, dass kostengünstige IMU-Sensoren mit sanfter linearer Bewegung eine Drift von 4.4° ($=5 \cdot 0.87^{\circ}$) auf 8.6° ($=5 \cdot 1.71^{\circ}$) akkumulieren innerhalb von 5 Minuten. Die Drift ist der Unterschied zwischen der realen Ausrichtung \vec{r} (vom genauen Drehplattensystem) und der virtuellen Blickrichtung \vec{v} (von θ_{rel}). Da sich der Fehler im Laufe der Zeit fast linear ansammelt, reicht es aus, ψ' am Ende des 30-Minuten-Intervalls zu messen.

Echte VR-Bewegung

In der Realität wird der Orientierungsfehler auch von verschiedenen Umgebungsbedingungen beeinflusst, wie z. B. der Aktivität des Benutzers (z. B. Gehen, Stehen und Drehen), der Bewegungsgeschwindigkeit

6 Orientierungsschätzung

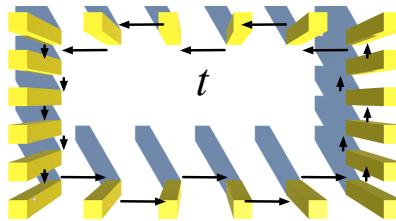


Abbildung 6.6: VR-Szene der Driftakkumulationsstudie: Die Teilnehmer gehen mit unterschiedlichen Geschwindigkeiten entlang der Trajektorie t .

des Benutzers [44] und der Situation, z. B. Ego-Shooter-Spiele oder Besuch eines virtuellen Museums. Daher wurde die Driftakkumulation in einer typischen VR-Erfahrung mit 34 Teilnehmern erfasst (Durchschnittsalter 25.03 Jahre, min.: 16, max.: 48; durchschnittliche Höhe 1.74 m, min.: 1.49, max.: 1.81; ohne Behinderung; alle sind Abteilungsmitglieder: Studenten, Ingenieure, Wissenschaftler; normal oder auf normales Sehen korrigiert; 11 tragen Kontaktlinsen oder Brillen; nur 2 hatten keine Erfahrung mit dem Gehen in VR-Umgebungen; 20 männlich, 14 weiblich).

Zuerst gehen sie natürlich und entspannt für 10 Minuten mit einer Durchschnittsgeschwindigkeit von 0.87 m/s (sie wurden gebeten, mit einer normalen Geschwindigkeit zu gehen; min.: 0.59 m/s, max.: 1.21 m/s, SD: 0.19 m/s) entlang der Zick-Zack-Trajektorie t , die in Abbildung 6.6 dargestellt ist. Um unterschiedliche Kopfdrehgeschwindigkeiten entlang t zu sehen, ist der Abstand zwischen den horizontalen Säulen dreimal so groß wie der Abstand zwischen den vertikalen Säulen (1 m). Zweitens laufen sie mit hoher Geschwindigkeit, aber immer noch auf natürliche und entspannte Weise (durchschnittlich: 1.43 m/s, min.: 1.23 m/s, max.: 2.01 m/s) entlang t , SD: 0.25 m/s für 10 Minuten. Die Teilnehmer beginnen immer mit einem frisch kalibrierten HMD. Der Fehler \vec{v} wurde wie zuvor gemessen, wobei p_{abs}^{ref} die Referenz war.

Die rechte Seite von Tabelle 6.1 zeigt die durchschnittliche akkumulierte Drift über die Zeit für die verschiedenen Bewegungsgeschwindigkeiten. Selbst bei einer niedrigen Durchschnittsgeschwindigkeit von 0.87 m/s (und einer durchschnittlichen Winkeldrehzahl von 32.36°) ak-

Tabelle 6.1: Rotationsgeschwindigkeit ω und akkumulierte Drift ψ für normale und schnelle synthetische sowie normale und schnelle reale Bewegungen.

	normal, synthetisch	schnell, synthetisch	normal, real (0.87 m/s)	schnell, real (1.43 m/s)
ω [°/s]	32	41	(avg.) 32.36 (min.) 0.05 (max.) 113.69 (SD) 11.3	(avg.) 41.18 (min.) 0.06 (max.) 276.24 (SD) 10.1
ψ [°/min]	0.87 (SD) 0.01	1.71 (SD) 0.02	(SD) 4.94 0.02	(SD) 5.49 0.02

kumulieren kostengünstige IMU-Sensoren eine Drift von etwa 24.7° ($= 5 \cdot 4.94^\circ$) innerhalb von 5 Minuten. Bei höheren Geschwindigkeiten tritt noch mehr Drift auf. Obwohl jeder Teilnehmer seinen Kopf unterschiedlich beschleunigt, ist die akkumulierte Drift für alle ungefähr gleich (SD: 0.18° bis 0.35°). Daher ist es ausreichend, die Durchschnittswerte in Tabelle 6.1 anzugeben. Es ist deutlich erkennbar, dass plötzliche und starke Geschwindigkeitsänderungen, d. h. nichtlineare Bewegungen oder reale VR-Bewegungen, in der Kopf-Körper-Bewegung zu größeren Orientierungsdriften führen als allmäßliche und kleine Änderungen, d. h. lineare Bewegungen oder synthetische Bewegungen.

Zusammenfassung

Reale Bewegungen führen zu durchschnittlichen Orientierungsfehlern, die fünfmal höher sind als bei synthetischen Bewegungen. Während die Driftakkumulationsstudie kostengünstige am Kopf montierte Sensoren sowohl in Szenarien linearer (Maschine) als auch nichtlinearer (menschlicher) Bewegung vergleicht, deckt sie auch typische menschliche Bewegungen in VR ab, z. B. Gehen mit unterschiedlichen Geschwindigkeiten, Laufen, Gehwege, Kurven usw., und repräsentiert somit reale Bewegungsszenarien.

6.4 Verwandte Arbeiten

Abschnitt 6.4.1 stellt herkömmliche Bayessche Methoden zur Orientierungsschätzung vor. Abschnitt 6.4.2 erörtert moderne Methoden der simultanen Lokalisierung und Kartografie. Dann führt Ab-

6 Orientierungsschätzung

schnitt 6.4.3 ähnliche Verfahren aus anderen Anwendungsbereichen ein. Abschnitt 6.4.4 diskutiert Methoden zur Schätzung der Richtung bei der Lokalisierung von Fußgängern. Und schließlich stellt Abschnitt 6.4.5 verwandte Arbeit vor, die die Fehlertoleranzbereiche der Orientierung in VR-Anwendungen untersucht.

6.4.1 Klassische Ansätze

VR-Systeme benötigen genaue Kopforientierungen. Traditionell verwenden sie (stationäre und lokale) Referenzsysteme mit absoluten Orientierungen der verfolgten Objekte. Normalerweise wird die Kopfhaltung deshalb im Verfolgungssystem, dem Referenzsystem, anstatt in der Nähe des HMD geschätzt. Letzteres würde allerdings die Latenz verringern und die Aktualisierungsraten erhöhen. Foxlin et al. [141] bspw. verwenden die IMU, sodass ein Referenzsystem nur die Schätzungen über die Zeit stabilisieren muss. Im Gegensatz zu dem in der Dissertation vorgeschlagenen Verfahren können sich Benutzer jedoch nicht frei in dem begrenzten Verfolgungsbereich bewegen. Foxlin et al. [142] stabilisieren das KF mit Referenzmomenten von fußmontierten Sensoren, d. h. sie müssen die wahren Sensorsignale kennen, wenn ein Fuß auf den Boden trifft, um die absolute Ausrichtung zu schätzen. Der hier vorgeschlagene Ansatz erfordert keine solchen zusätzlichen Sensoren. Darüber hinaus gibt die Ausrichtung der Füße (oder des Rumpfes) keinen Hinweis auf die Ausrichtung des Kopfes, da es sich um unterschiedliche Rotationssysteme handelt. Bis heute gibt es kein öffentlich bekanntes Rauschmessmodell, das die Dynamik des Kopfes genau wiedergibt und das länger als eine Minute zuverlässig funktioniert [170, 7]. Selbst mit optimierten Bayes-Filttern ist eine langfristige stabile IMU-basierte Verfolgung hochdynamischer Objekte (z. B. Kopf) auf Basis kostengünstiger IMUs für Endverbraucher unmöglich [142].

6.4.2 Simultane Lokalisierung und Kartografie

Kinect [250], Tango [474] und ARKit [117] verwenden SLAM, das nur unter eingeschränkten Bedingungen gut funktioniert (kleine Räume, heterogene Oberflächentexturen, statische Szenen, und homogener Ausleuchtung) [57]. Aufgrund methodischer (Erkennung labiler Merkmale [297, 57]) und physikalischer (Tiefenerfassung [230, 250, 474, 237]) Einschränkungen von RGB (RGB-Farbraum der Grundfarben Rot, Grün und Blau) [297]- und RGB-D-Sensoren (RGB-Farbraum mit Tiefeninfor-

mationen, engl. Depth), können heutige SLAM Verfahren die hier vorgeschlagenen dynamischen und flexiblen VR-Erlebnisse nicht für immersive, frei begehbarer VR-Anwendungen bereitstellen [103, 306, 133].

6.4.3 Anwendungsspezifische Ansätze

Orientierungsschätzer für die NP-Verfolgung mit IMUs unterscheiden sich von typischen VR-Systemen [103, 306]. Wenn hochpräzise und teurere Sensordaten verfügbar sind, schätzen einige Arbeiten sowohl die Position als auch die Ausrichtung mit Bayesschen Filtern [145]. Das üblicherweise verwendete EKF integriert doppelt über die Sensorsignale, um Orientierung und Position zu bestimmen, und erfordert präzise Modelle des Beschleunigungssensors, um die lineare Beschleunigung richtig zu extrahieren [361]. Für kostengünstige IMUs, die solche präzisen Messmodelle nicht unterstützen, sind Bayessche Filter keine praktikable Lösung, es sei denn, zuverlässige Magnetometerdaten können zur Reduzierung des Gier-Winkelfehlers verwendet werden [215, 7]. Im Gegensatz dazu sind für den hier vorgeschlagenen Ansatz kostengünstige IMUs ohne Magnetometer ausreichend.

Während Verfahren zur Erkennung menschlicher Aktivitäten wie Stehen, Gehen und Sitzen diese direkt mit IMUs erkennen können, die an anderen Körperteilen als dem Kopf angebracht sind, steht die Schätzung der Kopforientierung nicht im Mittelpunkt dieser Ansätze. Windau und Itti [472, 33] klassifizieren Aktivitäten mit am Kopf montierter IMUs, aber für gute Ergebnisse muss der Kopf eines Benutzers in die Bewegungsrichtung zeigen. Im Gegensatz dazu benötigt der hier vorgeschlagene Ansatz nur eine am Kopf montierte IMU und Benutzer können ihre Köpfe frei bewegen. Beauregard et al. [33] verwenden eine am Helm montierte IMU, jedoch für einen anderen Zweck, nämlich um die Schrittänge [225] und den Kurs mithilfe eines KNN zu schätzen. Dies funktioniert jedoch nur für feste Ausrichtungen von Helm zu Körper, d. h. solange die Ausrichtungen der Sensoren mit der Ausrichtung des Körpers übereinstimmen [224]. Im Gegensatz dazu können Benutzer bei dem hier vorgeschlagenen Ansatz ihre Köpfe frei drehen. Steed und Julier [413] kompensieren die Drift des Gier-Winkels zweier verschiedener Rotationssysteme (Rumpf und Hand), indem beide Rotationssysteme zu einem einzigen (den Rumpf) zusammengeführt werden. Dann sammelt sich die gesamte relative Gierdrift nur entlang einer Achse an, sodass ihre Methode keinen korrekten absoluten Gier-Winkel

6 Orientierungsschätzung

liefern kann. Sie nutzen auch das Wissen über das menschliche Bewegungsverhalten, d. h. Hand-zu-Körper-Bewegung, aus, um den absoluten Gier-Winkel basierend auf dem Kontext abzuleiten, z. B. nehmen sie an, dass VR-Benutzer beim Öffnen einer Tür auf ihre Hände schauen. Neben der Notwendigkeit, Sensorgeräte in den Händen zu halten, ist dies eine ziemlich starke Annahme. Im Gegensatz dazu geht die Methode dieser Dissertation davon aus, dass Menschen beim Gehen vorübergehend den Kopf gerade halten. In Abschnitt 6.7 wird gezeigt, dass diese Annahme sehr häufig der Fall ist.

6.4.4 PDR Ansätze

Menschenzentrierte Navigationssysteme, die die Fußgänger-Koppelnavigation mit IMUs verwenden, benötigen für zuverlässige Ergebnisse auch die Ausrichtung von Sensoren. Es gibt Studien für verschiedene Sensorkombinationen [526, 327, 28] und mit unterschiedlichen Positionen der IMU am Körper (Hände [368], Füße [182], Rucksäcke [442], Handgelenke [62] und Beine [292]). Konzeptionell schätzt PDR sowohl den Versatz der Position (aus der Schritterkennung und der Schrittängenschätzung) als auch die Orientierung des Versatzes (aus Gyroskopen und Magnetometern) [7, 170]. Bei fußmontierten Sensoren kann der resultierende Positionsfehler bis zu [0.3%; 2%] der gesamten zurückgelegten realen Entfernung [44] betragen, da die Systeme wiederkehrende Punkte ausnutzen können, wenn die IMU sich nicht bewegt. Sie können dann ihre Zustände zurücksetzen und genaue Verschiebungen berechnen. Mit ausschließlich am Kopf montierten IMUs kann dieses Verfahren nicht eingesetzt werden.

Einige PDR-Forschungen verwenden Magnetometer, um die Schätzungen der Kopforientierung zu stabilisieren [326, 452]. Die Idee ist, dass jeder Schritt zu einer bestimmten Kopfdrehung führt. Da das Kopfrotationssignalmuster *a priori* gelernt werden kann, kann es später erkannt und zur Laufzeit verwendet werden, um Kopforientierungsfehler zu korrigieren [367]. Konzeptionell können diese Ansätze auch für andere Arten von am Kopf montierten IMU-Sensoren verwendet werden. Es gibt jedoch weitere Nachteile, die der im Rahmen der Dissertation vorgeschlagene Ansatz vermeidet. Zunächst müssen herkömmliche Filter *pro Benutzer* trainiert und parametrisiert sein. Zweitens, wenn sich Körper und Kopf eines Benutzers synchron drehen, sind solche Filter verwirrt und müssen manuell wiederhergestellt werden. Im Gegen-

satz dazu vermeidet der hier vorgeschlagene Ansatz nicht nur unzuverlässige Magnetometer, sondern auch benutzerspezifische Filter, ist immun gegen synchrone Bewegungen von Körperteilen und lässt sich auf beliebige Körperteile wie Arme (Uhren an Handgelenken), Beine (Hosentasche) und Kopf übertragen.

6.4.5 Ansätze des umgeleiteten Gehens

Während sich die oben erwähnten verwandten Arbeit mit der Frage befassen, wie man zu den richtigen Orientierungsvektoren gelangt, beruht umgeleitetes Gehen (engl. redirected walking) auf der Verfügbarkeit genauer Posenschätzungen und speist künstliche Sensordrift in die VR-Ansicht ein, um die Wahrnehmung und Orientierung der Entfernung in der VR zu modifizieren. Ziel ist es, den verfügbaren wahrgenommenen virtuellen Raum zu vergrößern. Das Ergebnis ist, dass ein Benutzer in VR keine Verkleinerung von 24% oder eine Verkleinerung von 26% in seiner Gehstrecke erkennt. Darüber hinaus können virtuelle Kopfdrehungen bis zu 30% höher oder 20% niedriger sein als die realen Kopfdrehungen des Benutzers, ohne dass der Benutzer dies bemerkt [416]. Daher kann ein Benutzer bei einer Gehgeschwindigkeit von 1 m/s innerhalb von 5 m um 17° neu ausgerichtet werden, ohne dass der Benutzer dies bemerkt. Diese Drifttoleranz erlaubt dem hier vorgeschlagenen Klassifikator kleine Fehler zu machen [125].

6.5 Beiträge

Der Beitrag der Methode besteht darin, dass jede Gier-Orientierung, die direkt aus IMU-Sensordaten geschätzt wird, mit datengetriebener komplementärer Informationsfusion kalibriert oder korrigiert werden kann. Zusätzliche Informationen (absolute Positionen des Benutzers) werden verwendet, die nicht unter den gleichen Problemen wie die Orientierungsschätzung (akkumulierende Sensordrift) leiden und somit einen komplementären Effekt zu den Problemen herkömmlicher Gierschätzer haben. Das neuartige Verfahren löst die Problem, indem es die IMU-Daten anders betrachtet und verwendet, als bekannte Verfahren: Es erkennt Momente, in denen sich der Benutzer in seine Blickrichtung vorwärts bewegt, weil für diese $\vec{m}=\vec{r}$ -Momente oder $\omega=0^\circ$ die Sensordrift bekannt ist oder aus den Informationen eines komplementären Sensors (hier die Position) berechnet werden kann. Somit kann die Drift der virtuellen Ansicht angepasst werden. Die Alternativen, ein

6 Orientierungsschätzung

Magnetometer zu verwenden oder den Benutzer mit $\omega \approx 0^\circ$ bewegen zu lassen, sind nicht praktikabel, da ferromagnetische Störquellen in Innenräumen die Magnetfeld-basierte Orientierungsschätzung verfälschen und sich negativ auf die VR Erfahrung auswirken, m.a.W. reduzieren das Eintauchen und die Präsenz in der virtuellen Welt. Daher stellen sich folgende Forschungsfragen (RQ):

RQ4: Können datengetriebene Verfahren die Orientierungsschätzung eines Fußgängers aus verrauschten MEMS-Trägheitsmessungen verbessern?

RQ4a: *Mit welcher Genauigkeit können datengetriebene Methoden die Ausrichtung von Kopf-zu-Rumpf mit am Kopf platzierten Beschleunigungs- und Drehratensensoren abschätzen?*

RQ4b: *Kann eine datengetriebene Klassifizierung Momente in der menschlichen Bewegung finden, um die Ausrichtung verschiedener Körperteile zu bestimmen?*

6.6 Methode

Abschnitt 6.6.1 führt in den Prozess der Signalverarbeitung ein. In Abschnitt 6.6.2 werden die Datenvorverarbeitung und die Datenfilterung erläutert. In Abschnitt 6.6.3 wird der Einfluss menschlicher Bewegungen auf die Messungen der Trägheitssensoren erörtert. Abschnitt 6.6.4 beschreibt die statistischen und heuristischen Merkmale, die die datengetriebene Methode zur Analyse von Bewegungsmustern verwendet. Schließlich stellt Abschnitt 6.6.5 das Verfahren zur immersiven Anpassung der VR-Darstellung vor.

Es wird überwachtes maschinelles Lernen verwendet, um verschiedene Bereiche³ von ω zu klassifizieren. Wenn unter allen Möglichkeiten die $\omega=0^\circ$ -Momentklasse die höchste Wahrscheinlichkeit hat, wird ein $\vec{m}=\vec{r}$ -Moment festgestellt. Aus den IMU-Daten (Beschleunigungsmesser) wird die lineare Beschleunigungskomponente extrahiert, d. h. die reale Translationsbewegung (ohne Schwerkraft) in jede Richtung, und sie in Kombination mit den Gyroskopdaten verwendet, um bestimmte Merkmale abzuleiten, die einen bestimmten Bereich charakterisieren

³Ein Bereich beschreibt ein Intervall von benachbarten Gier-Winkeln, die einer ω -Momentklasse zugeordnet sind.

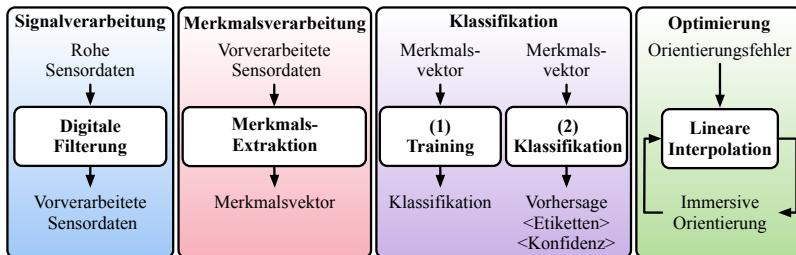


Abbildung 6.7: Verarbeitungskette zur Schätzung der Kopforientierung.

und um ω darzustellen. Es wird ein Klassifikator für alle ω -Klassen a priori für aufgezeichnete und gekennzeichnete Merkmalsdaten trainiert. Zur Laufzeit wird ω in Live-Sensordaten klassifiziert, um einen $\omega=0^\circ$ -Moment zu erkennen, wenn diese Klasse die höchste Wahrscheinlichkeit bzw. Vertrauen ergibt.

Abbildung 6.7 beschreibt die Verarbeitungskette zur Schätzung der Kopforientierung. Zunächst werden die rohen Sensorsignale mit digitalen Filtern (Abschnitt 6.6.1 und 6.6.2) geglättet und vorverarbeitet. In einem Trainingsschritt werden Merkmale (Abschnitt 6.6.4) für bekannte Bereiche von ω aus beschrifteten Trainingsmustern extrahiert, um einen Klassifikator zu trainieren. Während eine feinkörnige Auflösung von ω -Bereichsklassen die Klassifizierung und ihr Vertrauen verbessert, benötigt sie auch mehr Daten für das Training und mehr CPU-Zyklen für die Klassifizierung (Abschnitt 6.7.2 schlägt Parameter vor). Zur Laufzeit verarbeitet der trainierte Klassifizierer die Merkmale (geglätteter) unbekannter Signale und gibt die am besten passende ω -Bereichsklasse und deren Klassifizierungsvertrauen zurück. Für eine bessere Klassifizierungsrate werden physikalische Einschränkungen eingeschlossen, z. B. menschenzentrierte Bewegungseinschränkungen wie die Unmöglichkeit, den Kopf um $\pm 15^\circ/\text{ms}$ zu drehen. In $\omega=0^\circ$ -Momenten wird die Kopforientierungsdrift ψ' bestimmt und eine lineare Interpolation verwendet, um die Drift zu reduzieren, d. h. um \vec{v} an \vec{r} auf immersive Weise anzupassen, damit die Benutzer keine Anpassung bemerken (Abschnitt 6.6.5).

6 Orientierungsschätzung

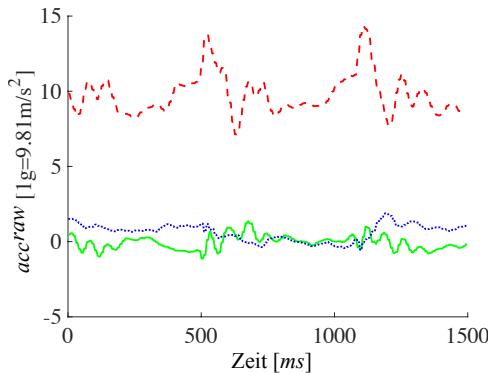


Abbildung 6.8: Rohes (engl. raw) Beschleunigungssignale acc^{raw} eines $\omega=-45^\circ$ -Moments. Exemplarische Signale der X- (rot), Y- (grün), und Z (blau)-Richtung.

6.6.1 Signalverarbeitung

Rohe Beschleunigungsmesser- und Gyroskopdaten von einem kostengünstigen IMU-Sensor sind zu verrauscht, um zuverlässige und möglichst allgemeine Merkmale zu extrahieren. Sie benötigen eine Vorverarbeitung. Abbildung 6.8 zeigt beispielhafte rohe Beschleunigungssignale acc^{raw} für ein $\omega=-45^\circ$ -Moment mit den gezeigten Sensorachsen in Abbildung 6.9. Typische kostengünstige Beschleunigungsmesser, wie sie in modernen Mobiltelefonen eingesetzt werden, verfolgen die Schwerkraft und Beschleunigung bei 200 Hz mit bis zu $\pm 16g$. Gyroskope verfolgen die rohe Winkelgeschwindigkeit gyr^{raw} bei 200 Hz mit bis zu $\pm 2000^\circ/s$. Zusätzlich zum Rauschen haben die Beschleunigungskurven eine Schwerkraftkomponente, die am besten in der gestrichelten roten Kurve (vertikaler Versatz von 9.81 m/s) zu sehen ist, die zeigt, wie sich der Kopf des Benutzers auf und ab bewegt, wenn er/sie läuft (ein Gangzyklus, d. h. zwei gezeigte Schritte). Um die Kopf-Körper-Haltung des Benutzers genau zu beschreiben, muss die Bewegung des Kopfes, seine Haltung und seine Drehung analysiert werden. Daher wird das acc^{raw} in seine Schwerkraftkomponente acc_{grav} , die die Pose beschreibt, d. h. Nick und Roll, und die lineare Beschleunigung acc_{lin} , die die Bewegung beschreibt, getrennt. Die Daten gyr^{raw} und acc^{raw} werden auch gefiltert, um sie zu glätten. Die Details der Eingabedatenverarbeitung folgen in Abschnitt 6.6.2.

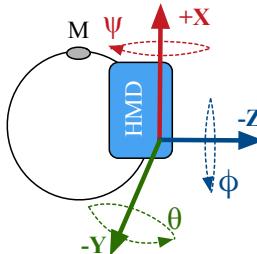


Abbildung 6.9: Seitenansicht des Kopfes mit realen Ausrichtungen X (rote gestrichelte Linie), Y (grüne Linie), und Z (blaue gepunktete Linie). Gier- ψ , Nick- θ und Rollwinkel ϕ . Positionssender M für die reale Bewegungsrichtung \vec{m} . ψ bezieht sich auf den Boden und entspricht \vec{r} in Abbildung 6.3. ψ' in Abbildung 6.3 ist die Ausrichtung des virtuellen Kopfes in Bezug auf \vec{r} .

6.6.2 Datenvorverarbeitung

Für die Vorverarbeitung der IMU-Daten werden Schiebefenster verwendet, d. h. sechs Fenster, um die Daten von acc^{raw} (3 Achsen) und gyr^{raw} (3 Achsen) zu speichern. Ein einfaches gleitendes Mittelwert Filter (SMA), das zur Beseitigung von Rauschen verwendet wird, ist zu einfach, da es signifikante Signaländerungen abflacht, die bspw. auf Bewegungsänderungen hinweisen. Im Gegensatz dazu entfernt ein Savitzky-Golay-Filter (SG) Signalrauschen, Vibrationen und (kleine) Bewegungsartefakte, während signifikante Änderungen im gefilterten Signal verbleiben. Daher wird ein SG-Filter mit einem kleinen Fenster mit der Länge $n=25$ verwendet, das ausreichend ist, um die Daten nur zu glätten, eine Polynomordnung von $P=3$ (höherer Ordnung erfasst nur Rauschen) und die üblichen SG-Filter Faltungskoeffizienten C_k . [388]

Beschleunigungsmesser-Datenfilterung

Um die gyr^{raw} -Daten vorzuverarbeiten, reicht es aus, das Rauschen zu eliminieren, indem es mit dem SG-Filter in gyr^{SG} geglättet bzw. veredelt wird. Da jedoch die Schwerkraftkomponente des Beschleunigungssignals die reale Pose, d. h. Nick und Roll, darstellt und die lineare Beschleunigung die Bewegung darstellt, wird nun das Beschleunigungsmessersignal in seine Schwerkraft- und linearen Komponenten getrennt. Daher wird zuerst die lineare Beschleunigung mit einem linearen rekursiven Filter isoliert und dann die Schwerkraft abgeleitet,

6 Orientierungsschätzung

m.a.W. Schwerkraft acc_{grav} = Rohbeschleunigung acc_{raw} minus lineare Beschleunigung acc_{lin} :

$$acc_{grav} = acc_{raw} - acc_{lin}. \quad (6.1)$$

Die oberen Graphen in Abbildung 6.10 zeigen acc_{lin}^{IIR} , die gefilterten Signale, die die linearen Beschleunigungskomponenten erst nach Anlegen eines Hochpass-(engl. High-Pass, HP-) ⁴ und eines Tiefpass-(engl. Low-Pass, LP-) ⁵ filters enthalten.

Laut Do et al. [101] gibt es sowohl für HP als auch für LP die Möglichkeit entweder ein Filter mit einer schnellen Fourier-Transformation (engl. Fast Fourier Transform, FFT), ein Filter mit einer endlichen Impulsantwort (engl. Finite Impulse Response, FIR) oder ein Filter mit einer unendlichen Impulsantwort (engl. Infinite Impulse Response, IIR) zu verwenden. Mathematisch liefert die FFT die genauesten Ergebnisse, ist jedoch langsam und passt daher nicht mit den Ansprüchen einer VR-Anwendung zusammen, da diese für die grafische Visualisierung bereits die meisten Ressourcen der mobilen HMD-CPU verbraucht. Sowohl FIR- als auch IIR-Filter erfordern mehrere Verarbeitungsdurchgänge für jedes Fenster. In Vorversuchen wurden im Rahmen der Dissertation FIR- und IIR-Filter verglichen. Es wurde festgestellt, dass für VR-Anwendungen IIR-Filter die FIR-Filter übertreffen, da sie bei einer ähnlichen Filterordnung den kleinsten Fehler, die schnellste Laufzeit und die geringste Verzögerung ergeben.

Für jede Beschleunigungsmesserachse werden daher zwei IIR-Filter (LP / HP) mit einem Butterworth-Filterdesign verwendet, von dem bekannt ist, dass es nach korrekter Vorinitialisierung schnell und zuverlässig ist [301]. Die linearen und Schwerkraftbeschleunigungskomponenten sind nicht SG-gefiltert, sondern IIR-gefiltert, um explizit die Gravitationskraft oder die lineare Beschleunigung zu extrahieren.

Da die menschliche Bewegung unterhalb einer Frequenz von 18 Hz [226] stattfindet, wird das Signal (nach Shannon-Nyquist) über 40 Hz abgetastet. Das LP kompensiert Artefakte wie extrem schnelle Kopfbewegungen oder Vibrationen, entfernt Geräusche und extrahiert signifikante menschliche Bewegungen mit einer Frequenz unter

⁴Hochpassfilter entfernt Komponenten mit geringer Schwerkraft.

⁵Tiefpassfilter entfernt häufig auftretende Rausch- und Bewegungsartefakte.

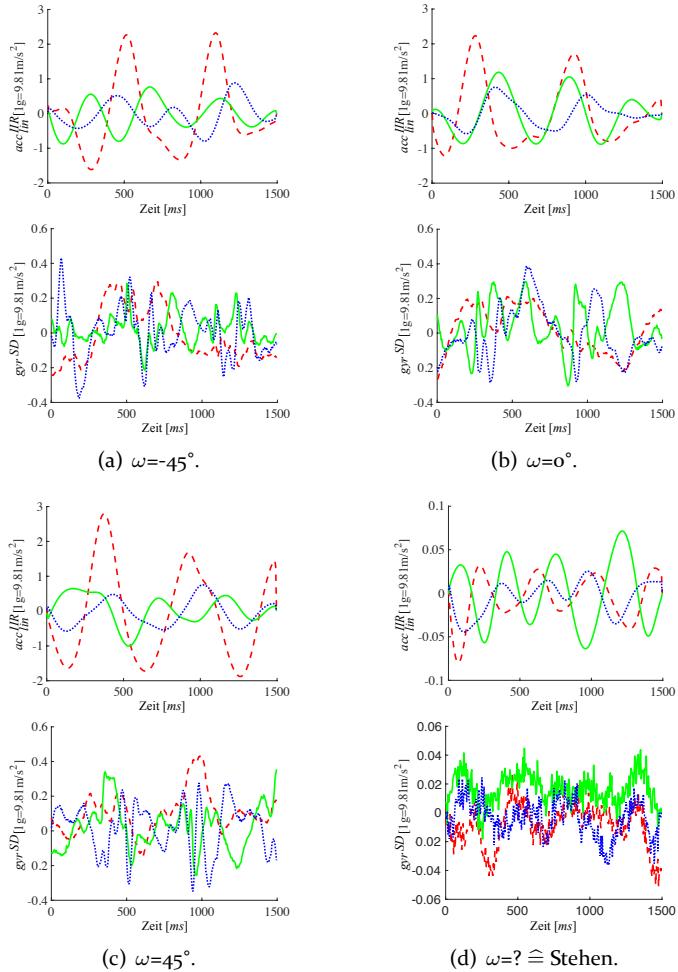


Abbildung 6.10: Oben: IIR (LP, HP)-gefilterte Linearbeschleunigungsdaten acc_{lin}^{IIR} . Unten: SG-gefilterte Gyroskopdaten gyr^{SG} . Es wird ungefähr ein Gangzyklus (= 2 Fußschritte) gezeigt. Koordinatensystem des Sensors am Kopf: rote gestrichelte Linien X oder ψ , grüne durchgezogene Linien Y oder θ und blaue gepunktete Linien Z oder ϕ .

6 Orientierungsschätzung

40 Hz.⁶ Die Grenzfrequenz des HPs wird auf 5 Hz eingestellt, um die langfristige vertikale Signaldrift zu kompensieren.

Jede neue Messung des Beschleunigungsmessers x_i wird in ein Schiebefenster mit der Länge $n=12.000$ geschoben. Dieses Fenster kann die Sensordaten von einer Minute speichern, da 200 Messungen pro Sekunde durchgeführt werden. Dies gibt dem IIR-Filter genügend Verlauf und ermöglicht die Erfassung aller verfügbaren Frequenzen / Aktivitäten, die in das Signal eingebettet sind. Das Butterworth-Filter liefert die gefilterten Beschleunigungen y_i für die Beschleunigungsmesserdaten x_i :

$$y_i = \frac{1}{a_0} \left(\sum_{i=0}^N b_i \cdot x_{i-1} - \sum_{j=1}^M a_j \cdot y_{i-j} \right), \quad (6.2)$$

wobei $a_{0\dots M}$ und $b_{0\dots N}$ die Filterkoeffizienten der Rückkopplung und der Vorwärtskopplung sind. Im Rahmen einer Parameteroptimierung wurden optimale Filterkoeffizienten abgeleitet. Es werden eine Filterordnung von $N_{low}=M_{low}=3$ für die Tiefpassfilter und $N_{high}=M_{high}=1$ für die Hochpassfilter verwendet. Die Filterordnung definiert die Anzahl der Durchgänge über das Schiebefenster. In einer Vorstudie wurde festgestellt, dass diese Werte einen angemessenen Kompromiss zwischen Filtertiefe und Rechenzeit darstellen.

Da die Zerlegung des Beschleunigungssignals (in acc_{grav}^{IIR} und acc_{lin}^{IIR}) mit einer starken Filterung und damit mit Informationsverlust von einmaligen Sensoreigenschaften einhergeht, wird auch das SG-gefilterte Rohbeschleunigungssignal (acc^{SG}) verwendet, um die Signaleigenschaften und die Beziehung zwischen der Schwerkraft und den linearen Beschleunigungskomponenten zu berücksichtigen.

Dies ergibt insgesamt vier Signalströme vorverarbeiteter Daten (mit jeweils drei Achsen), die zum Extrahieren von Merkmalen verwendet werden: SG-gefilterte Gyroskoprotationen gyr^{SG} , SG-gefilterte Beschleunigungen (einschließlich Pose und Bewegung) acc^{SG} und zwei IIR-gefilterte Beschleunigungen, getrennt in einen Strom für die

⁶Einige Ergebnisse der Experimente: (1) Frequenzen in [5; 20] Hz decken alle menschlichen Bewegungen in VR ab. (2) Benutzer bewegen sich in VR noch langsamer, als in der Realität, d. h. eine noch niedrigere Abtastung wäre möglich.

Schwerkraft (Pose) acc_{grav}^{IIR} und einen für die linearen (Bewegungs-) Beschleunigungen acc_{lin}^{IIR} .

6.6.3 Einfluss von Bewegung auf IMU-Signale

Um die Informationen über die Kopf-Körper-Haltung aus den vorverarbeiteten Daten zu extrahieren, werden zur vereinfachten Veranschaulichung der Methode exemplarisch drei (Klassen von) ω -Winkeln angenommen. Abschnitt 6.8 wertet einen umfangreicheren Klassifikator für 7 ω -Winkel aus. Abbildung 6.10 zeigt die IIR-gefilterten linearen Beschleunigungen acc_{lin}^{IIR} in der oberen Reihe und das SG-gefilterte Gyroskopsignal gyr^{SG} in der unteren Reihe. Vorerst wird Abbildung 6.10(d) ignoriert, da hier der Benutzer stillsteht.

In den ersten drei Fällen, in denen der Benutzer mit unterschiedlichen Gier-Orientierungen ω vorwärts geht, können Oszillationen in der X -Achse (rot gestrichelte Kurve) eindeutig identifiziert werden, wenn sich der Kopf des Benutzers bei jedem Schritt auf und ab bewegt. Da sich der Kopf bei jedem Schritt auch hin und her (Z -Achse) und seitwärts (Y -Achse) bewegt, treten in den blauen und grünen Kurven kleinere Schwingungen auf. Da die oberen drei Diagramme gleich aussehen, funktioniert ein Klassifizierer, der erkennt, ob ein Benutzer geradeaus in die Blickrichtung geht ($\omega=0^\circ$) oder ob der Kopf zur Seite gedreht ist, kaum wenn nur Beschleunigungssignale verwendet werden.

Wenn Benutzer gehen, gleichen sie die Auf- und Abbewegung ihres Kopfes mit einer nickenden Bewegung in der Nickachse θ (grüne durchgezogene Linie) aus. Im $\omega=0^\circ$ -Moment (Abbildung 6.10(b)) führt das Nicken zu einer deutlichen Schwingung von θ . Wenn der Kopf in Bezug auf die Bewegungsrichtung zur Seite gedreht wird, wird das Nicken sowohl über die Achse ϕ (blaue gepunktete Linie) als auch über die Achse θ (grüne durchgezogene Linie) verteilt. Es gibt ähnliche Rotationen, um die anderen Kopfbewegungen auszugleichen, die in den Beschleunigungsspitzen zu sehen sind. Daher sind die Gyroskopdaten auch erforderlich, um den ω -Winkel zu klassifizieren. Während bei Vorwärtsbewegungen das Beschleunigungssignal seine Kräfte (sowohl Schwerkraft als auch Linearität) über alle Achsen verteilt, konzentrieren sich die Kräfte hauptsächlich in der Z -Achse im $\omega=0^\circ$ -Moment und wechseln zur Y -Achse, wenn der Kopf zur Seite gedreht wird. Die Korrelation zwischen den Y - und Z -Achsen gibt

6 Orientierungsschätzung

somit einen Einblick, wie weit der Kopf nach links / rechts gedreht ist.

Abbildung 6.10(d) zeigt die gefilterten Signale, wenn der Benutzer steht. Sie unterscheiden sich erheblich (sowohl in Form als auch im Maßstab) von den Signalen, wenn der Benutzer geht. Ein Klassifikator kann diesen Fall somit leicht erkennen. Da es keine Bewegungsrichtung \vec{m} und keinen genau definierten ω -Winkel gibt, wenn der Benutzer steht, wird dies im Folgenden als $\omega=?$ dargestellt.

Neben der linearen Beschleunigung, die die Bewegung des Kopfes im Raum darstellt, ist auch die Haltung des Kopfes interessant, da sie anzeigen, wie Menschen ihren Kopf halten, während sie sich in bestimmten Bewegungszuständen befinden. Da die Schwerkraft die Neigung und Rollneigung eines Sensors in Bezug auf den Boden angibt, basiert die Haltung des Kopfes auf den Schwerkraftkomponenten pro Achse. Intuitiv wird deshalb auch der IIR-gefilterte Schwerkraftdatenstrom acc_{grav}^{IIR} benötigt, der in Abbildung 6.10 weggelassen wird.

6.6.4 Merkmalsauswahl

Nachdem in Abschnitt 6.6.3 motiviert wurde, dass die $4 \cdot 3 = 12$ -Signalströme genügend Informationen enthalten, um den ω -Winkel zu extrahieren und zu klassifizieren, werden sie in einen Strom von Merkmalen verwandelt, die die Intuition erfassen und die der Klassifizierungsalgorithmus optimal ausnutzen kann.

Mögliche Merkmale sind aus der Literatur bekannt [355, 27]. Sie haben unterschiedliche Laufzeitkomplexitäten (sowohl während des Trainings als auch zur Laufzeit) und ergeben unterschiedliche Vertraulichkeiten. Im Rahmen einer Vorstudie in der Dissertation wurde ein minimaler Satz von Merkmalen gefunden, die eine Klasse (Bereich von aufeinanderfolgender ω -Winkel) eindeutig beschreiben und von den Merkmalen anderer Klassen (Bereiche mit anderen ω -Winkelfolgen) trennbar sind. Zu beachten ist, dass für die Signalfilterung in Abschnitt 6.6.1 die Fenstergröße entsprechend den Anforderungen das Filter ausgewählt wird. In ähnlicher Weise müssen die Berechnungen der Merkmale ein bestimmtes Fenster der (gefilterten) Signaldaten verarbeiten. Die obige Diskussion von Abbildung 6.10 hat motiviert, dass mindestens die Signale eines vollständigen Gangzyklus benötigt werden, um den ω -Winkel zu klassifizieren. Aus einer Voruntersuchung

ist bekannt, dass das Hinzufügen weiterer Gangzyklen die Ergebnisse nicht verbessert, sondern die Merkmalsextraktion verlangsamt. Wenn Benutzer in einem typischen VR-Setup mit normaler Geschwindigkeit gehen, passt ein Zyklus in ungefähr 1.000 ms, d. h. in 200 (gefilterte) Signalwerte.⁷ In der Vorstudie wurde auch festgestellt, dass für jeden der 12 Signalströme die folgenden vier Merkmale (berechnet über das gesamte Fenster der Größe $n=200$) die erforderlichen CPU-Ressourcen niedrig halten und gleichzeitig eine gute Klassifizierung ergeben. Im Folgenden werden einige Gründe angegeben, warum die Merkmale erfassen, was ein Klassifikator benötigt.

Merkmal 1 - Mittelwert, μ

In einem Schiebefenster, das die Sensordaten von genau einem Gangzyklus enthält, bleibt der Mittelwert (engl. mean) μ der abgetasteten Werte konstant, auch wenn das Fenster über Signale die aktive Bewegung repräsentieren, gleitet. Konzeptionell gibt es immer ein oder zwei volle Wellen (ein Gangzyklus) im Fenster (abhängig von der Achse), nur der Schnittpunkt (engl. cut-off point) variiert an dem das Fenster beginnt. Wenn das Schiebefenster mehr (oder weniger) als einen vollständigen Gangzyklus enthält, dann variiert der Mittelwert mit dem Schnittpunkt, da ein variierendes Fragment einer Welle ein zusätzlicher (oder fehlender) Teil von einem Fenster ist. Daher schwingt der Mittelwert μ und erfasst (zu jedem Zeitpunkt), an welcher Stelle des Gangzyklus das Schiebefenster aktuell beginnt. Dies gilt sowohl für den Beschleunigungsmesser als auch für Gyroskopdaten. Das Merkmal μ enthält auch Informationen, die bei der Klassifizierung von $\omega = ?$ -Momente helfen, in denen sich der Benutzer nicht bewegt, da hier der Mittelwert der Werte nicht mehr (so stark) schwingt, sondern bei 0 bleibt, siehe Abbildung 6.10(d) (zu beachten ist der unterschiedliche kleinere Maßstab).

Merkmal 2 - Standardabweichung, SD

Die Standardabweichung SD repräsentiert die Intensität der Signalschwankungen (verursacht durch Nicken, Balancieren usw.). Da der Kopf in Folge des Gehens nach rechts und links dreht (entlang der Y-

⁷Laut Kader et al. [3] gehen Benutzer mit 1.5 Schritte/s mit 1.4 m/s in der Realität, sind aber tendenziell langsamer in VR: langsame Geschwindigkeit = 0.75 m/s, normale Geschwindigkeit = 1.0 m/s und schnelle Geschwindigkeit = 1.25 m/s.

6 Orientierungsschätzung

Achse) ist SD ein vernünftiges Merkmal für die Klassifizierungsaufgabe. Die ersten drei Spalten in Abbildung 6.10 zeigen, dass die Bewegung (und damit ihre SD) in $\omega=0^\circ$ -Momenten deutlich stärker als bei $\pm 45^\circ$ ist. In Abschnitt 6.6.3 wurde diskutiert, dass die Gyroskopdaten notwendig sind, um positive von negativen ω -Werten zu unterscheiden. Daher wird das SD Merkmale der Gyroskopdaten ebenfalls benötigt.

Merkmal 3 - Korrelation der Y- und der Z-Achse, $corr_{yz}$

Wie in Abschnitt 6.6.3 erörtert wurde, gibt die Korrelation zwischen der Y- und der Z-Achse $corr_{yz}$ einen Einblick darüber, wie weit der Kopf um die Gierachse gedreht ist. Die Werte der X-Achse sind mehr oder weniger irrelevant. In $\omega=0^\circ$ -Momenten ist die Kraft hauptsächlich auf der Z-Achse vorhanden, während sie in $\omega=\pm 45^\circ$ -Momenten auf die Y- und die Z-Achsen verteilt wird. Zur Verwendung als Merkmal wird daher die Korrelation berechnet:

$$\kappa = \tan^{-1}\left(\frac{Z}{Y}\right). \quad (6.3)$$

Aufgrund der Ausrichtung der Koordinatensysteme befindet sich κ in $[0^\circ, 180^\circ]$, da Benutzer den Kopf nicht in die Richtung ihres Rückens drehen können. Der Klassifikator kann ein $\omega=0^\circ$ -Moment erkennen, wenn $\kappa=90^\circ$ ist, mit einer Toleranz von $\pm 20^\circ$, da die meisten Menschen eine Drift des Gier-Winkels unter 20° nicht bemerken [416].

Merkmal 4 - Hauptkomponentenanalyse, pca

Es wird eine Hauptkomponentenanalyse pca [198] berechnet und der stärkste Eigenvektor verwendet, um das Datenfenster auf einen einzelnen Wert abzubilden, der als viertes Merkmal verwendet wird. Alle Eigenvektoren zusammen beschreiben die zeitabhängigen Varianzen zwischen allen Werten, die innerhalb des Fensters existieren, d. h. sie beschreiben, wie das Signal schwankt. Z. B. gibt es für die $\omega=-45^\circ$ -Momente aus Abbildung 6.10(a) Eigenvektoren, die den leichten Knick im X-Wert der Beschleunigung (bei ungefähr 750 ms) in Bezug auf die Signalspitze (bei ungefähr 500 ms) beschreiben.

Das Schiebefenster wird als ein 200-dimensionaler Vektor betrachtet, auf den eine Singularwertzerlegung (engl. Singular Value Decomposition, SVD) angewendet wird, um 200 Eigenvektoren und Eigenwerte abzuleiten, die beschreiben, wie die Werte im Fenster

verteilt sind. Da die Eingangssignale kontinuierlich sind, ergeben sich Fenster, die ähnliche Eigenvektoren (mit ähnlichen Eigenwerten) enthalten. Für jedes Fenster wird daher ein Histogramm über alle Eigenvektoren (multipliziert mit dem Datenvektor und skaliert mit dem Eigenwert, um einen einzelnen Wert zu erhalten) erstellt und ein Binning-Algorithmus findet darin aussagekräftige Cluster von Eigenvektoren. Dann wird die Gruppe mit dem größten Wert (der höchsten Varianz / Informationsdichte) als *pca*-Merkmal verwendet.

In einer Vorstudie im Rahmen der Dissertation wurde festgestellt, dass für die vorliegende Datengröße der Satz von 12 (einer pro Eingabestream) stärksten Histogrammspitzen eine gute Grundlage bietet, um $\omega=?$ - und $\omega=0^\circ$ -Momente von anderen ω -Winkeln (außerhalb der validen ω -Intervalle) zu trennen. Darüber hinaus kann *pca* auch negative von positiven ω -Werten unterscheiden, da sowohl die Gyroskop- als auch die Beschleunigungsmesserdaten signifikante Signalschwankungen aufweisen, die diese Bewegungen eindeutig beschreiben.

Zusammenfassung: Aus den 4·3-Signalströmen werden insgesamt $4 \cdot 3 \cdot 3 = 36$ Datenströme extrahiert (μ , *SD* und *pca*) plus 4 Datenströme für $corr_{yz}$, um die Achsen zu kombinieren. In Abschnitt 6.7.2 wird die Laufzeit im Hinblick auf die Wichtigkeit für die $36+4=40$ Merkmalsströme (engl. feature stream) bewertet und gezeigt, dass sie in Kombination die ω -Winkel gut genug trennen, um die $\omega=0^\circ$ -Momente zu erkennen, die benötigt werden, um die Drift der VR-Ansicht zu reduzieren. Andere Forschungen verwenden ebenfalls ähnliche Merkmalssätze. Sie sind ein guter Kompromiss zwischen Modellkomplexität, Rechenzeit und Klassifikationsgenauigkeit [290, 333].

6.6.5 Immersive Anpassung der VR Ansicht

Wenn $\omega=0^\circ$ -Momente lange genug beobachtet werden, um \vec{m} genau zu bestimmen, kann die aktuelle Drift ψ' abgeleitet werden. In einer Entfernung von 2 m und einem Positionsfehler von 0.10 m kann \vec{m} normalerweise um höchstens 2.86° falsch sein. Jedoch kann $\vec{v}=\vec{m}$ nicht einfach gesetzt werden, da dies zu sichtbaren Sprüngen in der Reihenfolge der visualisierten VR-Bilder führt, insbesondere wenn Benutzer weiterlaufen. Die Benutzer bemerken dann häufige Anpassungen der Ansicht, die die Immersion und Präsenz des VR-Erlebnisses stören. Um \vec{v} so anzupassen, dass es sich für den Benutzer natürlich anfühlt, wird

6 Orientierungsschätzung

eine lineare Interpolation [402] angewendet, die den geschätzten Gier-Orientierungsfehler ψ' schrittweise unbemerkt auf die aktuelle Orientierung \vec{v} der Ansicht anwendet. Nach Steinicke et al. [416] bemerkt ein VR-Benutzer keine kleinen Abweichungen zwischen der Ausrichtung der virtuellen und der physischen Ansicht. Daher wird also von der Startorientierung \vec{v} zur Endorientierung $\vec{v} + \psi'$ durch einen kleinen und immersiven Teil der Drift ω_{imm} zwischen zwei beliebigen aufeinanderfolgende Zeitrahmen interpoliert und ω_{imm} von ψ' subtrahiert:

$$\psi' \leftarrow \psi' - |\omega_{imm}|, \quad (6.4)$$

bis ψ' Null ist. Die Ansicht wird dann wie folgt angepasst:

$$\vec{v} \leftarrow (\omega_{imm} \cdot \vec{v}). \quad (6.5)$$

Zu beachten ist, dass die Ansicht nur in Bezug auf die Gierachse angepasst werden muss, da die Nick- und Rollachse fehlerfrei sind.

Aus Vorstudien im Rahmen der Dissertation ist bekannt, dass ein optimales unbemerktes Eintauchen (Immersion und Präsenz) erreicht wird, wenn (1) die Ansicht nur in Richtung der aktuellen Kopfdrehung angepasst wird, d. h. die Kopfdrehung leicht übertreibt, und (2) wenn die Anpassung nur durchgeführt wird, während der Benutzer selbst eine Gierrotation durchführt, da ein Benutzer die Anpassung der Ansicht am wahrscheinlichsten in stillen Momenten, m.a.W. wenn er/sie still steht und den Kopf nicht bewegt, bemerkt. Dies steht im Einklang mit den Ergebnissen in [415], wonach VR-Benutzer keine Übertreibungen / Dämpfungen von Rotationen um bis zu $+32\% / -25\%$ bemerken. Abschnitt 6.8.4 beschreibt, wie der kleine immersive Teil ω_{imm} ausgewählt wird: Je höher ω_{imm} ist, desto schneller kann die akkumulierte Drift eliminiert werden. Es gibt aber auch eine Obergrenze, oberhalb derer Benutzer die Ansichtsanpassung bemerken.

6.7 Experiment

Nach dem die Messinfrastruktur und System- und Methodenkonfiguration in Abschnitt 6.7.1 skizziert werden, wird in Abschnitt 6.7.2 der Datensatz beschrieben. Abschnitt 6.7.3 beschreibt die Konfiguration der untersuchten Klassifikatoren.

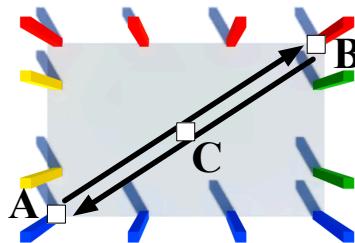


Abbildung 6.11: Draufsicht des VR-Szenarios zur Datenakquise. Benutzer bewegen sich zwischen A und B oder befinden sich bei C (Mitte). Der Abstand zwischen A und B beträgt 50 m.

6.7.1 Messinfrastruktur und -szenario

Messinfrastruktur

Auf einer Verfolgungs-Fläche von circa 40×35 m verwenden alle Experimente ein Samsung Galaxy Note 4-Mobiltelefon (Android 6.0.1, Qualcomm Snapdragon 805-CPU, 3 GB RAM) mit $6DOF$ IMU-Sensor von InvenSense (MPU-6500) an einem Samsung Gear VR HMD (SM-R320 ausgestattet mit einem BOSCH $6DOF$ IMU-Sensor BNO-055). Die IMU misst Beschleunigungen innerhalb von $[\pm 2, \pm 16] g$ und Gyroskoprotationen innerhalb von $[\pm 250, \pm 2000]^\circ/s$. Die Android-Sensor-API beschränkt alle Sensorwerte auf maximal 200 Hz. Diese maximale Abtastrate deckt den Frequenzbereich der menschlichen Bewegung ab. Obwohl eine Abtastrate von 40 Hz ausreichen würde, um das Nyquist-Shannon-Theorem zu erfüllen, werden 200 Hz verwendet, um unvorhersehbaren Leistungszustandumschaltungen (Stromsparplänen des Betriebssystems) und etwas Zeitjitter, die durch das Betriebssystem und die JVM (engl. Java Virtual Machine) verursacht werden, vorzubeugen. Die Abtastrate von 200 Hz stabilisiert auch die IIR-Filter. Die Filterparameter und Annahmen, die in Abschnitt 6.6 getroffen werden, entsprechen den Spezifikationen und Einschränkungen der hier verwendeten Hard- und Software.

Zusätzlich zu den IMU-Daten werden für das Training und die Auswertung der Klassifikatoren in Abschnitt 6.7.2 hochpräzise Gier-Orientierungsmessungen benötigen, um die IMU-Daten zu kennzeichnen. Solche Referenzen werden von einem Nikon IGPS-System, einem

6 Orientierungsschätzung

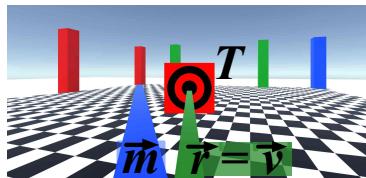


Abbildung 6.12: Ich-Ansicht des VR-Szenarios zur Datenakquise. Das Ziel T hilft dabei, einen gewünschten ω -Winkel beizubehalten. Blaue Linie = Gehweg, grüne Linie = Blickrichtung.

optischen laserbasierten Tracking-System mit einer durchschnittlichen vertikalen / horizontalen Genauigkeit von weniger als 10 mm bei 20 Hz bereitgestellt. Ein am Kopf montiertes Gerät wird verwendet, das zwei lokalisierbare Objekte in einem Abstand von 0.50 m trägt, aus denen die absolute Gier-Orientierung berechnet wird. Das iGPS-System wird auch verwendet, um die Gier-Orientierung (neu) zu kalibrieren und die Drift vor / nach den Studien zu messen.

Messszenario

Abbildung 6.11 ist eine Vogelperspektive auf das VR-Messszenario der Messkampagne. In allen Experimenten wird die VR-Szene so rechnerisch effizient wie möglich gestaltet, um eine konstante Bildrate von 60 Hz zu erreichen. Abbildung 6.12 zeigt die entsprechende Ego-Perspektive mit visuellen Elementen, die die Teilnehmer der Experimente anleiten. Die blaue Linie \vec{m} am Boden führt immer zur roten Säule (B) oder zur blauen Säule (A). Ein Ziel T und eine grüne Linie vom Benutzer in Richtung T bewegen sich mit dem Benutzer und helfen dabei, einen gewünschten ω -Winkel des Kopfes beizubehalten. Das Ziel T wird gemäß dem in einem Experiment verwendeten ω -Winkel platziert. Benutzer werden gebeten, auf natürliche Weise zu gehen, aber ihren Kopf so zu gieren, dass dieser immer mit einem beliebigen Pitch und Roll auf T gerichtet ist. Hierbei wird die grüne Linie verwendet, die sich immer in der Mitte des Bildschirms befindet.

6.7.2 Datensatz

Um die Merkmalsauswahl aus Abschnitt 6.6.3 und die Klassifikatorleistung zu bewerten, wurden Daten mit einer Gruppe von

34 Probanden gesammelt (Durchschnittsalter 23.16 [18, 36] Jahre; Durchschnittsgröße 1.74 [1.49 - 1.81]m; 20 Männer, 14 Frauen; niemand behindert). Die Probanden gingen 10 Mal natürlich von A nach B auf dem Weg \vec{m} und zurück in der in Abbildung 6.11 gezeigten VR-Umgebung. Die Teilnehmer haben sich vorab mit dem Aufbau und dem Zweck der Messungen vertraut gemacht. Während die IMU-Daten vom HMD gesammelt wurden, wurden auch \vec{r} mit dem iGPS-System gemessen, um genaue Gier-Orientierungen zu erhalten, d. h. eine Kennzeichnung der IMU-Messungen. Die Benutzer wurden gebeten, natürlich und entspannt mit normaler Geschwindigkeit zu gehen (durchschnittlich 0.87 m/s, min.: 0.58 m/s, max.: 1.19 m/s, SD: 0.18 m/s) und wurden gebeten ihre Köpfe mithilfe des Ziels T und der grünen Hilfslinie auf einem festen ω -Winkel [-45°; -30°; -15°; 0°; +15°; +30°; +45°] zu halten. Die VR-Anwendung verwendete auch ein Sprachfeedback, das einen Benutzer alarmierte, wenn er das Ziel- ω um mehr als 5° unterbot oder überschritt. Um beim Gehen eine nahezu natürliche Kopfhaltung zu erreichen, wurde keine starre Kopfnieigung und -rolle erzwungen. Die gesammelten Daten zeigen die typischen Kompensationsbewegungen der Köpfe entlang der Trajektorien der Benutzer (siehe Abschnitt 6.6) und die Daten enthalten auch natürliches Rauschen und Jitter um das Ziel- ω .

In einem Nachbearbeitungsschritt wurden die Daten von den Momenten bereinigt, in denen sich ω außerhalb der Zielzone befand, als auch von den Momenten, in denen sich die Benutzer bei B oder A umgedreht haben. Es wurden ungefähr 8h Bewegungsdaten aufgezeichnet. Gemäß Abschnitt 6.6.3 wurden diese vorverarbeitet und die Merkmalsströme extrahiert. Um eine Überabtastung von Minderheiten (engl. minority oversampling) zu vermeiden, wurden einige (zufällige) Stichproben aus den Daten entfernt, sodass jede ω -Klasse die gleiche Anzahl von Stichproben aufweist.

6.7.3 Konfiguration der Klassifikatoren

Im Rahmen der Dissertation wird die Leistung von drei Klassifikatoren des maschinellen Lernens untersucht. Eine Rastersuche⁸ wurde

⁸SVC: $k \in \{\text{linear}, \text{polynomial}, \text{gaussian}, \text{rbf}\}$; $\text{Cost} = \mathbf{19} \in \{10^{-5} : 1 : 10^5\}$; $\Gamma = \mathbf{5} \in \{10^{-5} : 1 : 10^5\}$; $p = \mathbf{3} \in \{1 : 1 : 10\}$. DT: $c \in \{\text{gini}, \text{entropy}\}$; $\text{depth}_{\max} = \mathbf{100} \in \{1 : 10 : 1000\}$; $\text{leaf}_{\min} \in \{\mathbf{1} : 1 : 15\}$; $\text{split}_{\min} \geq \mathbf{10} \in \{1 : 1 : 100\}$. Die beste Konfiguration ist **fett** gedruckt.

6 Orientierungsschätzung

Tabelle 6.2: Ergebnisse einer 10-fachen Kreuzvalidierung (engl. 10-fold cross-validation) in % für drei Klassifikatoren: SVM-Klassifikator in den oberen Zahlen der Zellen, Entscheidungsbaum (engl. Decision Tree, DT) in der Mitte, k-NN (engl. k-Nearest-Neighbor, dt. Nächste-Nachbarn-Klassifikation) unten. Echte ω -Klassen (Ref.) entlang der vertikalen Achse, die Klassifizierung erfolgt horizontal. Korrekte Klassifizierungen werden in **fett** und Fehlklassifizierungen in normaler Schriftart dargestellt. Die schattierte Zelle zeigt die Erfolgsrate, d. h. die Rate korrekter Klassifizierungen von $\omega=0^\circ$ -Momenten.

	SVM	DT	k-NN	0	0	0	0	0	0	$\omega=-45^\circ$
9	83	14	15	0	4	1	1	1	0	$\omega=-45^\circ$
13	78	15	26	1	4	2	1	1	0	$\omega=-30^\circ$
10	66	7	13	5	4	1	1	0	0	$\omega=-30^\circ$
1	9	80	5	4	1	1	0	0	0	$\omega=-15^\circ$
1	13	67	8	7	4	2	1	0	0	$\omega=-15^\circ$
1	10	67	13	5	4	1	0	0	0	$\omega=-15^\circ$
1	1	4	77	3	11	3	1	0	0	$\omega=0^\circ$
1	1	7	60	5	19	8	1	0	0	$\omega=0^\circ$
1	1	5	64	3	24	3	0	1	0	$\omega=0^\circ$
1	1	4	4	86	4	1	0	0	0	$\omega=+15^\circ$
2	0	7	7	72	10	3	0	0	0	$\omega=+15^\circ$
1	0	7	9	71	11	0	0	0	0	$\omega=+15^\circ$
1	1	1	10	6	74	9	0	0	0	$\omega=+30^\circ$
0	0	1	13	10	61	16	1	0	0	$\omega=+30^\circ$
0	0	0	11	9	62	8	0	1	0	$\omega=+30^\circ$
0	0	2	3	2	10	76	7	0	0	$\omega=+30^\circ$
1	1	1	6	3	12	67	10	0	0	$\omega=+30^\circ$
0	0	1	3	3	23	65	5	1	0	$\omega=+30^\circ$
1	1	1	0	0	1	11	86	0	0	$\omega=+45^\circ$
1	1	1	1	1	1	16	79	0	0	$\omega=+45^\circ$
0	0	1	1	1	2	21	74	0	0	$\omega=+45^\circ$
1	1	0	0	0	1	0	0	99	0	$\omega=?$
0	0	0	0	0	1	1	0	99	0	$\omega=?$
0	0	1	0	0	0	0	0	99	0	$\omega=?$

verwendet, um die Parameter der Klassifikatoren auf den Trainings- und Validierungsdaten zu optimieren.

Stützvektormaschine (SVM)

Erstens wird eine Stützvektormaschine (SVM) [87] mit einer kubischen Kernelfunktion untersucht:

$$K(x_q, x_i) = (1 + \gamma \cdot x_q^T x_i)^d. \quad (6.6)$$

Die Rastersuche liefert die beste Konfiguration für SVC: mit einem Kernel k mit Polynomfunktion, den Parametern Kosten=19 und $\Gamma=5$ und einer Polynomordnung $p=3$. Da eine Klassifizierung für mehrere Klassen benötigt wird, wird eine Eine-gegen-Alle (engl. One-vs-All, OvA) SVM Variante verwendet.

Entscheidungsbaum

Zweitens wird ein (Klassifikations)Entscheidungsbaum- (engl. Decision Tree, DT)-Modell [378] untersucht. Die Rastersuche liefert die beste Konfiguration für DT: nicht mehr als 100 Teilungen, eine minimale Blattzahl von 1, der Gini-Diversity-Index I_G als Teilungskriterium und anschließender Schnitt mit mehr als 10 Eltern pro Blatt.

k-Nächster-Nachbar

Drittens wird ein kubischer k-Nächster-Nachbar (engl. k-Nearest Neighbor, kNN) [89] untersucht. Die Rastersuche liefert die beste Konfiguration für kNN: Distanzparameter $k=3$, Distanzfunktion $(X \times Y)^n \rightarrow (X \rightarrow Y)$ und kubische Minkowski-Entfernungsmetrik.

6.8 Auswertung

Abschnitt 6.8.1 wertet die Klassifikationsverfahren aus. Anschließend wertet Abschnitt 6.8.2 den Einfluss der ausgewählten Merkmale auf die Klassifizierungsgenauigkeit aus. Abschnitt 6.8.3 evaluiert die Optimierung der Klassifikation. Schließlich bewertet Abschnitt 6.8.4 die immersive Anpassung der VR-Ansicht und Abschnitt 6.8.5 bewertet die Anwendbarkeit des Verfahrens in einer realen VR-Anwendung.

6.8.1 Auswertung der Klassifikatoren

Mit jedem Klassifizierer wird eine 10-fache Kreuzvalidierung für die mit $4 \cdot 3 \cdot 3 + 4 = 40$ gekennzeichneten Merkmalsströme durchgeführt. Die Validierung teilt die Daten in 10 gleich große Teilstichproben auf und verwendet neun Teilstichproben für das Training des Klassifikators und die verbleibenden für die Validierung. Dies wird 10 Mal wiederholt, wobei jede Unterprobe einmal der Validierungssatz ist.

Tabelle 6.2 enthält die gemittelten Ergebnisse. Alle Klassifikatoren können natürlich $\omega=90^\circ$ -Momente fast perfekt erkennen (Zelle unten rechts). Wenn die Teilnehmer gehen, liefert der SVM-Klassifikator die höchste korrekte Klassifizierungsrate und damit das höchste Vertrauen (Zellen entlang der Diagonale) für alle ω -Winkel. Alle Klassifikatoren trennen $\omega=0^\circ$ -Momente deutlich von anderen ω -Bereichen. Die angrenzenden Bereiche sind die schwierigsten. Der SVM-Klassifizierer klassifiziert die meisten (86%) der $\omega=0^\circ$ -Momente korrekt. Die Klassifizierungsrate, die *Erfolgsrate* eines Klassifikators, wird im Folgenden mit $\omega=0^\circ$ bezeichnet, weil diese Momente verwendet werden, um die Drift anzupassen und die absolute Orientierung zu bestimmen.

Um alle 40 Merkmale aller Datenströme zu verarbeiten, benötigt der SVM-Klassifikator 259 μ s (pro Aktualisierung), während DT und k-NN 29 μ s bzw. 109 μ s benötigen. Da der SVM-Klassifikator den DT-Klassifikator deutlich übertrifft, lohnt sich der Aufwand.

6.8.2 Auswertung der Merkmale

Die Klassifizierer werden sowohl mit unterschiedlichen Teilmengen der Merkmale (μ , SD , $corr_{yz}$ und pca) als auch mit verschiedenen Teilmengen der Eingabestreams (acc^{raw} , gyr^{raw} , acc^{SG} , gyr^{SG} , acc^{IIR} , und acc_{lin}^{IIR}) trainiert und verwendet. Die untersuchten Kombinationen werden in Tabelle 6.3 aufgelistet. Unabhängig davon, für welche Kombination von Merkmalen und Datenströmen die drei Klassifikatoren verwendet werden, übertrifft SVM die beiden anderen deutlich. Daher wird im Folgenden nur die Ergebnisse des SVM-Klassifikators diskutiert. Tabelle 6.3 zeigt die Erfolgsraten. Je mehr Merkmale und Eingabeströme verwendet werden, desto besser wird die Erfolgsrate (bis zu einem Maximum von 86%). Die letzte Spalte zeigt die durchschnittliche CPU-Zeit, die eine einzelne Klassifizierung für alle

Tabelle 6.3: Erfolgsraten (Rate korrekter Klassifizierungen von $\omega=0^\circ$ -Momenten) des SVM-Klassifikators in % mit einer Teilmenge der Merkmale (vertikal) und einer Teilmenge gefilterter Eingabedatenströme (horizontal). Das Symbol – zeigt ein Merkmal oder einen Datenstrom an, der ausgelassen wird. Das **fett 86** befindet sich ebenfalls in Tabelle 6.2. Unterstrichene Zahlen werden diskutiert.

		Ströme												CPU (alle Datenströme) [ms]
		acc^{raw}	–	acc^{SG}	–	acc^{SQ}	–	acc^{IR}	–	acc_{lin}^{SG}	–	acc_{lin}^{IR}	–	
Merkmale														
μ	–	–	–	–	66	49	6	66	47	68	34	61	61	62
–	$ SD $	–	–	–	69	60	68	63	75	64	69	57	56	58
–	–	$ corr_{yz} $	–	–	14	6	67	14	14	6	9	65	73	69
–	–	–	pca	–	71	12	66	64	12	65	11	64	68	70
μ	$ SD $	–	–	–	66	54	60	67	39	67	40	77	78	77
μ	$ SD $	$ corr_{yz} $	–	–	73	50	76	71	42	70	40	79	83	83
μ	$ SD $	–	pca	–	73	56	77	73	44	72	44	79	80	81
–	–	$ corr_{yz} $	pca	–	70	13	74	70	25	68	19	72	73	75
μ	$ SD $	$ corr_{yz} $	pca	–	73	52	78	73	42	72	41	81	84	83
														86
														259

6 Orientierungsschätzung

ausgewählten Datenströme und Merkmale benötigt. Je mehr Daten verarbeitet werden, desto länger dauert eine Klassifizierung (bis zu 259ms).

Wenn nur die Rohsignale zur Klassifizierung von ω -Momenten verwendet werden, ist die am besten erreichbare Erfolgsrate 78%. Es ist daher besser, stattdessen SG-gefilterte Datenströme zu verwenden (81%). Die sensorspezifischen Informationen (Rauschen, Reaktionszeit, Min-Max-Bereich), die durch die Filterung verloren gehen, schaden nicht (siehe SD unter acc^{raw} 69% gegenüber acc^{SG} 63%), da die Kombination mehrerer Merkmale mit gefilterten Eingabeströmen die Klassifizierung anhand von Rohdatenströmen deutlich übertrifft (selbst in allen möglichen Kombinationen). Darum behandeln die Box in Tabelle 6.3 und die folgende Diskussion nur gefilterte Datenströme.

Um die Auswirkung der Merkmale auf die Erfolgsrate des Klassifikators zu bewerten, werden die Effekte der Verwendung einzelner Merkmale diskutiert. Die *pca*-Funktion allein bietet die höchste Erfolgsrate (71%). Die Funktion $corr_{yz}$ extrahiert ähnliche Informationen aus den Datenströmen und ist ebenso genau mit ähnlichen Rechenkosten. Die Funktionen μ und SD sind schlechter, unabhängig davon, wie viele Eingabeströme verwendet werden. Dies liegt daran, dass μ nur den aktuellen Punkt in einem Gangzyklus beschreibt und SD nur Energieinformationen liefert, d. h. nur $\omega=?$ Von $\omega=0^\circ$ trennt, da sich ihre Abweichungen am meisten unterscheiden. Weitere Eingabeströme wiederholen diese Informationen und erhöhen nicht die Erfolgsrate.

Die Kombination mehrerer Merkmale führt zu höheren Erfolgsraten. Während eine Kombination von μ und SD die Klassifizierung bereits verstärkt, hilft ein zusätzliches $corr_{yz}$ oder *pca* noch mehr, da diese Merkmale unterschiedliche Eigenschaften beschreiben, z. B. Bewegungsrichtung oder ein abstraktes Signalmuster. Das ist auch der Grund, warum eine Kombination aus $corr_{yz}$ und *pca* allein schlechter abschneidet: Der Status und der Typ der Bewegung stehen dem Klassifizierer nicht zur Verfügung. Der vollständige Funktionsumfang aller Eingabeströme ergibt eine Erfolgsrate von **86%**.

6.8.3 Optimierung der Klassifikationsgenauigkeit

Für die Eliminierung der Drift und Anpassung der VR-Ansicht sind die Klassierungsergebnisse sogar besser als *nur* die Erfolgsrate. Da

Menschen eine Drift von 20° tolerieren, ohne es zu bemerken, sind auch Fehlklassifizierungen in eine benachbarte ω -Klasse ($\pm\omega=15^\circ$) tolerierbar. Dann liefert der SVM-Klassifikator in $86+3+6=95\%$ der Fälle ein korrektes Ergebnis. Nach dieser größeren Klassifizierung werden aufeinanderfolgende Ergebnisse mit den Gyroskopsignalen verglichen. Da es eine Korrespondenz geben muss, da Menschen zwischen den Messungen nicht zu viel den Kopf drehen können, können alle verbleibende Fehlklassifizierungen behoben werden. Abschnitt 6.8.5 wertet eine Anwendung des Klassifikators in einem realen Anwendungsfall aus.

6.8.4 Auswertung der Anpassung der VR-Ansicht

Die Anpassung der Sicht in dynamischen Situationen wurde im Zusammenhang mit Umgeleitetem Gehen [416] ausführlich untersucht, d. h. während sich ein Benutzer geht, joggt oder rennt. Zwischen zwei aufeinanderfolgenden Bildern kann die virtuelle Bewegung / Drehung des Benutzers im VR-Display um +30% oder -20% übertrieben oder abgeschwächt werden, ohne dass der Benutzer dies bemerkt. Für dynamische Szenarien mit einem sich bewegenden Benutzer werden diese Ergebnisse verwendet und $\omega_{imm}=0.30 \cdot d$ gewählt, um die Drift anzupassen, während sich der Kopf des Benutzers um d° dreht.

Während dieser Dissertation gab es keine Forschungsergebnisse, die sich auf statische Szenarien ohne Bewegungen konzentrieren. Dieses Szenario ist allerdings erforderlich, um bspw. eine Drift zu korrigieren, wenn Benutzer stillstehen um sich auf ein Ziel zu konzentrieren. Ohne eine Körperbewegung gibt es weitaus weniger Kopfbewegungen, mit denen eine Driftkorrektur vor dem Benutzer versteckt werden kann. Im Stehen reagieren Benutzer empfindlicher auf visuelle Ungenauigkeiten. Das statische Szenario ist daher der schlechteste Fall für die Auswahl eines ω_{imm} für die Ansichtsanpassung, die unter dem Erkennungsschwellenwert bleibt.

Für eine solche Studie wurde eine Gruppe von 52 (anderen) Probanden verwendet (Durchschnittsalter 25.82 [19, 61] Jahre; Durchschnittsgröße 1.72 [1.51 - 1.87]m; 34 männlich, 18 weiblich; niemand ist eingeschränkt oder behindert). Den Teilnehmern wurden der Prozess und der Zweck des Tests im Voraus erklärt und sie wurden gebeten, signifikante Jitter und / oder Sprünge der Kameraansicht anzukündigen, sobald sie sie bemerkten. Die Teilnehmer standen stationär an

6 Orientierungsschätzung

Position C in der in Abbildung 6.11 gezeigten virtuellen Szene. Anfangs waren ihre Köpfe nach Norden gerichtet (rote Säulen). Es wurde eine anfängliche Drift von $\psi' = 90^\circ$ angewendet, d. h. die virtuelle Ansicht der Benutzer \vec{v} war nach Osten gerichtet (grüne Säulen).

Alle Benutzer mussten die folgenden zwei Aufgaben ausführen:
(a) Sie wurden gebeten, ihre Köpfe währenddessen ruhig zu halten. Die Ansichtsanpassung wurde durchgeführt, bis die Benutzer die geringste Störung erkannt haben. Dann wurde der aktuelle ω_{imm} Wert abgespeichert. In der anschließenden zweiten Studie (b) drehten die Benutzer wiederholt auf natürliche Weise den Kopf von links nach rechts und zurück, d. h. von ungefähr -90° bis ungefähr $+90^\circ$ und zurück, während wieder die Ansichtsanpassung durchgeführt wurde. Die durchschnittliche Kopfdrehrate betrug $37^\circ/s$. Nach wie vor haben die Benutzer den Test abgebrochen, sobald sie eine Störung erkannt haben. Wieder wurde das ω_{imm} dieses Augenblicks gespeichert. In beiden Fällen wurde von einem anfänglichen $\omega_{imm} = 0^\circ$ ausgegangen und ω_{imm} langsam um $0.1^\circ/s$ bzw. $0.0017^\circ/frame$ erhöht. Die VR-Szene wurde mit einer konstanten Bildrate von 60 Hz visualisiert.

Insgesamt bemerkten 89% der Benutzer in Studie (a) die Driftkorrekturen bereits bei einem kleinen $\omega_{imm} = 0.015^\circ/frame$ (unter $1^\circ/s$). Die am wenigsten sensiblen Benutzer entdeckten Störungen bei $\omega_{imm} = 0.038^\circ/frame$ ($2.28^\circ/s$). Als die Benutzer in Studie (b) den Kopf drehten, bemerkten die meisten von ihnen (92%) die Ansichtsanpassung erst viel später, als ω_{imm} bereits $0.08^\circ/frame$ ($4.8^\circ/s$) erreichte. Mit $\omega_{imm} = 0.11^\circ/frame$ ($6.6^\circ/s$) bemerkten alle Teilnehmer Störungen.

Zusammenfassung: Das Ergebnis ist, dass die Ansicht jederzeit angepasst werden kann, indem ein nicht wahrnehmbares $\omega_{imm} \leq 0.015^\circ/frame$ ($0.9^\circ/s$) verwendet wird. Benutzer bemerken dann die Ansichtsanpassung nicht, selbst wenn sie stillstehen. Wenn sie sich bewegen, kann die Obergrenze unter $\omega_{imm} = 0.08^\circ/frame$ ($4.8^\circ/s$) gesetzt werden, um nicht wahrnehmbare Ansichtsanpassungen zu implementieren. In einer Vorstudie wurde festgestellt, dass es in der Praxis keine so großen Abweichungen gibt, da selbst die Sensoren, die in kostengünstigen Mobiltelefonen verbaut werden, nur um etwa $0.0013^\circ/frame$ ($\leq 0.078^\circ/s$) driften, d. h. um 20° nach 5 Minuten. Selbst wenn die $\omega = 0^\circ$ -Momente nur selten in der Bewegung der Benutzer auftreten, reicht daher ein kleineres ω_{imm} aus.



Abbildung 6.13: VR-Szenario des Bewertungsexperiments, das die Eignung für den täglichen Gebrauch belegt. [363]

6.8.5 Anwendbarkeit für reale Anwendungsfälle

Die Fähigkeit, $\omega=0^\circ$ -Momente zu erkennen und die Ansicht anschließend unbemerkt anzupassen, um die Drift zu bereinigen, ist nur relevant, wenn $\omega=0^\circ$ -Momente oft genug auftreten. Daher wird untersucht wie oft und wie lange Benutzer in einer frei begehbaren, großformatigen Mehrbenutzer-VR in ihre Blickrichtung gehen.

79 (andere) Probanden wurden befragt (Durchschnittsalter 32.34[18, 63] Jahre; Durchschnittsgröße 1.71[1.49 - 1.96]m; 43 Männer, 36 Frauen; niemand ist behindert), um sich in einem virtuellen Museum frei zu bewegen, das sechs verschiedene Exponate von Dinosauriern in Originalgröße bietet (siehe Abbildung 6.13) und das auf einer Fläche von $45m \times 35$ m erkundet werden kann [363].

Dieselben Samsung GearVR-HMDs, aber jetzt mit einer Mischung aus Samsung Galaxy S7- und S8-Mobiltelefonen, wurden verwendet, um die Sensorströme (acc_{raw} und gyr_{raw}) und ein hochfrequenzba siertes Tracking-System aufzuzeichnen ($CEP_{95}=0.20$ m bei 20 Hz), das die absoluten Positionen der Köpfe (ähnlich wie Tiemann et al. [434]) während der gesamten Erfahrung verfolgt. Da Benutzer eine Abweichung von 20° [416] tolerieren, ohne es zu bemerken, und es mindestens 5 Minuten dauert, bis sich die Abweichung auf etwa 20°

6 Orientierungsschätzung

($4.94^\circ/\text{min}$) ansammelt wurde der Gier-Winkel vor den Experimenten manuell kalibriert und nur die ersten drei Minuten der aufgezeichneten Daten (pro Person) verwendet, um die (fast driftfreie) Schätzung der Kopforientierung als Referenz zu verwenden.

Es wurde keine feste Kopforientierung erzwungen. Die Teilnehmer wurden gebeten, das Museum so zu erkunden, als ob sie ein echtes Museum mit normaler und entspannter Bewegungsgeschwindigkeit besuchen würden. Die Benutzer wurden nur in die Studie und ihre Risiken eingeführt, waren sich jedoch des Studiendesigns und der Hypothesen nicht bewusst. [363] Im Nachgang wurden Segmente der Daten gelöscht, wenn Benutzer nicht auf einer geraden Linie gingen (erkannt durch die Positionsschätzung), und Segmente, die schnelle oder signifikante Änderungen der Kopfdrehung in der Gierausrichtung enthalten. Die verbleibenden Sequenzen wurden anhand eines aufgenommenen Draufsichtvideos überprüft, das die Ausrichtung und Bewegung von Kopf und Körper aller Benutzer während der Experimente zeigt.

Aus den resultierenden Sequenzen wurden die $\omega=0^\circ$ -Momente gezählt. Ein solcher Moment wurde nur gezählt, wenn er mindestens 1 m oder mindestens 2 s zu Fuß dauert. Im Durchschnitt führt jeder Teilnehmer 41 separate $\omega=0^\circ$ -Momente innerhalb des 3-Minuten-Intervalls (min: 36, max: 54, SD: 7.4) bei einer durchschnittlichen Gehgeschwindigkeit von 0.74 m/s durch (min: 0.56 m/s, max: 1.60 m/s, SD: 0.14 m/s) aus und legt eine durchschnittliche Entfernung von 1.18 m (min: 0.51 m, max: 2.27 m, SD: 0.37 m) zurück. Daher kann die Ansichtsanpassung alle 4.4 s ($=180\text{ s}/41$) einleiten. Es wurden auch andere ω -Momente gezählt, z. B. $\omega=+15^\circ$. Die angrenzenden Bereiche ($\omega=\pm 15^\circ$) traten weitaus seltener auf (12 Mal), und wenn Benutzer den Kopf so weit gedreht hatten, lagen ihre Durchschnittsgeschwindigkeiten bei 0.43 m/s und Sie legten im Durchschnitt nur Entferungen von 0.89 m (innerhalb von 2 s) zurück. Dies zeigt, dass Benutzer häufiger und länger in ihre Blickrichtung gehen. Erwartungsgemäß standen die Benutzer für ein Museum etwas mehr still, als sie sich bewegten. Durchschnittlich standen die Benutzer 245 Mal still ($\omega=0^\circ$ -Momente), während sie sich nur 208 Mal weiter bewegten ($\omega=?$ -Momente). Insgesamt gibt es genug $\omega=0^\circ$ -Momente für eine robuste Kursschätzung.

In weiteren Vorstudien wurden die Einschränkungen der Methode (siehe Diskussion der Einschränkungen in Abschnitt 6.9) in realen Anwendungen genauer untersucht. Abbildung 6.14 zeigt beispielhaft eine der Vorstudie, in der der Prozess der Kalibrierung des Gier-Winkels in einer typischen VR-Anwendung mit einem künstlichen Gierwinkelfehler von 90° untersucht wurde.

6.9 Diskussion

Dieser Abschnitt diskutiert mögliche Einschränkungen des Verfahrens. Zuerst werden Auswirkungen der Positionsgenauigkeit diskutiert. Dann wird der Einfluss der Sequenzlänge erörtert, bevor allgemeine Einschränkungen diskutiert werden.

Auswirkung der Positionsgenauigkeit

Die Transformation zwischen den Koordinatensystemen (Position und IMU) ist fehlerfrei möglich, da das absolute Koordinatensystem basierend auf den absoluten Positionen bekannt ist und der Fehler der Orientierung \vec{m} , die aus zwei konsekutiven Positionen (mit einem maximalen Positionsfehler von ± 0.20 m der ersten und zweiten Positionen) ermittelt wird, mit zunehmendem Abstand zwischen zwei Positionen linear abnimmt. Bei einer Distanz von 1.0 m zwischen zwei konsekutiven Positionen ergaben sich für \vec{m} ein durchschnittlicher Fehler von 11.3° und ein maximaler Fehler von 21.8° . Eine Distanz von 2.0 m lieferte einen durchschnittlichen Fehler von 5.7° und einen maximalen Fehler von 11.3° . Und eine durchschnittliche Distanz von 1.25 m pro Fenster führte zu einem durchschnittlichen Fehler von 2.6° und zu einem maximalen Fehler von 4.1° . Die Orientierung \vec{v} wird gegen die bekannte Orientierung \vec{m} (bzw. \vec{r}) bei $\omega=0^\circ$ (neu)kalibriert.

Somit ist klar, dass das Verfahren auch sehr ungenaue GPS-basierte absolute Positionen verwenden kann, wenn der Positionsvektor, der \vec{m} liefert, ausreichend groß gewählt wird, d. h. längere Positionsbeobachtungs-/Aufzeichnungsintervalle. Da der Positionsvektor den realen absoluten Gier-Winkel darstellt, muss sichergestellt werden, dass sein Fehler gering ist (z. B. liefert GPS mit einem $CEP_{95}=1.9$ m eine durchschnittliche Positionsgenauigkeit von 0.38 m nach einer zurückgelegten Strecke von 10 m).

6 Orientierungsschätzung

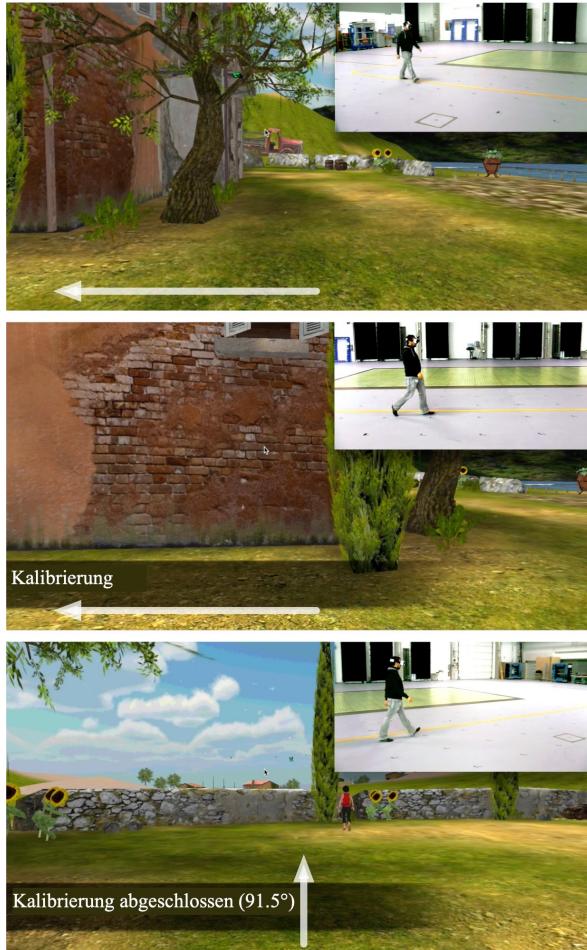


Abbildung 6.14: Kalibrierung des Gier-Winkels in einer VR-Anwendung mit einem künstlichen Gierwinkelfehler von 90° . Bilderfolge von oben nach unten: Der VR-Benutzer geht (\vec{m}) und schaut (\vec{r}) in der realen Welt geradeaus (Bildausschnitt oben rechts je Bild), während er sich aufgrund des Gierwinkelfehlers in der virtuellen Welt seitwärts bewegt. Der weiße Pfeil zeigt die Bewegungsrichtung des Bildes. Die kubische SVM löst die Kalibrierung aus (im mittleren Bild) und korrigiert die Ansicht um 91.5° (im unteren Bild). Der Korrekturfehler (1.5°) ergibt sich aus der ungenauen Schätzung der Bewegungsrichtung aufgrund ungenauer Positionen (\vec{m}). Für eine übersichtliche Darstellung wurde die VR-Ansicht nicht immersiv angepasst.

Einfluss der Sequenzlänge

Der Ansatz beruht auf der Verfügbarkeit der Sensordaten eines vollständigen Gangzyklus in einem Schiebefenster fester Größe sowie eines (ungenauen) absoluten Positionsvektors (durchschnittliche Fehler von 2.8° bei 1.25 m zurückgelegter Strecke pro Fenster). Dies stellt in der Praxis keine Einschränkung dar.

Erstens, obwohl Benutzer in VR langsamer laufen als in der realen Welt, reicht ein Schiebefenster von 1 s (200 Hz) aus, da es zwei Schritte enthält und einen Positionsvektor von etwa 1 m bei einem vernachlässigbaren Positionsfehler ($0.064 \text{ m} / \text{m}$ führt zu 3.7°) überspannt.

Zweitens zeigt die Bewertung in Abschnitt 6.8.5, dass in einem realen Anwendungsfall variable Bewegungsgeschwindigkeiten die Klassifizierungsgenauigkeit nicht beeinträchtigen.

Drittens, eine Fensterlänge von 200 Hz deckt sowohl das Laufen als auch das Rennen vollständig ab. Dennoch kann selbstverständlich für höhere Geschwindigkeiten ein noch größeres Fenster verwendet werden. Schließlich sind für relevante Fenstergrößen sowohl die Merkmalsextraktion als auch die Klassifizierung schnell genug, um in anderen Latenzen, wie z. B. der Funkübertragung, verborgen zu sein. Sowohl das SG- als auch das IIR-Filter liefern einen Signalstrom, der im Durchschnitt um 47 ms gegenüber dem absoluten Positionsstrom verzögert ist (min.: 27 ms, max.: 76 ms, SD: 6.3 ms, theoretisch $1/8$ der Schiebefenstergröße). Es wurde festgestellt, dass die Vorverarbeitung des hochfrequenzbasierten Positionsverfolgungssystems unter einer Funkverzögerung leidet und sich im Durchschnitt um 57 ms verspätet (min.: 32 ms; max.: 76 ms, SD: 5.9 ms). Somit wird die Gesamtverzögerung gemittelt und ist daher vernachlässigbar.

Viertens hilft ein Vorverarbeitungsschritt, die Klassifizierung nur auszuführen, wenn die Benutzer auf geraden Trajektorien gehen. Die minimale zurückgelegte Strecke beträgt $\geq 1 \text{ m}$ oder die Dauer beträgt $\geq 1 \text{ s}$. Die Verzögerung, die die SG- oder IIR-Filter zu den Signalströmen hinzufügen (Verzögerung von maximal 125 ms, d. h. $1/8$ eines Schiebefensters), muss nicht explizit behandelt werden, da sie einerseits mit der Verzögerung des Positionierungssystems herausgemittelt wird und andererseits werden nur die Momente $\omega=0^\circ$ mit hoher Vertraulichkeit

6 Orientierungsschätzung

(ungefähr 95%) verwendet, die sicherstellen, dass wirklich nur ω° -Momente gefunden werden.

Allgemeine Limitationen

Die Probanden besuchten das Museumserlebnis mit einer Dauer von 10 min, SD=0.54 min. Die Auswahl der Dauer basiert auf den Ergebnissen der Schätzung des Gier-Winkels von kostengünstigen Sensoren (z. B. Samsung GearVR mit Note 4), die um fast $5^{\circ}/\text{min}$ abweicht. So sammelt sich nach 5 min (in dem VR-Museum) eine quasi deterministische Gierdrift von etwa 20° an.

Die Genauigkeit der Trainingsdaten gefährdet die Klassifizierungsleistung und damit die Wirksamkeit des Ansatzes. Die Klassifizierung kann nur so genau sein wie die Daten, auf denen sie trainiert wird. Je vielfältiger die Daten sind, mit denen das Modell trainiert wird, z. B. mehr Teilnehmer, mehr Variationen und verschiedene Arten von Bewegungen, desto höher ist das Vertrauen der Klassifizierung zur Laufzeit. Das Sammeln von weiteren ausgeglichenen Daten und das (erneute) Trainieren des Klassifizierungsmodells zur Laufzeit verbessert daher die Effektivität des Klassifizierers. Aber auch ohne ein solches (Nach-)Training zeigt die Auswertung in Abschnitt 6.8.5, dass die Methode im täglichen Gebrauch gut funktioniert, selbst bei unbekannten Bewegungen unbekannter VR-Benutzer.

Zu beachten ist, dass es keine Einschränkung des Ansatzes ist, dass der Gier-Winkel nur auf $\omega=0^{\circ}$ -Momente geschätzt wird. Das in der Dissertation vorgeschlagene PDR Verfahren benötigt keine vollständige (Neu-)Kalibrierung der Pose, da Nick- und Rollbewegungen bereits durch Beschleunigungsmesser- und Gyroskopinformationen genau bestimmt werden. Die Kalibrierung der Orientierung ermöglicht die Stabilisierung der Ausrichtung der Trajektorie über lange Zeiträume und ermöglicht dabei Sensoren über unterschiedliche und hochdynamische Koordinatensysteme hinaus (Kopf und Rumpf in Sport und VR Anwendungen) zusammenzuführen.

Im Vergleich zu hochmodernen KF, die Schwierigkeiten haben, die nichtlineare Kopf-Körper-Pose über die Zeit korrekt zu schätzen, akzeptiert die Methode eine nicht wahrnehmbare Drift und lernt Momente, um verrauschte Sensordaten zu abstrahieren, die eine nicht-

lineare Bewegung darstellen, und lernt diese zur Laufzeit zuverlässig (wieder) zu erkennen und gelegentlich zur Korrektur akkumulierter Abweichungen verwenden können.

Darüber hinaus kann das Klassifizierungsverfahren leicht um andere Sensoren wie Barometer und Magnetometer erweitert werden. Außerdem kann die Methode durch Verfahren des tiefen Lernens ergänzt werden. Bspw. könnte eine Methode mit (B)LSTM-Zellen die ω -Momente vorhersagen ohne eine handgemachte Merkmalsanalyse zu benötigen. Eine solche Methode wurde im Rahmen der Dissertation veröffentlicht, um Gangphasen in einem kontinuierlichen IMU-Signal vorherzusagen [129]. Abschnitt 8.4.1 fasst diese Methode kurz zusammen.

6.10 Zusammenfassung

Dieses Kapitel zeigt, wie langfristig stabile absolute Orientierungen (auch mit Sensoren aus unterschiedlichen Koordinatensystemen, bspw. Kopf-Körper-Orientierungen) aus ungenauen Positionen und verrauschten Trägheitssensoren (bspw. am Kopf) abgeschätzt werden. In einem großen Raum und mit dieser Keine-Pose-Verfolgung bemerken die VR-Benutzer den maximalen Fehler von $\pm 15^\circ$ nicht, da die Schätzung verwendet wird, um die Drift des angezeigten VR-Bildes immersiv zu korrigieren.

Um dieses Ziel zu erreichen, wurden eine Reihe von verallgemeinerten Merkmalen vorgestellt, die aus gefilterten Sensordaten extrahiert werden können, die ein SVM-Klassifizierer (nach einem überwachten Training mit beschrifteten Daten) verwenden kann, um genau die Momente zu erfassen, in denen Benutzer mit dem Kopf nach vorne gerichtet gehen und in dem ein VR-System somit die akkumulierte Drift ableiten kann. Die SVM liefert die zuverlässigsten Ergebnisse (Klassifikationsgenauigkeit >95%) bei niedrigem Rechenaufwand und kurzer Rechenzeit. Eine lineare Interpolation wird verwendet, um die Drift der VR-Ansicht reibungslos und unbemerkt anzupassen und dabei deutlich unter der Schwelle der Wahrnehmbarkeit zu bleiben.

In typischen Mehrbenutzer- und großflächigen VR-Szenarien, z. B. Museen und Themenparks, kann diese Technik die Kopf-zu-Körper-Haltung eines Benutzers mehrmals pro Minute zuverlässig bestimmen, wenn der Benutzer in die Bewegungsrichtung blickt, auch bei natürlicher und entspannter Bewegung (einschließlich des Kopfes). Darüber

6 Orientierungsschätzung

hinaus beeinflusst das Verfahren die Visualisierung auf rechenschwachen VR-Systemen nicht. Die Methode aus [129] könnte in zukünftigen Arbeiten zur Orientierungsschätzung untersucht werden.

Dieser Kalibrierungsmechanismus kann alle Orientierungsmethoden des Stands der Technik und deren Fehler kalibrieren. Die kalibrierte Orientierung wird in Kapitel 7 verwendet, um die Trajektorie eines Fußgängers oder Athleten zu rekonstruieren.

7 Posenschätzung

Einordnung

Abbildung 7.1 zeigt die Zuordnung der Posenschätzung in der übergeordneten Verarbeitungskette der Dissertation. Aus Positionsschätzungen eines Funklokalisierungssystems (Kapitel 4), Geschwindigkeits schätzungen (Kapitel 5) eines lose platzierten verrauschten Trägheits sensors und der kalibrierten Orientierung (Kapitel 6) gibt die Posenschätzung rekonstruierte Trajektorien aus. Dieses Kapitel baut auf den zuvor veröffentlichten Ergebnissen [131, 133, 130, 323, 129] auf. Die Ergebnisse dieses Kapitels sind allerdings bis heute, 12. Oktober 2021, unveröffentlicht. Im Rahmen dieser Forschungsarbeiten wurden zwei Patente eingereicht [118, 120].

Kurzzusammenfassung

In Lokalisierungs-, Verfolgungs- und Navigationsanwendungen ist es wichtig, menschliche Bewegungen genau zu bestimmen und ihre Trajektorie vorherzusagen, Brems- oder Ausweichmanöver einzuleiten, um Unfälle zu vermeiden, Kollisionen zu vermeiden und eine effektive Mensch-Computer-Interaktion bspw. in Anwendungen der augmentierten und virtuellen Realität zu erreichen. Wenn der retrospektive Pfad der (Verkehrs)Teilnehmer zum aktuellen Zeitpunkt bekannt ist, führen Verfahren zur Posenschätzung eine (kurzfristige) Vorhersage des zukünftigen Pfades und des Geschwindigkeits- und

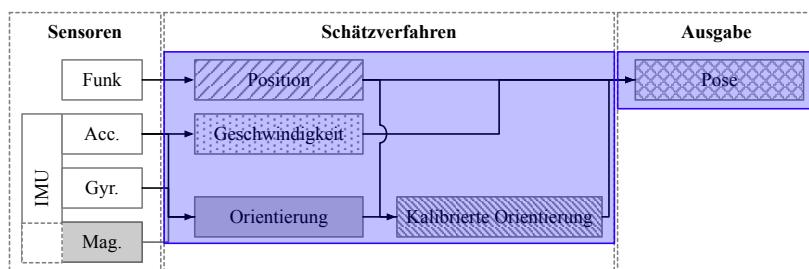


Abbildung 7.1: Einordnung der Posenschätzung in die übergeordnete Verarbeitungskette (blaue Kästchen umschließen die beteiligten Komponenten).

Richtungsänderungsverhaltens von Personen und Fahrzeugen durch.

Insbesondere in schwierigen Situationen hilft ein komplementäres Sensorsystem, bestehend aus Funk- und IMU-Sensoren, Informationen über die aktuelle Pose im Laufe der Zeit zu erhalten. Diese Sensorsysteme sind heute in jedem modernen Mobiltelefon und häufig in moderner tragbarer Haut- oder Modeelektronik (engl. *wearable technology*) verfügbar und schätzen oder prognostizieren Posen von Objekten. Die Fusion von funkbasiert Lokalisierungs- und Trägheitssensorik stabilisiert die Posenschätzung und erhöht deren Genauigkeit. Das funkbasierte Lokalisierungssystem liefert absolute Positionen. Der Beschleunigungsmesser eines Trägheitssensors liefert Informationen zu Beschleunigung, Geschwindigkeit und Entfernung relativ zum Objekt. Die Gyroskop- und Magnetometersensoren eines Trägheitssensorsystems liefern eine relative und absolute Ausrichtung des Objekts.

Die optimale Fusion dieser multimodalen Informationen ist jedoch noch nicht gelöst. Bayessche Fusionsverfahren repräsentieren den Stand der Technik, wie z. B. Kalman- und PF. Das Bewegungsmodell des KF ist typischerweise sehr generell, um möglichst viele Bewegungsformen abzudecken, wohingegen das Mess- und Prozessrauschen sehr anwendungsspezifisch definiert wird. Im Gegensatz dazu sind PF nicht auf Gaußsche Verteilungen beschränkt, sondern leiden zur Laufzeit unter einem sehr hohen Rechenaufwand, haben ernsthafte Schwierigkeiten bei der Korrektur von Messfehlern und haben auch kein optimiertes Bewegungsmodell.

Die vorangegangenen Experimente im Rahmen dieser Dissertation haben zwar gezeigt, dass datengetriebene Ende-zu-Ende-Lernmethoden komplexe, dynamische, nichtlineare Bewegungsmuster von Menschen lernen können, um so die absolute Position (in Kapitel 4), die Geschwindigkeit und Distanz (in Kapitel 5) und die Orientierung (in Kapitel 6) genauer und zuverlässiger zu bestimmen als Methoden des Stands der Technik. Es ist jedoch unklar, ob die Schätzungen dieser datengetriebenen Methoden zu genaueren und robusteren Posen und Trajektorien beitragen und wie diese Schätzungen optimal zu einer genauen und robusten Pose fusioniert werden können. Bekannt ist, dass in klassischen Systemen wie PDR und SINS zur Fusion multimodaler Informationen (Funk- und Trägheitsmessungen),

Positions- und Orientierungsfehler von wenigen Dezimetern oder Grad oder ein (vorübergehender) Ausfall des gesamten Lokalisierungssystems zu gravierenden Posenfehlern führen und so im zeitlichen Verlauf fehlerhafte Trajektorien rekonstruieren.

Deshalb werden in diesem Kapitel Methoden zur Fusion von Positionen, Geschwindigkeiten, Beschleunigungen und Orientierungen untersucht: eine modellbasierte PDR, eine KF-basierte und eine neue datengetriebene Methode. Die Experimente belegen, dass der datengetriebene Schätzer, das individuelle Bewegungsverhalten von Athleten trotz der rotationsinvarianten Platzierung der Trägheits- und Funksensoren erlernen kann und die höhere Genauigkeit und Robustheit der Eingabekomponenten (aus den vorherigen Kapiteln) zu genaueren Posen und Trajektorien führen. Die Experimente zeigen auch, dass die datengetriebene Methode die Fusion der Daten implizit erlernen kann und Posen und Trajektorien genauer und robuster vorhersagen kann als die beiden anderen Methoden des Stands der Technik. Darüber hinaus wird gezeigt, dass datengetriebene Verfahren die Eingabesequenzen so verarbeiten, dass sie von bekannten elliptischen Trajektorienformen auf unbekannte zufällige Trajektorienformen verallgemeinern. Außerdem bestätigen die Experimente, dass die neue Methode erfolgreich zukünftige Bewegungsdaten vorhersagt (extrapoliert), implizit Sensorrauschen entfernt und unterabgetastete Funksignale mit Trägheitsmessungen interpoliert.

Das Kapitel ist folgendermaßen strukturiert: Abschnitt 7.1 führt die Posenschätzung ein. Abschnitt 7.2 beschreibt die Herausforderungen der Posenschätzung. Abschnitt 7.3 diskutiert vorausgegangene Arbeiten aus dem Stand der Technik. Aus der Diskussion leitet Abschnitt 7.4 die Beiträge der hier erforschten Posenschätzung ab. Abschnitt 7.5 beschreibt die Verarbeitungskette der modell- und der datengetriebenen Methoden und deren Architekturen. Abschnitt 7.6 beschreibt das Experiment, das zur Auswertung der Methode verwendet wird. Abschnitt 7.7 präsentiert die Ergebnisse der Auswertung der modellbasierten Methode. Abschnitt 7.8 präsentiert die Ergebnisse der Auswertung der datengetriebenen Methode. Abschnitt 7.9 diskutiert die Ergebnisse und Limitierungen der Posenschätzung. Abschnitt 7.10 fasst die wesentlichen Inhalte des Kapitels zusammen und gibt einen Ausblick.

7.1 Einführung

Die Herausforderungen von Lokalisierungsanwendungen bestehen darin, menschliche Bewegungen genau zu bestimmen, den retrospektiven Verlauf der Pose zu berücksichtigen und den Bewegungsverlauf vorherzusagen, um etwa eine Unfallprävention [15], Kollisionsvermeidung [334] sowie eine effektive Mensch-Computer-Interaktion [245], z. B. in AR und VR Anwendungen [362, 362, 122, 363, 284], zu ermöglichen. So ist im Bereich der Fußgängernavigation die exakte räumliche 2D-Körperpose, also der Aufenthaltsort (Position) und die räumliche Ausrichtung (Orientierung), entscheidend, um etwa blinden Menschen die effiziente und sichere Überquerung einer verkehrsbelebten Straße zu erleichtern [389] oder Kollisionen zwischen Nutzern und Umgebung in der virtuellen Realität zu vermeiden bzw. immersive Erfahrungen zu ermöglichen [362, 363, 129]. Ebenso trägt die exakte Bestimmung der räumlichen Pose von Verkehrsteilnehmern und deren Posenverlauf zur Verbesserung der Fahrsicherheit und der Unversehrtheit von Fußgängern und Fahrradfahrern maßgeblich bei [30]. Auch im Sport ist eine genaue Posenschätzung von Athleten und Zubehör notwendig, um den Spielablauf und die Leistung der Sportler automatisch zu erfassen.

Bereits ein Posenfehler von wenigen Dezimetern bzw. Grad oder ein (zeitweiliger) Ausfall des gesamten Lokalisierungssystems würde dazu führen, dass sich die jeweiligen Teilnehmer an einer scheinbar versetzten Position befinden, sich in eine vermeintlich falsche Richtung fortbewegen oder deren Aufenthaltsort und räumliche Ausrichtung (zeitweise) nicht mehr zu bestimmen sind. Insofern erschweren Posenfehler nicht nur die Bestimmung der momentanen Pose und des zukünftigen Posenverlaufs eines Objektes enorm, sondern haben insbesondere ungeahnte Auswirkungen auf die Sicherheit mit ggf. lebensbedrohlichen Folgen des einzelnen Verkehrsteilnehmers [527]. Daher werden seit Jahrzehnten Lokalisierungstechnologien in Innen- und Außenbereichen beforscht, um den wahrscheinlichen Posenverlauf eines Objektes zu einem bestimmten Zeitpunkt (in der Zukunft) exakt zu bestimmen. Ein Schwerpunkt liegt darin, wie (nicht) lineare Bewegungen von Objekten mithilfe klassischer Verfahren der Zeitreihenanalyse wie renommierten Bayesschen Filtern effektiv modelliert werden können.

Verwandte Arbeiten

In vorangegangenen Forschungen wurden unterschiedliche komplementäre Sensoriken beforscht, um eine möglichst genaue Pose zu ermitteln. Prominente Arbeiten kombinieren Kamera- und Inertialsensorik, hängen aber stark von Sichtverbindung und Belichtung ab und ermitteln typischerweise relative Positionen, da sie keinen Bezug zum Weltkoordinatensystem haben (SLAM [347, 59] und VIO [171, 142, 510, 483]). Andere Arbeiten kombinieren Schall- und Inertialsensorik [280] oder verschiedene Funk- und Inertialsensoren (Bluetooth [267], WiFi [346]) erzielen allerdings sehr niedrige Genauigkeit und funktionieren nur auf kurze Distanzen zwischen Sendern und Empfängern. Die Forschungen haben gezeigt, dass mithilfe einer Komplementärsensorik, bestehend aus (Schmal-, Breit- und Ultrabreitband)Funk- und inertialer Messsensorik, die heute in jedem modernen Mobiltelefon verfügbar sind, Informationen über die aktuelle Pose im zeitlichen Verlauf bereitgestellt werden können, um insbesondere in schwierigen Situationen oder ungünstigen Umgebungen zuverlässig akkurate und präzise Posen von Objekten zu schätzen bzw. zu prognostizieren [54]. Die Fusion von Funkortung (absolute Positionen) mit Inertialsensorik (Geschwindigkeiten und Distanzen des Beschleunigungssensors sowie Orientierungen des Gyroskops bzw. Magnetometers) stabilisiert und erhöht die Genauigkeit Posenschätzung [269].

Herausforderungen der Funksensorik

Bei laufzeitbasierten Schätzverfahren, die Funksensorik einsetzen, wird die absolute Position bspw. anhand von ToA bzw. TDoA-Verfahren ermittelt, d. h. es werden die verschiedenen Ankunftszeiten der Funksignale an den jeweiligen Antennen gemessen und daraus die Position bestimmt [307]. Besteht direkter Sichtkontakt zwischen Funkempfänger (Antenne) und dem zu verortenden Objekt (Funksender), lassen sich absolute Positionen mit einem MAE von rund 0.20 m [127] zuverlässig schätzen. Da Funksignale jedoch durch abschirmende Objekte wie Wände, (metallische) Gegenstände oder andere Barrieren umgelenkt, gestreut oder gedämpft und an glatten Oberflächen reflektiert werden, d. h. sich über mehrere Wege ausbreiten können, lässt sich die Signalausbreitung nicht deterministisch bestimmen, sodass die absolute Positionsgenauigkeit aufgrund verfälschter TDoA-Messwerte

insbesondere in OLoS und NLoS-Umgebungen, abnimmt [312].

Wie in Kapitel 4 gezeigt wird, können Positionen in Funkumgebungen mit Mehrwegeausbreitungen mit einem Fehler von <0.12 m ermittelt werden [127]. Allerdings hängen diese Ansätze vom Wissen über die Umgebung ab, sodass sie typischerweise nur für die Funkausbreitungsumgebung funktionieren, für die sie trainiert wurden. Daher wird die Positionsschätzung mit Geschwindigkeits- und Orientierungsschätzungen von Trägheitssensoren zusammengeführt, um genauere und robustere Posen zu schätzen, die in verschiedenen Umgebungen funktionieren. Darüber hinaus ist die Kapazität von Funksystemen begrenzt, sodass sie nur einer begrenzten Anzahl von Teilnehmern Kommunikations- und Lokalisierungsoptionen bieten können.

Es wird daher geforscht, ob komplementäre Sensoren langfristig eine genaue und robuste Position bestimmen können und nur gelegentlich durch eine Funkposition stabilisiert werden müssen. Auf diese Weise kann die Kapazität der Lokalisierungskomponente reduziert und die Kommunikationskapazität, bspw. für standortbasierte Unterhaltungsanwendungen maximiert werden.

Herausforderungen der Inertialsensorik

Ähnlich wie funkbasierter Lokalisierungsverfahren leiden auch bekannte Verfahren zur Positionsbestimmung mit Trägheitssensorik unter verschiedenen Problemen. Trägheitsnavigationssysteme (SINS) ermitteln durch mathematisches Integrieren von Inertialsensormessungen, wie bspw. Winkelgeschwindigkeiten (Gyroskop) oder der erdmagnetischen Flussdichte (Magnetometer) die relative bzw. absolute Orientierung der IMU [475]. Werden die gravitationssubtrahierten linearen Beschleunigungsmessungen¹ des Beschleunigungssensors einmalig integriert, so lässt sich die relative Distanz- bzw. Geschwindigkeitsänderung und bei doppeltem Integrieren die relative Positionsänderung der IMU bestimmen [448]. Jedoch führen insbesondere mit Signalrauschen behaftete IMU-Sensordaten von günstigen

¹Die Beschleunigungsmessung einer IMU besteht aus einer Gravitations- (beschreibt die Lage des Sensors im Raum) und einer linearen Beschleunigungskomponente (beschreibt die Bewegung des Sensors). Um die aktuelle Geschwindigkeit oder die vom Sensor zurückgelegte Entfernung abzuschätzen, muss daher die Gravitationskomponente eliminiert, bzw. subtrahiert werden.

Inertialsensoren, wie sie in handelsüblichen Mobiltelefonen verbaut sind, mit jedem Integrieren zu einem stark akkumulierenden Fehler, eine doppelte Integration führt bereits zu einem Positionsfehler von mehreren Metern innerhalb weniger Sekunden [243, 68].

Im Gegensatz zu SINS, ermitteln PDR Systeme Schritte- und Schritt-längen und deren Orientierung, um eine Position zu bestimmen. Allerdings sind diese hochgradig nutzer- und anwendungsspezifisch und können somit typischerweise keine genaue und robuste Position ermit-teln. Inspiriert von der Funktionsweise klassischer PDR Systeme zeigen Kapitel 5 und 6, wie mit datengetriebenen Verfahren eine genaue Ge-schwindigkeit (Schrittänge) und Orientierung mit Trägheitsmessun-gen erzielt werden können. Allerdings ist ungeklärt, wie Positionen, Ge-schwindigkeiten und Orientierungen zu einer genauen Pose optimal, bspw. ohne Positionssprünge, robust gegen Datenlücken und mit wei-tem Prognosehorizont, zusammengeführt werden können.

Methoden zur Posenschätzung

Aktuelle Methoden zur laufzeitbasierten Posenschätzung basieren bspw. auf Bayes-Filtern (KF und PF [73]), nutzen (Kombinationen von) Funk- und Inertialsensorik [269] und ermöglichen die Bestim-mung der Pose von Objekten [243], wie bspw. Fahrzeugen oder Perso-nen. Einfache physikalische Phänomene wie lineare Bewegungsverläu-fe, bspw. von Fahrzeugen, werden mit bekannten physikalischen Geset-zen oder klassischen handgefertigten laufzeitbasierten Verfahren wie Bayes-Filtern, genau vorhergesagt. Allerdings schätzen diese herkömm-lichen Verfahren die nichtlinearen Bewegungen einer Person nur unge-nau [143]. Menschen ändern bspw. ihre Bewegungsgeschwindigkeit und ihren Kurs aufgrund von Hindernissen oder anderen Umwelteinflüssen kurzfristig [49]. Darüber hinaus sind die biomechanischen Eigen-schaf-ten eines Menschen personenspezifisch, d. h. sie sind abhängig von Alter, Geschlecht und Körperzustand [128]. Bayes-Filter unterstützen zwar aufgrund ihrer rekurrenten Architektur, die Schätzung vergange-ner, aktueller und zukünftiger Bewegungszustände, erfordern jedoch eine aufwendige, manuelle Parametrisierung der kinematischen Eigen-schaften des Objekts und der sich im Zeitverlauf verändernden Um-gebungsbedingungen, wie etwa Sensorrauschen, NLoS-Szenarien oder andere Interferenzen [469]. Insofern ist die Parametrisierung des nicht-deterministischen menschlichen Bewegungs- und Umgebungsverhal-

tens der Messumgebung mit Bayes-Filtern sehr aufwendig umzusetzen [127]. Im Gegensatz dazu haben aktuelle Forschungsarbeiten gezeigt, dass datengetriebene Ende-zu-Ende Lernverfahren leichter zu modellieren sind als KF und in der Lage sind, komplexe, nichtlineare Bewegungsmuster von Menschen zu lernen und selbst variable Messumgebungsveränderungen im Kern ihrer Architektur erfolgreich abzubilden, selbst über lange Zeiträume [78, 34, 128].

Lösungsansatz

Die Schlüsselidee des Verfahrens zur genauen und robusten Posenschätzung, das in dieser Dissertation vorgestellt wird, besteht darin, die niederfrequente datengetriebene Positionsverfolgung (siehe Kapitel 4) mit hochfrequenten datengetriebenen relativen Geschwindigkeiten (siehe Kapitel 5) und mit der langzeitstabilen niederfrequenten, datengetriebenen Gier-Orientierung (siehe Kapitel 6) zu fusionieren.

Ein neuartiges KNN bildet Zeitreihendaten, nämlich Funk- und Inertialsensordaten zu den beobachteten 2D-Körperposen einer Person im Zeitverlauf, als Eingabefolge auf konsekutive Referenzdaten (Referenzpose) ab und erzeugt Zeitreihendaten (zukünftige 2D-Körperposen) als Ausgabefolge zur Beschreibung menschlicher Bewegungen. Experimente vergleichen herkömmliche Ansätze, nämlich eine modellbasierte PDR-Methode, die sequentiell Positionen und Geschwindigkeiten zusammenführt, und ein optimiertes KF mit dem neuartigen Verfahren auf den Ausgabedaten bzw. Ergebnissen der vorangegangenen Kapitel 4 bis 6. Die Experimente zeigen, dass das neue Modell präzise, anpassbare und verallgemeinerbare Bewegungsmodelle lernt und die Positionsgenauigkeit der bekannten Methoden übertrifft. Dies zeigt sich besonders bei plötzlichen, zufälligen Bewegungsänderungen, bei denen es kürzere Einschwingzeiten und insgesamt geringere Fehler zeigt. Das neuartige Verfahren sagt Zufallstrajektorien bis zu zwei Sekunden in die Zukunft mit einem Fehler von unter 0.12 m voraus. Tiefere Modelle, z. B. Modelle mit mehr Schichten waren nicht signifikant genauer als flachere. Eine Überprüfung der Relevanz der Netzsichten zeigt auch, dass tiefere Schichten für die Ausgabe nicht relevant sind.

7.2 Problembeschreibung

Zunächst führt Abschnitt 7.2 das Problem ein. Dann erörtern Abschnitt 7.2.1 und Abschnitt 7.2.2 die Herausforderungen der Fusion unterschiedlicher Sensordatenströme. Abschließend schlägt Abschnitt 7.2.3 eine Lösung vor.

Die Koppelnavigation ist Grundlage für viele heutige Navigationsanwendungen, darunter die Navigation von Fußgängern, Fahrzeugen und Flugzeugen [58]. Ist die Ursprungposition p_0 eines Objektes bekannt, so lässt sich unter Zuhilfenahme der Orientierung θ und der Geschwindigkeit v , Beschleunigung acc oder der zurückgelegten Distanz d die laufende näherungsweise absolute Pose p_1, p_2, \dots, p_n bestimmen [58]. Ebenso lässt sich die zukünftige absolute Pose anhand vergangener Posen p , wie in Kapitel 3.2.1 beschrieben, ermitteln.

Verschiedene Umgebungseinflüsse, wie etwa Sensorinterferenzen, darunter (zeitweilige) Signalverluste, -dämpfung, -streuung, -beugung, -brechung, -reflektion, -drift und Sensorrauschen führen hierbei zu einer Kurs- bzw. Positionsabweichung. Um die Koppelnavigation hinreichend störungs- und ausfallsicher zu betreiben, werden Positionen bspw. der Fusionslokalisierung eingesetzt [211]. So können gleichermaßen die IMU-Messungen die Messungen des Lokalisierungssystems korrigieren und umgekehrt [269]. Inertiale Trägheitsmesseinheit werden in regelmäßigen Abständen anhand der komplementär eingesetzten Lokalisierungstechnologie re-kalibriert [269]. Da die Messdaten, die von funkbasierten Lokalisierungssystemen bereitgestellt werden, ebenfalls Signalinterferenzen unterliegen, können Trägheitsmessungen insbesondere bei Funk-Signalverlust, die Koppelnavigation weiterhin aufrecht erhalten [269].

7.2.1 Datenlücken und asynchrone Datenströme

Bekannte Fusionsmethoden leiden jedoch unter mehreren Problemen. Sensoren liefern Messwerte mit unterschiedlichen und variablen Datenraten, die von vordefinierten Filtern nicht verarbeitet werden können, da modellgetriebene Ansätze fordern, dass verschiedene Sensoren zeitsynchronisiert sind und Sensordatenströme nicht ausfallen. Da moderne Funk- und Trägheitssysteme lose gekoppelt sind, wird die Funkposition von einem System in der Umgebung bestimmt und muss

an das zu lokalisierende Objekt zurückgesendet werden, oder die Trägheitsmessungen müssen zusätzlich zu den emittierten Funksignalen an die Lokalisierungsumgebung übertragen werden. Beide Fälle führen zu nichtdeterministischen Verzögerungen, die die Lokalisierung in Echtzeit verhindern, und diese Zeitverzögerung kann nur annähernd durch ein vordefiniertes (starres) Modell beschrieben werden. Im schlimmsten Fall sind Datenströme asynchron und es fehlen einzelne Daten. In den Experimenten (Abschnitt 7.6) werden diese Auswirkungen auf die Genauigkeit der Schätzer untersucht.

7.2.2 Nichtdeterministische Kalibrierung

Außerdem ist ungeklärt, wann und mit welcher Gewichtung eine aktuelle Position oder Orientierung optimal zur Korrektur und Kalibrierung beiträgt, um akkumulierte Fehler in einem Trägheitsmesssystem oder fehlerhafte Funkpositionen zu korrigieren und Posen und Trajektorien optimal zu rekonstruieren.

Eine Korrektur der aktuellen Posenschätzung ist nur dann sinnvoll, wenn die Korrekturinformationen (Positionen oder Geschwindigkeiten) fehlerfrei oder genauer als die aktuelle Pose sind. Klassische, handgefertigte Verfahren bestimmen diese Zusammenhänge häufig fehlerhaft. Da z. B. die Fehler im Trägheitsmesssystem von der anwendungsspezifischen Bewegungsdynamik abhängen, sodass viele Änderungen in Geschwindigkeit und Richtung zu höheren Fehlern führen oder die Fehler im Funksystem von den dynamischen, umgebungsspezifischen Signalausbreitungseigenschaften abhängen. Andererseits ist eine ständige (in jedem Zeitschritt) Kalibrierung der Geschwindigkeitsschätzung mit einer Position nicht sinnvoll, da die Funkposition typischerweise über kurze Zeiträume größere Fehler verursachen würde als die Geschwindigkeitsschätzung. Daher ist es unmöglich zu bestimmen, wann eine Kalibrierung mit einer Position sinnvoll ist, um die Posengenauigkeit und -robustheit zu maximieren.

Es ist auch unklar, mit welcher Gewichtung und Aktualisierungsrate die Position und Orientierung proportional zur aktuellen Geschwindigkeitsschätzung, bzw. Schrittlängenschätzung, zur Kalibrierung oder Korrektur verwendet werden soll. Ein einfaches, unüberlegtes Anpassen der aktuellen Positionsschätzung des Trägheitssensorsystems an die aktuelle Funkposition kann zu schwerwiegenden Sprüngen in der korri-

gierten Position führen, oder im schlimmsten Fall könnte die Schätzung des Trägheitssensorsystems genauer sein und die Funkposition würde einen Positionsfehler verursachen.

7.2.3 Lösungsvorschlag

Daher wird in Abschnitt 7.5 ein datengetriebenes Verfahren vorgeschlagen, das unterschiedliche Datenraten der Sensoren, nicht synchronisierte Messwerte und variable Zeitverzögerung verarbeiten kann und implizit entscheidet, wie und wann die Funkposition und Orientierung verwendet werden, um eine optimale Pose zu schätzen. Die Kernidee besteht darin, Kombinationen der Eingangsströme, nämlich Funkposition und Geschwindigkeit und Orientierung der Trägheitsmessungen, zu variieren und diese unter Verwendung eines datengetriebenen Verfahrens auf Referenzdatenströme, nämlich Referenzposition und -orientierung, abzubilden. LSTM-Zellen können implizit Datenlücken und andere Artefakte in den Datenströmen identifizieren und lernen, diese zu kompensieren. Der datengetriebene Ansatz lernt implizit aus den Daten, wie und wann die jeweiligen Korrekturinformationen in die Fusion einfließen müssen, um eine optimale Pose zu schätzen.

7.3 Verwandte Arbeiten

Abschnitt 7.3.1 stellt Arbeiten vor, die Kalman- oder Partikel-Filter zur Koppelnavigation bzw. Fusion von Trägheitsmessungen mit Funkpositionen einsetzen. Abschnitt 7.3.2 stellt Arbeiten vor, die dazu datengetriebene Verfahren einsetzen.

7.3.1 Rekursive probabilistische Verfahren

Um Lokalisierung in Städten, Schluchten, Tälern und Innenräumen, in denen die Funklokalisierung eingeschränkt oder nicht verfügbar ist, durchzuführen, ist eine mögliche Lösung die Koppelnavigation mittels Fusion von Inertialsensordaten mit Funksensordaten [24]. Diese sequentielle Fusion, schätzt die aktuelle Position aus der vorherigen Position, sobald eine Positionsänderung erkannt wird [223]. Somit enthält die aktuelle Position die Fehler des Trägheits- oder Funksystems, bspw. fehlerhafte Schrittängen und Geschwindigkeiten durch unterschiedliche Körpergrößen, durch Mehrwege gestörte Funkpositionen und fehlerhafte Orientierungen durch ferromagnetische

Störungen des Magnetfeldsensors [223, 12, 358, 224].

KF fusionieren Funk- und Trägheitssensoren. Meistens [481, 241, 242, 72, 456] wird das KF in unterschiedlichen Varianten eingesetzt, um das Messrauschen der Funk- und Trägheitssensoren zu erfassen. Die Messkovarianz wird dabei oftmals auf vorliegende Trainingsdaten optimiert. Das Bewegungsmodell wird anwendungsspezifisch möglichst generell definiert. Beides führt dazu, dass diese Fusionsart stark anwendungsspezifisch ist und aufwändig parametrisiert werden muss. Leichte Abweichungen der Sensoren oder der Bewegung führen zu einer Abweichung der Mess- und Prozesskovarianz und des Bewegungsmodells und so zu gravierenden Schätzfehlern.

Gusenbauer et al. [163] nutzen ein lineares KF, um GNSS-, Magnetometer- und Beschleunigungssensordaten mit einer Genauigkeit von $MAE=4.2\text{ m}$ zu fusionieren. Zhuang et al. [519] nutzen ein EKF, um WLAN- und Inertialsensordaten mit einer Genauigkeit von $MAE=4.19\text{ m}$ zu fusionieren und sind etwas robuster gegenüber Ausreißern als Li et al. [272], die eine Genauigkeit von $MAE=6.2\text{ m}$ erzielen. Tao et al. [274] begründen die hohe Ungenauigkeit existierender Verfahren damit, dass ferromagnetische Metalle, die sich in der Umgebung der Inertialsensorik befinden, die Magnetometerdaten verzerrn, die Orientierungsschätzung verfälschen und somit die Posengenauigkeit senken. Sczyslo et al. [393] nutzen ebenfalls ein KF, um Positionen, die aus ToA-Messungen eines Ultrabreitband-Lokalisierungssystems trilateriert wurden, mit Beschleunigungsmessungen eines Trägheitssensors mit einer Genauigkeit von $MAE=0.57\text{ m}$ zu fusionieren. Der Zustandsvektor des KF setzt sich aus Positionen des UWB-Systems, Beschleunigungen und davon integrierten Geschwindigkeiten v_n des Trägheitssensors zusammen. Im Gegensatz dazu nutzen Perttula et al. [338] ein PF, um Gyroskop- und Beschleunigungssensordaten mit ToA Werten eines UWB Funksystems zu fusionieren. Sie erzielten unterschiedliche Genauigkeiten, wenn sie die Sensoren am Rumpf eines Menschen befestigten ($MAE=9.9\text{ m}$) oder an der Taille ($MAE=10.3\text{ m}$). Das deutet darauf hin, dass auch ein PF, der nichtlineare Bewegungen modellieren kann, starken Fehlervarianzen unterliegt. Daher wurde in den Experimenten (Abschnitt 7.6) ein renommiertes KF eingesetzt, um den Stand der Technik bestmöglich darzustellen.

Abgrenzung: Diese probabilistischen Methoden, die alle (nur) den aktuellen und vorherigen Zustand (Messung und Schätzung) berücksichtigen, können daher keine Datenlücken und langfristigen Datenanomalien und -beziehungen identifizieren. Darüber hinaus sind Art und Gewichtung der Fusion a priori fest definiert. Im Gegensatz dazu ermöglicht die hier vorgeschlagene Methode, diese Herausforderungen direkt und implizit aus Trainingsdaten zu lernen und liefert deutlich genauere Posen. Kapitel 2 diskutiert probabilistische (KF und PF) Fusionsverfahren ausführlich.

7.3.2 Datengetriebene Verfahren

KNN lernen Korrelationen von Merkmalen, indem Sensordaten dem Netz als Eingabe zugeführt und gegen ein Trainingsziel \hat{y} trainiert werden. Durch kontinuierliche Veränderung der Gewichte w , unter Annäherung des minimalen Fehlers E , sind KNN in der Lage, Fehler, wie Sensorrauschen in den Eingabedaten zu minimieren, fehlende Sensordaten zu interpolieren und zukünftige Werte zu extrapoliieren.

Während es einige Arbeiten gibt, die Geschwindigkeiten mit RNNs [223] oder verstärkendem Lernen [78] schätzen, um daraus eine Distanz und daraus eine unkalibrierte Trajektorie zu rekonstruieren, gibt es kaum Verfahren, die multimodale Informationen, nämlich Funk- und Trägheitsmessungen, mit datengetriebenen Verfahren untersucht haben. Zyner et al. [527] nutzen LSTM Zellen, um zukünftige Trajektorie eines Fahrzeugs vorherzusagen. Das Modell nutzt GPS-Positionen, die Orientierung einer IMU und Geschwindigkeiten eines Odometriesensors des Fahrzeugs und rekonstruiert die Trajektorie in 90.66 % aller Fälle korrekt. Allerdings werden die Referenzmessungen aus den Messungen hergeleitet und sind kaum vergleichbar. Im Gegensatz dazu reduzieren Yao et al. [495] das Sensorrauschen von unterschiedlichen Signalen, wie GPS und IMU. Sie nutzen ein CNN und ein RNN, um sowohl relative Bewegungen der IMU, als auch globale Bewegungen des GPS Systems zu extrahieren und um Signaldysfunktionen zu modellieren. Ihr Ansatz erzielt einen mittleren absoluten Fehler von 4.043 m ($SD=0.524$ m), während die klassische Sensordatenfusion einen mittleren absoluten Fehler von 6.065 m ($SD=0.565$ m) erreicht.

Da beide Vorarbeiten die Bewegungen von Fahrzeugen untersuchen, ist noch unklar, wie Sensorinformationen, die die Bewegung des Men-

schen beschreiben, optimal zusammengeführt werden können. Die Kernidee dieses Kapitels ähnelt den Ideen von Zyner et al. [527] und Yao et al. [495], passt ihre Ideen aber an die Domänen der schmalen bis ultrabreitbandigen Funklokalisierung und 6DoF-Trägheitssensoren an. Im Gegensatz zu diesen Vorarbeiten wird im Rahmen dieser Dissertation eine groß angelegte Architektur- und Parameterstudie durchgeführt, um die optimale datengetriebene Architektur für diese Domänen abzuleiten. Darüber hinaus werden die Auswirkungen unterschiedlicher Eingabe- und Ausgabedatenströme, der Ausfall von Datenströmen, die Vorhersagegenauigkeit, das Einschwingverhalten in hochdynamischen Situationen und der Einfluss der Sequenzlänge der Eingabeströme auf die Genauigkeit und Robustheit untersucht.

Zusammenfassung: Die Literaturübersicht zeigt, dass diese Dissertation die einzige öffentlich bekannte Forschungsarbeit ist, die die genaue und zuverlässige Erfassung, Verfolgung und Fusion menschlicher Bewegungen mit datengetriebenen Verfahren in hochdynamischen Situationen mit abrupten Änderungen in Geschwindigkeit und Richtung, mit verrauschten Trägheitsmessungen von lose platzierten Sensoren und Funksignalen in Mehrwegeumgebungen untersucht. In Kapitel 2.3.6 werden die wenigen öffentlich bekannten datengetriebenen Fusionsverfahren ausführlich diskutiert.

7.4 Beiträge

Da unklar ist, wie sich unterschiedliche Eingabeveriablen, nämlich Position, Beschleunigung, Geschwindigkeit und Orientierung, deren Fehler und Fusion auf die Genauigkeit der zu lernenden Körperpose auswirken, ist das übergeordnete Forschungsziel dieses Kapitels, verschiedene Sensordatensätze aufzuzeichnen und deren Auswirkungen auf die Genauigkeit der menschlichen Körperposenschätzung im Zusammenhang mit verschiedenen Architekturen datengetriebener Verfahren zu untersuchen. Dabei stellen sich folgende Forschungsfragen:

RQ5: Können datengetriebene Verfahren die Fusion verschiedener Sensordatenströme (Trägheitssensor und Funksystem) im Vergleich zu modellbasierten Verfahren verbessern?

RQ5a: Wie genau können datengetriebene Verfahren zur Fusion verschiedener Sensordatenströme die Pose eines Fußgängers in dynamischen Situationen schätzen?

RQ5b: Wie weit voraus können datengetriebene Verfahren zur Fusion verschiedener Sensordatenströme die Pose eines Fußgängers in dynamischen Situationen vorhersagen?

RQ5c: Welche Vorteile bieten datengetriebene Verfahren zur Fusion verschiedener Sensordatenströme eines Fußgängers in dynamischen Situationen?

RQ5d: Gibt es einen Einfluss verschiedener Kombinationen von Eingabevervariablen, z. B. Position, Geschwindigkeit und Beschleunigung auf die Schätzgenauigkeit von rekurrenten datengetriebenen Modellen?

7.5 Methoden

In diesem Abschnitt wird die neuartige Verarbeitungskette im Detail diskutiert. Zunächst wird die Datenvorverarbeitung beschrieben (Abschnitt 7.5.1). Dann wird die Hauptverarbeitung mit der modellbasierten PDR Architektur (Abschnitt 7.5.2) und der innovativen datengetriebenen Architektur präsentiert (Abschnitt 7.5.3).

7.5.1 Datenvorverarbeitung

Die Datenverarbeitung erfolgt für die einzelnen Eingangsvariablen Position (Kapitel 4), Geschwindigkeit (Kapitel 5) und Orientierung (Kapitel 6), so wie es in den jeweiligen Kapiteln beschrieben ist. Abbildung 7.2 zeigt, wie die Daten für die Posenschätzung zusammengesetzt sind. Informationen fließen in der Verarbeitungskette von links nach rechts. Die grauen Komponenten visualisieren die Daten, die bereits durch die Positions-, Geschwindigkeits- und Orientierungsschätzung ermittelt wurden.

Die Datenvorverarbeitung synchronisiert die Eingabedatenströme zeitlich (Funkposition p_{radio} des Funksystems $radio$, Geschwindigkeit v , Orientierung θ_{ori} , Referenzpose ref und Beschleunigung acc), bereinigt sie und bereitet sie vor (durch Stichprobewiederholung und Interpolation) und bündelt die sauberen Datenströme. Die sauberen Datenbündel werden dann weiter zu Segment- (s) und Fenster- (w) -bündeln verarbeitet, wie in Kapitel 5.5.1 beschrieben. Jede dieser Bündelvarianten enthält Funkpositionen p_{radio} , Funkorientierung

7 Posenschätzung

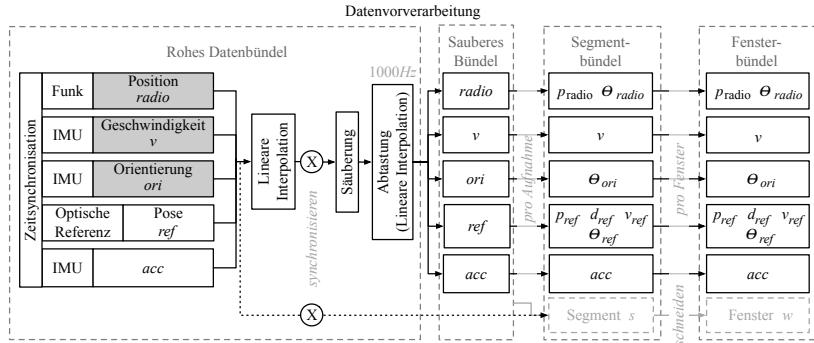


Abbildung 7.2: Ansicht der Vorverarbeitungskette zur Posenschätzung.

θ_{radio} (Bewegungsrichtung zweier aufeinanderfolgender p_{radio}), die Geschwindigkeitsschätzung v , die Orientierungsschätzung θ_{ori} , die Referenzpose ref (mit p_{ref} , d_{ref} , v_{ref} und θ_{ref}) und die rohe 3D-Beschleunigung acc . Es ist zu beachten, dass die aktuelle Orientierung θ_{ori} für alle Fensterbündel unter Verwendung einer kubischen Stützvektormaschine klassifiziert wird. Die Mess- und Referenzsensoren zur Orientierungsschätzung in Kapitel 6 werden gegen die Sensoren zur Geschwindigkeitsschätzung aus Kapitel 5 ausgetauscht.

In Abschnitt 7.7 werden p_{radio} , v , θ_{ori} und (optional) acc zusammengeführt und die Ergebnisse gegen ref ausgewertet. Segmente s repräsentieren die sauberen vorverarbeiteten Daten pro Aufnahme, z. B. ein Athlet mit Funk- und Trägheitssensor. Die Fenster w repräsentieren die Eingabesequenzen für die Posenschätzung, die entlang s gleiten. Es ist wichtig zu beachten, dass diese Vorverarbeitung nur für bestimmte Auswertungen der datengetriebenen Verfahren verwendet wird. Für eine allgemeine realistische Verarbeitung in Echtzeit werden die Daten direkt vor der linearen Interpolation synchronisiert, d. h. die Daten werden direkt beim Eintreffen zusammengeführt und zu Segmenten oder Fenstern weiterverarbeitet. Abbildung 7.2 zeigt diese direkte Verarbeitung durch den gestrichelten Pfeil, der im Segmentbündel endet.

7.5.2 Klassische modellbasierte Methode

Um die Posen und korrespondierende Trajektorie zu rekonstruieren, wird p_0 aus dem *radio*-System als Ausgangsposition verwendet und θ und ρ transformiert (ρ = Abstand, der aus v abgeleitet wird, z. B. $\vec{v}=1 \text{ m/s} \rightarrow \vec{d}=1 \text{ m}$, mit $dt=1.0 \text{ s}$ und $\theta \in \{\theta_{ori}, \theta_{radio}, \theta_{ref}\}$ pro Fenster):

$$p(x) = p_0(x) + \rho \cdot \cos(\theta) \quad (7.1)$$

und

$$p(y) = p_0(y) + \rho \cdot \sin(\theta). \quad (7.2)$$

Wenn die Orientierung θ_{ori} für die Kalibrierung verwendet wird, wird θ entsprechend korrigiert:

$$\theta = \theta_{ori} \quad (7.3)$$

und $p(x)$ und $p(y)$ werden durch das neue θ angepasst. Wenn die Position mit einer aktuellen Funkposition p_{radio} kalibriert wird, werden $p(x)$ und $p(y)$ entsprechend eingestellt:

$$p(x) = p_{radio}(x), \quad (7.4)$$

$$p(y) = p_{radio}(y). \quad (7.5)$$

7.5.3 Datengetriebene Methode

Im Rahmen einer großangelegten Studie wurden unterschiedliche Architekturen (unterschiedliche Varianten von Schichten und deren Anordnung) und Zellkonzepte (Neuronen der Schichten) untersucht, um eine optimale Methode zur Posenschätzung bzw. Trajektorienrekonstruktion herzuleiten.

Zellkonzept

Im Rahmen der Studie wurden die folgenden rekurrenten Zellkonzepte untersucht: Das Standard RNN [109], IRNN [254], LSTM [191] und GRU [76]. Weitere öffentlich bekannte Zellkonzepte von Jozefowicz et al. [217] und Greff et al. [155] wurden nicht untersucht, da Voruntersuchungen zu sehr ungenauen Ergebnissen führten. Details zu den einzelnen Konzepten werden in Abschnitt 3.7 beschrieben. Da das LSTM-Zellkonzept zuverlässig konvergiert und die Posen am genauesten schätzt, werden der Übersichtlichkeit halber nur die Ergebnisse dieses Zellkonzepts im Abschnitt 7.8 berichtet.

Architektur

Neben dem Zellkonzept ist die Struktur, die die Zellen verbindet und einbezieht, für die optimale Modellarchitektur wichtig. Während die Standard-RNN-Zellen mit großem Erfolg bei einer Vielzahl von Problemen verwendet wurden, von künstlichen Additionsaufgaben bis zur Musikgenerierung [191, 217, 254], wurden viele verschiedene Netze vorgeschlagen, die rekurrente Zellen verwenden, um komplexere Konzepte zu entwerfen. Diese Netze reichen von verschiedenen Variationen von tiefen RNNs [180, 330] bis zu spezialisierteren Architekturen wie bidirektionale RNNs [392], Enkodierer- Dekodierer-Netzen [76, 423] oder aufmerksamkeitsbasierte Konzepte [23]. Die Quintessenz ist, dass jede dieser Architekturen für ein ganz spezifisches Anwendungsgebiet, eine spezifische Domäne, entwickelt wurde, von der Bildinterpretation, über Sprach- und Wörterkennung bis zur Übersetzung [23, 76]. Da keine der Domänen der Lokalisierungsdomäne gleicht, müssen die bekannten Verfahren auf die Domäne der Posenschätzung adaptiert und deren Leistung in der neuen Domäne evaluiert werden (Abschnitt 7.8).

Inspiriert von der Idee von Pascanu et al. [330, 331] werden zusätzlich zu gestapelten RNNs tiefe vorwärtsgekoppelte Netze für die Übergangsfunktionen verwendet, um die Eingabe und Ausgabe zu verarbeiten, d. h. Feed-Forward-Schichten (FF) direkt nach der Eingabeschicht und direkt vor der Ausgabeschicht. Dies ermöglicht es dem Netz, komplexere Berechnungen zwischen den Zeitschritten durchzuführen, um die Genauigkeit zu erhöhen. Im Gegensatz zu Pascanu et al. [330] werden keine Tiefenübergangsfunktionen verwendet, da diese das Verschwinden und Explodieren des Gradienten fördern und das Training erschweren.

FF-Schichten werden in vielen Anwendungen mit großem Erfolg verwendet [255], um (räumliche) Merkmale optimal aus hochdimensionalen Daten zu extrahieren. Rekurrente Schichten werden mit großem Erfolg verwendet, um Merkmale optimal im Zeitverlauf zu verfolgen. Um die Vorteile beider Verfahren bestmöglich zu nutzen, werden die FF und RNN Schichten gestapelt. Die Ausgänge einer vorderen Schicht werden in jedem Verarbeitungsschritt der nächsten Schicht zugeführt. Insbesondere für Aufgaben, die eine umfangreiche Verarbeitung der Eingabedaten erfordern, haben tief gestapelte RNNs die einschichtigen RNNs übertroffen [127, 423].

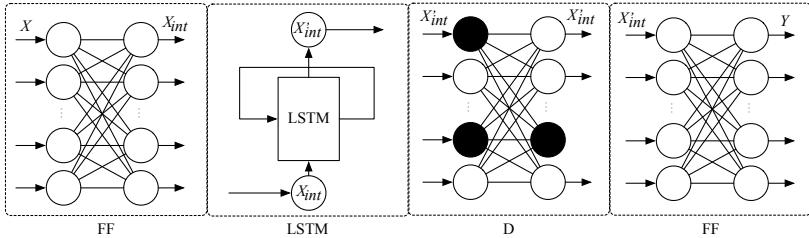


Abbildung 7.3: FRFNN-Architektur zur Fusion multimodaler Sensorsignale. Vorwärtskopplungsschichten (FF) werden verwendet, um die Eingabe und Ausgabe der LSTM-Schicht zu verarbeiten. Eine Verwerfungsschicht (D) nach der LSTM-Schicht verringert die Möglichkeit einer Überanpassung und ermöglicht eine Unsicherheitsschätzung.

Die Schlüsselidee der FRFNN (Feedforward-Recurrent-Feedforward Neural Network, FRFNN)-Architektur besteht darin, die vor- und nachgeschalteten Vorwärtskopplungsschichten zu verwenden, um das Eingangssignal X aus seiner hohen Dimension (Funkposition p_{radio} , Geschwindigkeit v , Beschleunigung acc und Orientierung θ_{ori}) in eine optimale interne Dimension X_{int} für die RNN-Architektur zu transformieren. Abbildung 7.3 zeigt die endgültige Architektur. Die RNN-Architektur kann diese Darstellung optimal verarbeiten und die interne Darstellung X'_{int} an die nachgeschaltete Vorwärtskopplungsschicht weiterleiten. Diese transformiert die interne Darstellung des RNN X'_{int} in die Zieldarstellung einer anderen Dimension (Pose) Y . Die multimodale Eingabedomäne ($p_{radio}, v, acc, \theta_{ori}$) wird somit über ein RNN in eine Zieldomäne (Pose) übersetzt. Die FRFNN-Architektur, die die höchste Genauigkeit erzielt, kombiniert vorgeschalteten FF-Schichten vor LSTM-Zellen mit nachgeschalteten FF-Schichten. Eine Verwerfungsschicht zwischen LSTM-Zellen und nachgeschalteter FF-Schicht reduziert Effekte der Überanpassung und ermöglicht die Bestimmung der Unsicherheit des Verfahrens.

Abbildung 7.4 zeigt die Struktur einer beispielhaften Eingabesequenz X . Ein Fenster w gleitet über die Daten des Segmentes s . Jedes Fenster enthält mehrdimensionale Vektoren, die die EingabevARIABLEN über bspw. 128 Zeitschritte enthalten. Die Position wird durch zwei Dimensionen, nämlich x und y, als 2D-Vektor repräsentiert.

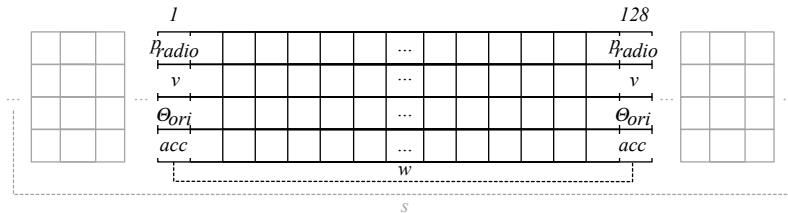


Abbildung 7.4: Eingabedaten (X) des datengetriebenen Posenschätzers. Ein Fenster w mit der Länge 128 Zeitschritte gleitet über das Segment s mit den Signalvektoren: zweidimensionale Funkposition p_{radio} , skalare Geschwindigkeit v , skalare Orientierung θ_{ori} und SMV(acc).

Begründung der Wahl der Architektur

Da Kombinationen von CNN- und RNN-Schichten bereits signifikant genauere Ergebnisse für die Geschwindigkeitsschätzung liefern (Kapitel 5) als Architekturen, die nur auf CNN- oder RNN-Schichten basieren, werden Kombinationen dieser Schichten auch für die Posenschätzung bewertet. Zusätzlich zu gestapelten CNN-RNN-Architekturen werden Vorwärtskopplungsschichten (FF), bidirektionale RNN, gestapelte RNN, dichte RNN und tiefe RNN-Architekturen und -Kombinationen dieser untersucht. Details zu diesen Architekturen sind in Abschnitt 3.7.5 beschrieben. Da die FRFNN-Architektur schneller konvergiert, stabiler trainiert und die höchste Posengenaugigkeit erzielt, werden nur Ergebnisse von FRFNN in Abschnitt 7.8 berichtet. Details zur Parameterstudie werden in Abschnitt 7.6.3 erörtert. Es ist zu beachten, dass das RNN-Zellkonzept durch verschiedene rekurrente Zellkonzepte, wie z. B. LSTM, GRU und IRNN, ersetzt werden, um das optimale Zellkonzept (LSTM) abzuleiten.

7.6 Experimente

Abschnitt 7.6.1 beschreibt die Infrastruktur des Versuchsaufbaus. Anschließend wird der Datensatz erläutert (Abschnitt 7.6.2). Abschließend zeigt Abschnitt 7.6.3 die Parametrisierung der FRFNN- und der KF-Methoden.

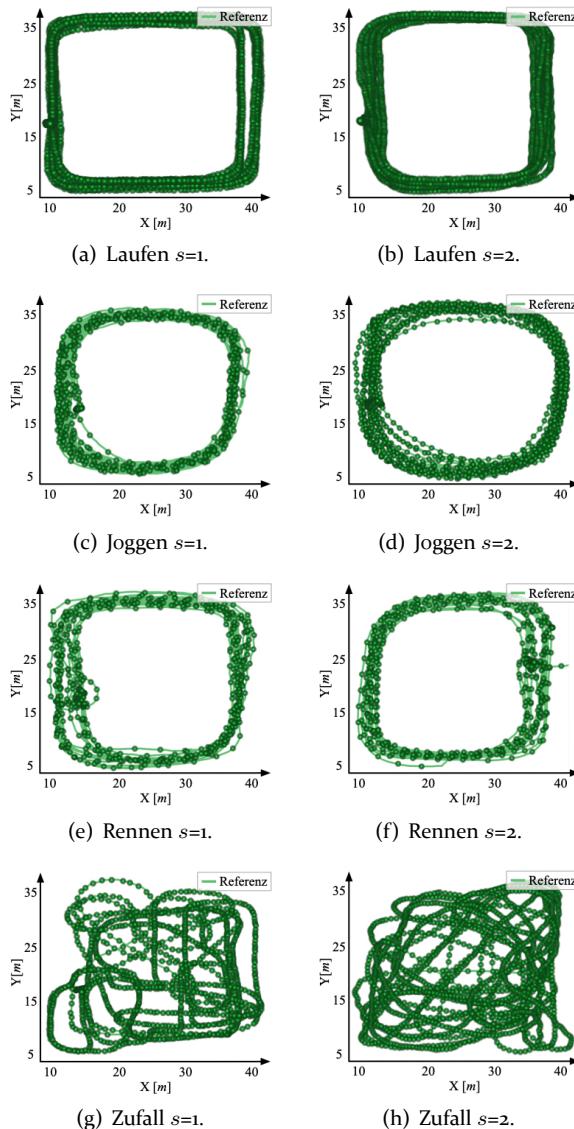


Abbildung 7.5: Exemplarische Referenztrajektorien des Segmente Bündels (jeweils zwei Segmente s der Aktivitäten Laufen, Joggen, Rennen und Zufall).

7.6.1 Messinfrastruktur

Da die Daten für die Posenschätzung aus der Datenerfassung der Geschwindigkeitsschätzung stammen, wird der Messbereich, das Referenzsystem, das System zur Messung der Funklokalisierung, die IMU-Messgeräte, die IMU-Software, die Zeitsynchronisation der Systeme und das verwendete Rechensystem in Kapitel 5.6.1 detailliert beschrieben. Zusätzlich werden die Mess- und Referenzsysteme der Positions schätzung explizit in Kapitel 4.6.2 beschrieben, während die Systeme zur Orientierungsschätzung in Kapitel 6.7.1 erläutert werden.

7.6.2 Datensatz

Die Datenerfassung wird bereits in Kapitel 5.6.2 detailliert beschrieben. Die zugrundeliegenden Datensätze werden in Kapitel 5.6.3 detailliert eingeführt. In der Studie zur Geschwindigkeitsschätzung (siehe Kapitel 5) wird festgestellt, dass Schiebefenstern der Größe 128 Hz und einer Überlappung von 50% ($N_w \cdot 0.5=64$) genügend herausragende Merkmale bei geringen Rechenkosten enthalten, um die höchste Genauigkeit der Geschwindigkeitsschätzung zu erreichen. Das bedeutet, dass für die Geschwindigkeitsschätzung bereits 1.28 s langfristige Beziehungen menschlicher Bewegung abdecken. Zu beachten ist, dass diese Parameter die besten Ergebnisse für die Varianten V_1 , V_2 und V_3 des Datensatzes liefern.

Im Rahmen der Experimente zur Posenschätzung in Abschnitt 7.8 werden zusätzlich Schiebefenster im Intervall von 0.5 s bis 30 s und Überlappungen von einem Zeitschritt bis 100% ($N_w \cdot 1.0=128$) untersucht. Zu beachten ist, dass jedem Zeitpunkt eines Schiebefensters und natürlich auch jedem Schiebefenster (dem letzten Zeitpunkt) eine Referenzposition, -orientierung und Funkposition zugeordnet sind. So können datengetriebene Verfahren lernen Eingabesequenzen, sowohl auf einzelne Referenzpositionen als auch auf eine Referenztrajektorie (=Sequenz aus Referenzpositionen) abzubilden. Die Statistik des Datensatzes zur Auswertung der Posenschätzer in Abschnitt 7.7 wird in Tabelle 7.1 beschrieben.

Der Orientierungsschätzer aus Kapitel 6, mit der kubischen Stützvektormaschine, liefert korrekte Orientierungen für V_2 in 96.7 von 100 Fällen ($SD = 4.01^\circ$) und für V_3 in 98.3 von 100 Fällen korrekte

Orientierungen ($SD = 3.89^\circ$). Der Positionsschätzer aus Kapitel 4 liefert Positionen mit einem Fehler von $MAE=0.1731$ m ($SD = 0.031$ m) für V_2 und einem Fehler von $MAE=0.1477$ m ($SD = 0.028$ m) für V_3 .

Vorstudien zeigten, dass die Skalierung der Zielvariablen ref keinen Einfluss auf die Schätzgenauigkeit hat, allerdings erhöhte die Skalierung der Eingangsvariablen für das Training die Genauigkeit.²

7.6.3 Parametrisierung der Posenschätzer

Die Vergleichsstudie der Dissertation bewertet zwei verschiedene Posenschätzer, den naiven modellbasierten PDR-Ansatz in Abschnitt 7.7 und den datengetriebenen Ansatz FRFNN in Abschnitt 7.8. FRFNN wird zusätzlich in einigen Experimenten mit einem optimierten KF verglichen. Da der modellbasierte PDR-Ansatz keine Parametrisierung erlaubt, wird im Folgenden nur die Parametrisierung des datengetriebenen Posenschätzers und des KFs, erläutert.

FRFNN

Um die optimale Architektur und ihre Parameter für die FRFNN Methode zu finden, wurde eine Rastersuche durchgeführt.³ Dabei wurden auch verschiedene Zellkonzepte und Konnektivitätskonzepte durchsucht. Bei der Suche wurden alle Zellkonzepte und Konnektivitätskonzepte identisch mit Bezug zur Anzahl der verdeckten Zwischenschichten und die Anzahl der Zellen untersucht. Die Schichten wurden an die Eingabe- und Ausgabedimensionen angepasst, die je nach Verfahren die höchste Genauigkeit liefern. Die verschiedenen Abtastraten (50, 100, 200 und 400 Hz) und Fensterlängen (64, 128, 256 und 512 Hz)

²RNN-Zellkonzepte verwenden $tanh(x)$ -Aktivierungsfunktionen, die eine Skalierung der Daten erfordern, siehe Abschnitt 3.4.6.

³FRFNN Rastersuchparametern: Löser ∈[SGD, Adam, Adadelta und RMSProp]; $\beta_1, \beta_2=0.01$; Momentum=0.9; FF/CNN/TCN/RNN/IRNN/GRU/LSTM/BLSTM Schichten=3 [1, 2, 3, 4, 8 und 16]; Zellen pro Schicht=120 [1:1:50 und 50:10:500]; Anfängliche Lernrate (LR)=0.001 ∈[1.0:0.1:0.0001]; LR Lernrate senken alle n Perioden $n \in [0, \text{textbf{10}}, 50, 100]$ Epochen; LR Lernrate Senkrate je Periode=0.5; Stapelgröße ∈[128, 256, 512, 1024, 2048]; Regularisierung ∈[L₁, L₂, Huber-Loss, Log-Cosh Loss]; Parameter (gesamt) ∈[100, 1.000, 10.000, 100.000 und 1.000.000]; Mischen ∈[Nein,pro Epoche]; Gradientenbeschränkung=max(Eingabe); Verwerfschichten: [vor oder nach den RNN Schichten in der Mitte]; Rate der Verwerfschicht = 0.5 ∈ [0.1 : 0.1 : 0.7]; Für jede Kombination von f_s (50, 100 200, 400) und N_w (64, 128, 256, 512). Die Parameter, die die höchste Genauigkeit erzielten werden mit fetten Zahlen dargestellt.

wurden auf die unterschiedlichen Eingabevarianten (Kombinationen aus p_{radio} , v , acc und θ_o) ausgewertet (beste Konfiguration in Fett-druck). Die Eingangssequenzlänge von $N_w=128$ enthält maximal 1.28 s (128/100 Hz) Bewegungsdaten. Jeder Parametersatz wurde für maximal 100 Epochen trainiert, wobei das frühzeitige Anhalten auf der Leistung bzw. Genauigkeit des Validierungssatzes basiert, um eine Überanpassung zu verhindern. Die resultierende beste Kombination von FF + LSTM + FF (mit 120 LSTM-Zellen), siehe Abschnitt 7.3, liefert die genauesten Posenschätzungen bei kurzen Inferenzzeiten.

Begründung der Parametrisierung

Optimierungsalgorithmen. Die Optimierungsalgorithmen Adam, Adadelta und RMSProp übertrafen die Variante des Standard Mini-Batch Stochastic Gradient Descent (SGD) deutlich. Zwischen Adam, Adadelta und RMSProp gibt es keine nennenswerten Unterschiede der Posengenauigkeit. Daher wurde für alle Experimente der Adam Optimierungsalgoritmus mit den Standard-Parametern ($\beta_1 = 0.9$, $\beta_2 = 0.999$ und $\eta = 10^{-8}$) [235] verwendet.

Lernratenanpassung. Die optimale initiale Lernrate ist 10^{-3} . Die Lernrate wurde um 0.5 reduziert, wenn sich der Trainingsverlust über 10 Epochen nicht signifikant verbessert hat, um die Konvergenz zu verbessern und die Lernratenanpassung fein abzustimmen.

Fehlerfunktionen. Es wurden verschiedene Fehlerfunktionen (L_1/MAE , L_2/MSE , Huber Loss/Smooth MAE, and Log-Cosh Loss) evaluiert. Da keine Fehlerfunktion die L_2 -Norm übertrifft, die im Wesentlichen der mittleren quadratischen Fehlerfunktion (engl. MSE) entspricht, wird diese für alle Experimente verwendet.

Modellgröße. Der wichtigste Einfluss auf die Modellierungskapazität neuronaler Netze ist die Anzahl der verwendeten Parameter [228]. Es wurden unterschiedliche Größen (100, 1.000, 10.000, 100.000 und 1.000.000) evaluiert. Eine Modellgröße von 100.000 Parametern liefert für die unterschiedlichen Architekturen die genauesten Posen.

Zellkonzepte. Die folgenden rekurrenten Zellkonzepte wurden verglichen: Standard RNN [109], IRNN [254], LSTM [191] und GRU [76]. Alle ausgewählten Zellen wurden in mehreren früheren Arbeiten

untersucht [81, 155, 254]. LSTM liefert die genauesten Posen.

Konnektivitätskonzept. Es wurden verschiedene Konnektivitäten zwischen Zellkonzepten untersucht: gestapelte RNNs [127], die die Ausgabe jeder Schicht an die Nächste weitergeben; dichte RNNs [153], die Verbindungen überspringen, damit das Netz die Netztiefe selbst auswählen kann und BLSTMs [128], die die Informationen sowohl aus der Zukunft als auch aus der Vergangenheit betrachten. Da die Modellkombination FF + LSTM + FF mit 100.000 Parametern die genauesten Posen schätzt, wird diese in Abschnitt 7.8 diskutiert.

Verborgene Zellen. Es wurde untersucht, wie sich die Anzahl an verborgenen Zustandseinheiten auf die Posengenauigkeit auswirkt (1 bis 50 mit einer Schrittänge von 1; und 50 bis 500 mit einer Schrittänge von 10). Damit ein Modell plausible Schätzungen liefert, sind mindestens 14 verborgene Zellen erforderlich. Mehr verborgene Zustandseinheiten führen zu genaueren Modellen. Modelle mit einer größeren verborgenen Zustandsgröße haben einen kleineren Vorhersagefehler. Eine optimale Leistung wird bei 120 verborgenen Zuständen erreicht. Mehr Zellen verbessern die Posengenauigkeit nicht.

Netztiefe. Es wurde die Auswirkung der Anzahl (1, 2, 3, 4, 8 und 16) der Schichten auf die Posengenauigkeit untersucht. Die Anzahl ist ein Schlüsselfaktor, um komplexere Verhaltensweisen zu modellieren, indem eine Hierarchie immer komplexerer Funktionen erstellt wird [180, 255]. Flachere Netze (weniger Parameter) ermöglichen eine geringere Latenz, eine bessere Parallelisierung und eine einfachere Modellstruktur. Sehr tiefe Modelle (über 8 Schichten) konvergieren nicht. FRFNN mit 3 Schichten und Verwerfungsschicht (FF + LSTM + D + FF) liefert die genauesten Posen.

Kalman-Filter

Es wird ein lineares KF-Modell aus dem Stand der Technik verwendet. Diese Modelle gelten unter der Annahme driftfreier Signale als optimale Zustandsschätzer [88]. Dieses Bayessche Modell beschreibt die Bewegungsübergangsfunktion (das Bewegungsmodell) und die Entwicklung des Messrauschen und das Prozessrauschen als lineare Funktionen, die durch Gaußsches Rauschen beeinflusst werden. Das KF Modell und seine Funktionsweise werden detailliert in

Abschnitt 3.5.1 beschrieben.

Um das KF auf den jeweiligen Trainingsdaten zu optimieren, wird KF zunächst mit einem Startzustand $x_0=0$, einer Kovarianz $P=I$, Prozessrauschen $Q=0.1$, Messrauschen $R=\sigma^2=0.1$ und der Übergangsfunktion = konstante Geschwindigkeit modelliert. Da empirische Kenntnisse über die Daten vorhanden sind (m.a.W. der Trainingsdatensatz), wird der KF optimal auf den jeweiligen Trainingsdatensatz konfiguriert. So passt die jeweilige Konfiguration optimal zu den Eingabedaten, und das jeweilige KF Modell ist dann ein optimaler Schätzer für den jeweiligen Trainingsdatensatz. Dazu wird das KF basierend auf den Vorhersagen p_{radio} , v und (optional) acc wie folgt parametrisiert. Die Kovarianzen des KF des Messrauschens und des Prozessrauschen werden anhand der Vorhersagen von p_{radio} , v und (optional) acc der Trainingsdaten optimiert, bis das KF eine optimale Leistung für die Validierungsdaten p_{radio} , v und (optional) acc liefert und somit optimierte Posen auf unbekannten Testdaten vorhersagen kann.

7.7 Auswertung der modellbasierten PDR Methode

Abschnitt 7.7.1 führt Metriken ein, die zur Bewertung der Trajektorienrekonstruktion verwendet werden und liefert Vorabinformationen. Anschließend werden allgemeine Erkenntnisse berichtet (Abschnitt 7.7.2). Dann werden die Auswirkungen der Geschwindigkeit auf die Posenschätzung (Abschnitt 7.7.3) und die Auswirkungen der Neukalibrierung (Abschnitt 7.7.4) auf die Posenschätzung berichtet. Abschließend fasst Abschnitt 7.7.5 die Erkenntnisse zusammen.

Während die Fusion der modellbasierten PDR-Methode (nur) von der Genauigkeit der Geschwindigkeitsschätzung (aus Kapitel 5) abhängt, hängt die datengetriebene Methode dagegen von allen Eingabeveriablen ab. Daher stellen sich zu den jeweiligen Methoden unterschiedliche Forschungsfragen, die von den jeweiligen Methoden nur explizit durch unterschiedliche Experimente beantwortet werden können. Deshalb wird die Posenschätzung unabhängig in zwei Phasen ausgewertet. Dieser Abschnitt 7.7 berichtet die Ergebnisse der modellbasierten PDR-Methode unabhängig von Abschnitt 7.8, der die Ergebnisse der datengetriebenen Methode berichtet. Abschnitt 7.9 ver-

Tabelle 7.1: Statistik des Datensatzes zur Auswertung der Posenschätzer (der Datensatz wird in Kapitel 5 Abschnitt 5.7.1 beschrieben).

Name	Prob.	Total [#]	Training [#]	Valid. [#]	Test [#]	Dauer	Distanz	v_{ref} [m/s]
	[#]	Segment Fenster/ Merkmal	Segment Fenster/ Merkmal	Segment Fenster/ Merkmal	Segment Fenster/ Merkmal	[min]	[km]	\emptyset min max
Genauigkeit								
V1	20	160 112.500	112 78.750	16 11.250	32 22.500	1.203	185.40	2.5 0.8 7.9
V2	20	160 97.950	112 68.565	16 9.795	32 19.590	1.045	161.42	2.6 0.8 7.8
V3	20	- 31.200	- 21.840	- 3.120	- 6.240	666	71.9	1.8 0.8 3.6
Trajektorie								
Ausgelassene Prob.	2	16 9.795	- - -	- - -	16 9.795	105	16.14	2.5 0.7 3.7

gleicht und diskutiert dann die Ergebnisse der beiden Auswertungen. Es ist zu beachten, dass beide Methoden unter Verwendung der gleichen Daten von den beiden ausgelassenen Probanden bewertet werden.

Da das modellbasierte PDR-Verfahren ein typisches PDR-Verfahren ist, hängt es direkt von der Genauigkeit der Geschwindigkeitsschätzung oder der daraus integrierten Entfernung ab (Abschnitt 7.7.3). Darauf werden die Auswirkungen der Geschwindigkeitsschätzer PDR, ML-GP, RoNIN, C/RNN und Hybrid aus Kapitel 5 auf die Posengenauigkeit des modellbasierten PDR Verfahrens in Abschnitt 7.7.3 detailliert untersucht. Im Prinzip wird die Position nur anfänglich benötigt, um das modellbasierte PDR-Verfahren im Weltkoordinatensystem auszurichten, und kann dann verwendet werden, um mögliche akkumulierte Fehler bei der Geschwindigkeitsschätzung zu kalibrieren (Abschnitt 7.7.4). Um die Genauigkeit der fünf Methoden der Geschwindigkeitsschätzung im Hinblick auf die Rekonstruktion der Pose oder der Trajektorie unter Verwendung von Geschwindigkeitsschätzungen und Referenzorientierungen explizit zu untersuchen, wurden sie unter Verwendung von Daten aus den beiden ausgelassenen Probanden bewertet.

7.7.1 Metriken

Nachfolgend werden Metriken zur Genauigkeitsbewertung der Trajektorienrekonstruktion aller Aktivitäten der ausgelassenen Probanden vorgestellt. Vor den Auswertungen wurden alle Methoden bestmöglich trainiert (PDR auf V2, andere auf V3). Die Methoden wer-

den in ihrer ursprünglichen Form bewertet. Nur Hybrid verwendet LKF.

Die Trajektorien werden basierend auf den Geschwindigkeitsschätzungen v' (oder v'' , da Hybrid einen LKF verwendet) und den Referenzorientierungen θ_{radio} des RTLS (siehe Abschnitt 5.6) rekonstruiert. θ_{radio} wird in Kombination mit der geschätzten Geschwindigkeit verwendet, um von polaren in kartesische Koordinaten umzuwandeln. Dann wird die Anfangsposition p_{ref} (die Referenzposition des RTLSs) addiert, um die geschätzte Position \hat{p} zu berechnen. Dieser Prozess richtet auch die geschätzte Position mit dem Koordinatensystem der Referenzdaten aus.

Für die beiden ausgelassenen Probanden, die nach den vier Aktivitäten (Gehen, Joggen, Laufen und Zufall) gruppiert werden, listet Tabelle 7.2 die sechs *Geschwindigkeitsfehlermetriken* auf, die aus der Genauigkeitsauswertung bekannt sind, und die Tabelle 7.2 listet ähnliche *Positionsfehlermetriken* (MAE, MSE, RMSE und CEP) auf: diesmal in [m] zwischen p_{ref} und \hat{p} . Die erste Gruppe von Zeilen in Tabelle 7.2 zeigt die Positionsfehler der fünf Methoden ohne Rekalibrierung (pur). Die zweite Gruppe von Zeilen zeigt die Fehler, wenn \hat{p} alle 30 Sekunden neu kalibriert wird (rekalibriert), d. h. wenn die geschätzte Position \hat{p} alle 30 s durch die aktuelle Referenzposition p_{ref} ersetzt wird.

Weiter werden sowohl die Referenz als auch die rekonstruierten Trajektorien in den TR-Abbildungen 7.6 bis 7.9 visualisiert, sowohl ohne (links) als auch mit Neukalibrierung (rechts). Die Grafiken zeigen Positionen mit einer Rate von ungefähr $2 \times$ pro Sekunde ($1.6 = 100 \text{ Hz} / (N_w / 2 = 64 \text{ Hz})$) und visualisieren ein Fragment von 180 Sekunden eines zufällig ausgewählten Probanden der zwei ausgelassenen Probanden, da sich der andere ausgelassene Proband ähnlich bewegt, siehe Abschnitt 5.7.4, beeinflusst dieser die Ergebnisse nicht.

Eine Vorstudie zeigte, dass die Neukalibrierung des aktuellen \hat{p} mit der Referenzposition die Positionsfehler für alle fünf Schätzer über alle Fehlermetriken hinweg verbessert (-283%). Kürzere Kalibrierungsintervalle führen zu genaueren Positionen. 1 s Intervalle reichen aus, um $\hat{p}=p_{ref}$ zu erreichen. Während für die f_s -Methoden Neukalibrierungsintervalle über 30 s keine Auswirkung hatten, verbesserten selbst 100 s Neukalibrierungsintervalle die Rekonstruktionsfehler von C/RNN und Hybrid. Für die Auswertung wurden daher 30 s-Intervalle ausgewählt,

Tabelle 7.2: Rekonstruktionsfehler der modellbasierten PDR Methode der Trajektorien in [m] der ausgelassenen Probanden. Die höchsten Genauigkeiten sind **fettgedruckt**.

Datensatz Ausgela- sene Prob.	PDR K_g						ML-GP						RoNIN						C/RNN						Hybrid					
	MAE	MSE	RMSE	CEP_{95}	MAE	MSE	RMSE	CEP_{95}	MAE	MSE	RMSE	CEP_{95}	MAE	MSE	RMSE	CEP_{95}	MAE	MSE	RMSE	CEP_{95}	MAE	MSE	RMSE	CEP_{95}	MAE	MSE	RMSE	CEP_{95}		
pur	Gehen	4.28	25.51	4.84	8.01	3.15	22.88	3.93	7.18	2.41	10.05	2.77	4.45	0.71	0.87	0.93	1.83	0.59	0.49	0.70	1.19									
	Joggen	8.56	98.15	9.81	15.52	3.93	22.86	4.52	7.44	6.38	54.64	7.36	11.39	1.17	2.76	1.49	3.04	0.69	0.59	0.76	1.22									
	Rennen	9.09	120.54	9.84	14.42	5.70	42.05	6.44	9.94	2.74	12.65	3.55	6.61	1.20	2.18	1.34	2.06	0.95	1.36	1.13	2.07									
	Zufällig	4.50	34.32	5.38	9.31	2.88	13.56	3.56	6.71	1.87	5.77	2.35	4.67	1.52	4.11	1.86	3.98	0.92	1.37	1.12	2.16									
rekalibriert	Gehen	2.38	9.42	3.04	5.88	1.67	6.61	2.38	4.84	1.47	3.22	1.79	3.54	0.57	0.54	0.74	1.60	0.36	0.23	0.47	0.93									
	Joggen	2.36	10.92	3.19	6.89	2.17	6.91	2.63	4.55	1.67	4.98	2.17	4.36	0.68	0.88	0.89	2.02	0.34	0.20	0.44	1.00									
	Rennen	3.13	18.18	4.14	8.43	2.23	8.97	2.97	6.79	1.73	5.08	2.24	4.39	0.75	0.85	0.92	1.89	0.33	0.22	0.46	1.00									
	Zufällig	2.58	15.11	3.54	7.70	1.77	5.92	2.39	5.28	1.54	4.11	3.14	6.01	0.92	1.49	1.17	2.50	0.49	0.52	0.66	1.25									

da diese auch typische Anwendungsfälle nachahmen, die entweder GPS oder ein RTLS mit einer niedrigen Aktualisierungsrate verwenden und Rechen- und Kommunikationslast einsparen.⁴

7.7.2 Allgemeine Erkenntnisse

Die allgemeinen Erkenntnisse der Rekonstruktion der Trajektorien sind: (1) Für keine der Methoden besteht eine direkte Korrelation zwischen hohen Geschwindigkeitsfehlern (Tabelle 7.2) und hohen Positionsfehlern (Tabelle 7.2), z. B. gibt es für die Aktivität Zufällig hohe Geschwindigkeitsfehler für alle Methoden, während die Positionsfehler entweder für PDR, ML-GP und RoNIN niedrig oder für C/RNN und Hybrid quasi linear sind. (2) Bei allen Aktivitäten macht Hybrid die kleinsten Positionsfehler, gefolgt von C/RNN. PDR liefert die schlechtesten Ergebnisse, siehe Tabelle 7.2. Diese Reihenfolge ist analog zu der Reihenfolge für die Genauigkeit der Geschwindigkeitsschätzer. (3) Je kleiner die Positionsfehler in Tabelle 7.2 sind, desto besser ist die Übereinstimmung zwischen den MAE-, CEP_{95} -, MSE- und RMSE-Werten und damit der Abstand zwischen den geschätzten Positionen und der Referenz, auch ohne Neukalibrierung (siehe die TR-Abbildungen 7.6 bis 7.9). Bei schwerwiegenden Ausreißern (hohe MSE oder RMSE Werte) wird die rekonstruierte Trajektorie für PDR, ML-GP und RoNIN deformiert, sichtbar durch starke Abweichungen zwischen den Formen der Trajektorien (siehe

⁴Weniger Kommunikationslast bedeutet eine höhere Kapazität für die Lokalisierung, d. h. mehr lokalisierbare Teilnehmer im Funksystem.

7 Posenschätzung

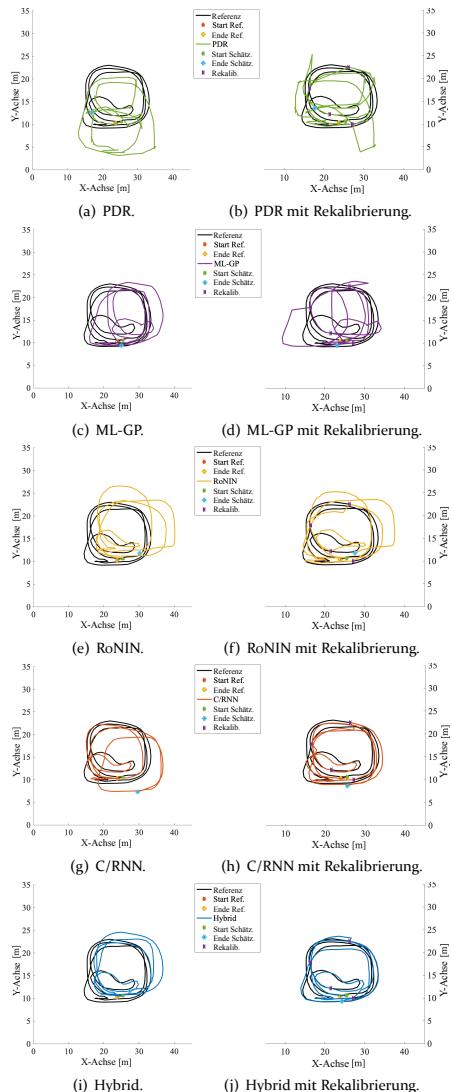


Abbildung 7.6: Draufsicht auf rekonstruierte Trajektorien; 3 min von Gehen des ausgewählten Probanden A (linke Spalte: pure Methoden; rechte Spalte: Methoden mit Neukalibrierung alle 30 Sekunden).

Tabelle 7.2). Die Trajektorien von C/RNN und Hybrid sehen deutlich besser aus, da die Formen der Trajektorien hier weniger von der Referenz abweichen. (4) Die Geschwindigkeit des Probanden wächst von den TR-Abbildungen 7.6 zu 7.8; und mit zunehmender Geschwindigkeit driften⁵ die rekonstruierten Trajektorien mehr von der Referenz ab.

Unerwarteterweise behandeln alle Methoden das kompliziertere Muster der Aktivität *Zufällig* recht gut. PDR, ML-GP und RoNIN erzielen geringere Positionsfehler als beim Joggen und Rennen. C/RNN und Hybrid zeigen eine bessere Leistung. (5) Bei allen Methoden halbiert eine Neukalibrierung die Positionsfehler über alle Metriken für alle Aktivitäten hinweg fast. Der Vergleich der Zeilen *pur* mit den Zeilen *rekalibriert* in der Tabelle 7.2 und den Spalten der TR-Abbildungen 7.6 mit 7.9 verdeutlicht dies.

7.7.3 Auswirkung der Geschwindigkeit

Zwischen den linken Spalten der TR-Abbildungen 7.6 bis 7.9 variiert die Bewegungsgeschwindigkeit. Alle Methoden liefern die genaueste Rekonstruktion der Trajektorie beim Gehen. Aber selbst für eine so langsame Aktivität rekonstruieren PDR, ML-GP und RoNIN Trajektorien, die von der Referenz abweichen, siehe TR-Abbildung 7.6 (links), während Hybrid nahezu fehlerfrei rekonstruieren kann.

In Tabelle 7.2 verbessern sich die Fehler und Ausreißer von PDR zu Hybrid. Je größer die Ausreißer sind, desto weiter von der Referenz entfernt ist die rekonstruierte Trajektorie. Für höhere Geschwindigkeiten bleibt die allgemeine Reihenfolge der Rekonstruktionsfähigkeiten der Methoden stabil (PDR ist am schlechtesten, Hybrid ist am besten). Der Drift ist bei PDR, ML-GP und RoNIN stärker ausgeprägt als bei C/RNN und Hybrid, die beide praktisch keinen Drift aufweisen, siehe TR-Abbildungen 7.7 und 7.8. Beim Joggen driften PDR, ML-GP und RoNIN in der TR-Abbildung 7.7 (a-c) nach Südosten. Das weist darauf hin, dass sie zum Unterschießen (engl. undershoot) neigen, da die

⁵Drift: wird als kleiner Fehleranteil definiert, der sich im Laufe der Zeit (über längere Zeiträume) quasi linear ansammelt und zu Deformierungen der Trajektorien führt, bspw. spiralförmige Verformungen, respektive Abweichungen von der Referenz. Im Gegensatz dazu haben *driftfreie* Positionen über lange Zeiträume keine derartigen Auswirkungen, dennoch können auch hier über kurze Zeiträume Sprünge und Zick-Zack-Wackeln (durch hohe MSE und RMSE Werte) auftreten.

7 Posenschätzung

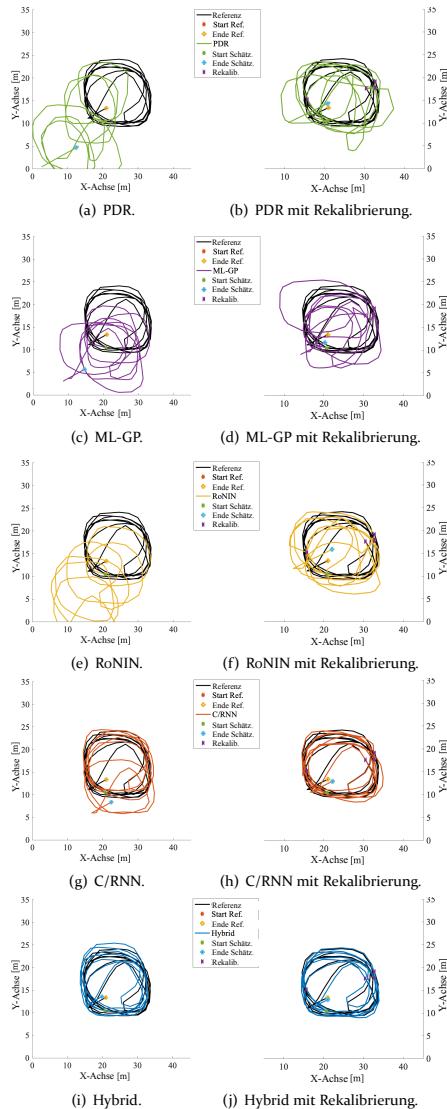


Abbildung 7.7: Draufsicht auf rekonstruierte Trajektorien; 3 min von *Joggen* des ausgelassenen Probanden A (linke Spalte: reine Methoden; rechte Spalte: mit Neukalibrierung alle 30 Sekunden).

Werte der geschätzten Positionen mit der Zeit kleiner werden als die der Referenz. Für die Aktivität *Rennen* driften ihre TR-Abbildungen 7.8 (a-c) nach Norden oder Osten und Überschießen (engl. overshooting), da ihre geschätzten Positionen größer werden.

Der K_g -Wert von PDR stellt das biomechanische Modell falsch dar, da für schnelle Bewegungen (Rennen und Zufällig) ein anderer Wert als beim Gehen erforderlich ist. Es gibt spiralförmige Rekonstruktionskurven, wenn PDR die Spitzen falsch klassifiziert. Bei ML-GP kann der Drift auf die schwache Entrauschungsleistung von ML-GP zurückzuführen sein, da bei höheren Geschwindigkeiten die IMU-Signale stärker verrauscht sind. Im Gegensatz zu PDR und ML-GP sind die Ergebnisse von RoNIN trotz des verrauschten Signals besser für die Aktivität *Rennen* als für die Aktivität *Joggen*, siehe Tabelle 7.2. Der Grund für den geringeren Drift kann sein, dass hohe Geschwindigkeiten zu mehr kurz- und langfristigen Rauschschwankungen (hohe häufige Schwingungen sehr hoher und niedriger Spitzen) im Signal führen und dass RoNIN einen solchen Eingangsdatensatz besser als PDR und ML-GP entrauscht. C/RNN und Hybrid zeigen auch hier ihre Stärken, da sie offenbar lernen, die kurz- und langfristigen Drifts der Ausreißer (MSE- und RMSE-Werte) zu interpretieren. Während es in der C/RNN-Kurve nur eine einzige Nichtübereinstimmung gibt (TR-Abbildung 7.7(g)), gibt es für Hybrid kein einziges solches Artefakt (TR-Abbildung 7.7(i)).

Alle Methoden erzielen eine geringere Abweichung bei der Aktivität *Zufällig* als bei *Joggen* und *Rennen*. Dafür könnte es zwei Gründe geben: Erstens gibt es Segmente mit sehr langsamer Geschwindigkeit (bspw. Gehen), in denen alle Verfahren gute Rekonstruktionsfähigkeiten zeigen. Zweitens können die häufigen Geschwindigkeitsschwankungen helfen, wenn sie die Richtung ändern und die lineare Akkumulation des Drifts unterbrechen bzw. konstruktiv kompensieren. Trotz des geringen Drifts gibt es immer noch große Positionsfehler, siehe die Sprungpositionen von PDR, ML-GP und RoNIN in der TR-Abbildung 7.9. Die Geschwindigkeitsänderung mildert die Skalierungs- und Sprungpositionsfehler nicht. Im Gegensatz dazu, zeigen C/RNN und Hybrid nicht viel Drift und haben auch weniger Positionsfehler und scheinen daher am besten gelernt zu haben, Interferenzen zu unterdrücken.

7 Posenschätzung

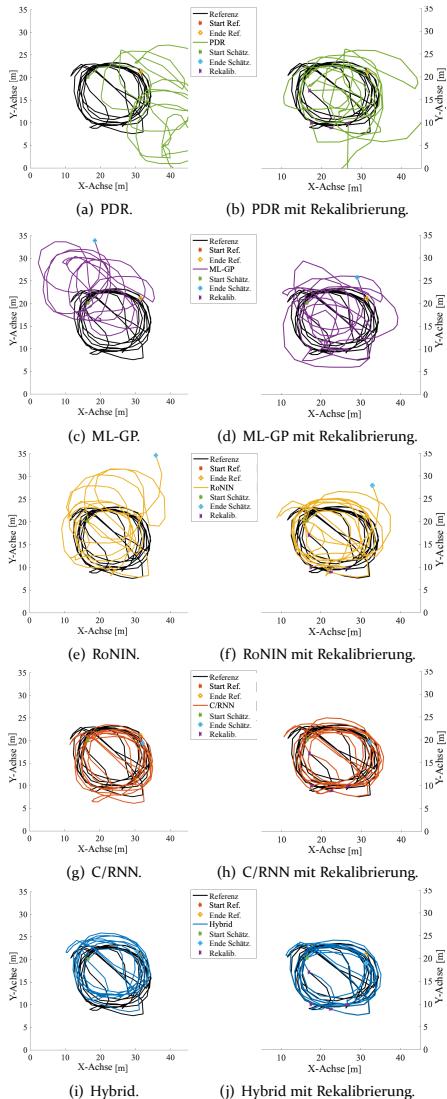


Abbildung 7.8: Draufsicht auf rekonstruierte Trajektorien; 3 min von *Rennen* des ausgelassenen Probanden A (linke Spalte: reine Methoden; rechte Spalte: mit Neukalibrierung alle 30 Sekunden).

7.7.4 Auswirkungen der Neukalibrierung

Die Auswirkungen der Neukalibrierung auf die Rekonstruktion der Trajektorie werden deutlich, wenn man die Trajektorienrekonstruktionen aller fünf Methoden mit und ohne Neukalibrierung in den TR-Abbildungen 7.6 mit 7.9 (linke und rechte Spalte) vergleicht. Dabei sind zwei allgemeine Aspekte zu beachten: Erstens ist ersichtlich, dass alle Positionsfehler in allen Metriken für alle Aktivitäten bei Verwendung der Neukalibrierung erheblich abnehmen. Im Allgemeinen hilft die Neukalibrierung am meisten, wenn die Geschwindigkeit langsam ist. Tabelle 7.2 zeigt, dass die Neukalibrierung die Positionsfehler für PDR, ML-GP und RoNIN fast halbiert, während C/RNN und Hybrid weniger Auswirkungen haben. Zweitens ist ersichtlich, dass PDR, ML-GP und RoNIN, die ohne Neukalibrierung am meisten drifteten, am meisten davon profitieren (vergleiche die TR-Zahlen 7.7(a, c, e) mit 7.7(b, d, f)). C/RNN und Hybrid ohne Neukalibrierung nähern sich bereits stark der Referenz an, sodass eine Neukalibrierung einen geringeren Zuwachs der Positionsgenauigkeit bewirkt.

Während die Neukalibrierung die Trajektorienrekonstruktion von PDR, ML-GP, C/RNN und Hybrid für die Aktivität *Zufällig* signifikant verbessert, zeigt RoNIN immer noch schwerwiegende Positionsfehler, siehe TR-Abbildung 7.9. Dies könnte auf die Tatsache zurückzuführen sein, dass RoNIN die Abhängigkeiten verschiedener Bewegungen und Geschwindigkeiten möglicherweise nicht ausreichend lernt und daher große Ausreißer falsch verarbeitet, was dann zu Positionssprüngen führt. Da C/RNN und Hybrid mit und ohne Neukalibrierung am besten funktionieren, scheinen DL-basierte Methoden eine Vielzahl von Geschwindigkeiten besser zu handhaben als die anderen Methoden. Interessanterweise ist anhand der Leistung von Hybrid für die Aktivität *Zufällig* (vergleiche die Trajektorie von *Start Schätz.* bis *Ende Schätz.* in der TR-Abbildung 7.9(j)) ersichtlich, dass Hybrid alle plötzlichen Orientierungsänderungen einwandfrei bewältigt, da es weder extreme Positionssprünge noch einen Drift gibt.

7.7.5 Zusammenfassung

Je schneller die Geschwindigkeit ist, desto mehr Drift sammelt sich in einer rekonstruierten Trajektorie an. Die Positionssprünge, das Wackeln und der Jitter sind bei vielen Richtungsänderungen stärker ausge-

7 Posenschätzung

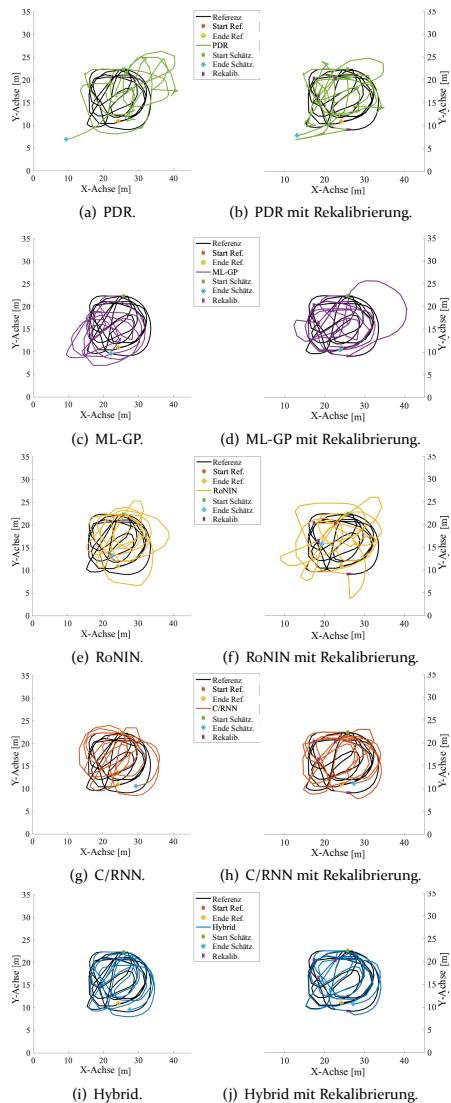


Abbildung 7.9: Draufsicht auf rekonstruierte Trajektorien; 3 min von Zufällig des ausgelassenen Probanden A (linke Spalte: reine Methoden; rechte Spalte: mit Neukalibrierung alle 30 Sekunden).

prägt. Daher funktioniert die Rekonstruktion der Trajektorie am besten, wenn sich die Geschwindigkeit ändert, um sich von schnellen Bewegungen und Driften zu erholen, und wenn sich nur wenige Richtungsänderungen ergeben. Durch das Anwenden von Neukalibrierungen werden die geschätzten Trajektorien immer näher an die Referenz approximiert, selbst für C/RNN und Hybrid, die ohne Neukalibrierung bereits gut sind. C/RNN und Hybrid scheinen drei wesentliche Dinge zu lernen: (1) wie man mit unterschiedlichen Bewegungsgeschwindigkeiten und dem entsprechenden Signalrauschen umgeht, da alle Fehlermetriken für alle Aktivitäten niedrig sind; (2) wie man die Skalierung der Geschwindigkeit korrekt interpretiert, da sie genaue Positionen zeigen selbst bei schnellen Bewegungsänderungen; und (3) wie man die Eingaben am besten entrauscht, da sie im Durchschnitt kleinere MSE- und RMSE-Werte erreichen als die anderen Methoden.

7.8 Auswertung der datengetriebenen Methode

Abschnitt 7.8.1 liefert Vorabinformationen zum Studienkonzept und allgemeingültigen Eigenschaften der Studie. Abschnitt 7.8.2 führt Metriken ein, die zur Bewertung der Trajektorienrekonstruktion verwendet werden. Anschließend werden die Auswirkungen der Eingabevariationen (Abschnitt 7.8.3), die Auswirkungen des Vorhersegehorizonts (Abschnitt 7.8.4), die Auswirkungen der Sequenzlänge (Abschnitt 7.8.5), die Auswirkungen dynamischer Bewegungen (Abschnitt 7.8.6), die Auswirkungen unbekannter Trajektorienformen (Abschnitt 7.8.7) und die Auswirkungen von Datenlücken (Abschnitt 7.8.8) auf die Genauigkeit der Posenschätzung berichtet. Abschließend fasst Abschnitt 7.8.9 die Erkenntnisse zusammen.

Die datengetriebene Methode hängt nicht nur von der Genauigkeit der Geschwindigkeitsschätzung ab, sondern auch davon, dass die Posen in jedem Zeitschritt in Abhängigkeit von verschiedenen Eingangsvariablen optimal zusammengeführt werden. In diesem Abschnitt wird der direkte Einfluss aller Eingangsvariablen untersucht und wie diese optimal, durch verschiedene Trainingskonfigurationen, fusioniert werden, sodass die höchsten und zuverlässigsten Posenschätzungsergebnisse erzielt werden.

7.8.1 Vorabinformationen

Studienkonzept

Im Folgenden wird festgestellt, dass sich die Rekonstruktion der Pose und Trajektorie mit datengetriebenen Methoden sehr verbessern, wenn die Geschwindigkeit als zusätzliche Information in ein Modell eingeführt wird. Da die Geschwindigkeit bereits in Abschnitt 7.7 und Kapitel 5 ausführlich besprochen wird, werden in diesem Abschnitt Probleme behandelt, die für datengetriebene Verfahren spezifisch sind und deren Unterschiede, Möglichkeiten und Schwächen im Vergleich zum modellbasierten PDR Verfahren hervorgehoben.

Die Berichte der Experimente (Abschnitt 7.8.3 bis 7.8.8) sind folgendermaßen strukturiert: zu Beginn wird über den Hintergrund jedes Experiments informiert, dann wird die Konfiguration des jeweiligen Experiments beschrieben, die Auswertung diskutiert und wesentliche Ergebnisse zusammengefasst.

Konfiguration

Für jedes Experiment wird die jeweilige Konfiguration (Trainings- und Testparameter) angegeben. Wenn keine expliziten Informationen angegeben sind, werden die Daten für die Experimente wie in Abschnitt 7.6.2 und 5.6.2 beschrieben vorverarbeitet. Die Positions-, Geschwindigkeits- (Hybrid verwendet ein LKF) und Orientierungsschätzer werden a priori bestmöglich auf Datensatz V₃ trainiert und optimiert. Die Methoden werden auf den Daten der beiden ausgelassenen Probanden bewertet. Die Zusammensetzung der Trainings-, Validierungs- und Testdaten ist in Tabelle 7.1 aufgeführt.

Da die Positions-, Geschwindigkeits- und Orientierungsschätzer unterschiedliche Fensterüberlappungen und Aktualisierungsraten aufweisen können, können ihre Datenraten variieren. Daher werden die Eingabedatenströme dieser Schätzer linear auf eine einheitliche Datenrate und Sequenzlänge abgetastet. Um die Auswirkung von Datenlücken auszuwerten werden die Daten in Abschnitt 7.8.8 explizit nicht interpoliert. Sofern keine andere Konfiguration angegeben ist, werden die Ergebnisse für die folgende Konfiguration verwendet (**dicke** Schrift kennzeichnet die Werte, die die genaueste Pose erzielten): Sequenzlänge (N_w) = **128** Werte (mögliche Testwerte sind 64, 128, 256 und 512); Abtatsrate

(f_s) =**100** Hz; Variation der Reihenfolge der Eingabesequenzen = **1** (1 bedeutet, dass die Reihenfolge der Eingabesequenzen in jedem Stapel variiert wird, 0 bedeutet das Gegenteil); Vorrausblick = 0 Wert (Vorrausblick = 1 bedeutet bei einer Abtatsrate von 100 Hz werden Daten, die 10 ms=1/100 Hz in der Zukunft liegen vorhergesagt).

Orientierung

Die Referenzorientierung θ_{ref} wird direkt vom optischen Referenzsystem bereitgestellt. Da sich sowohl Sensoren des Funk- als auch des Trägheitssystems im Rahmen der Studie in einem Rotations- und Koordinatensystem befinden, kann die Orientierung auch direkt aus zwei aufeinander folgenden Positionen bestimmt werden θ_{radio} .

Allerdings ist zu beachten, dass der Abstand zwischen den Positionen mit dem Orientierungsfehler korreliert. Dies liegt daran, dass die Positionen einen Schätzfehler aufweisen, sodass die Start- und Endpositionen möglicherweise keine gerade Linie überspannen und somit keinen exakten Winkel der tatsächlichen Bewegung darstellen. Mit größerem Abstand kann der Positionsfehler reduziert werden. Je größer der Abstand, desto genauer wird die Orientierung, wenn sich der Sensor kontinuierlich in die gleiche Richtung bewegt. Im Gegensatz dazu ist der Orientierungsfehler umso größer, je kleiner der Abstand (je langsamer die Fortbewegung) ist. Abschnitt 6.9 diskutiert die Zusammenhänge quantitativ. Bei sehr langsamem Gehgeschwindigkeiten ($>1.25 \text{ m/s}$) liegt der durchschnittliche Orientierungsfehler bei 2.6° (maximaler Fehler von 4.1°), wenn die Schätzungen der Start- und Endpositionen sehr ungenau ($>\pm 0.20 \text{ m}$) sind.

In den folgenden Studien (siehe Abschnitt 7.8.3) wird festgestellt, dass die Orientierungsschätzung aus den Positionen θ_{radio} in der vorliegenden Datensituation (zwei Sensoren im gleichen Rotationssystem) geringere Schätzfehler ergibt als die θ_{ori} der Kalibrierungsmethode aus Kapitel 6. Da die Orientierung θ_{radio} quasi linear von der Schätzgenauigkeit der Position p_{radio} abhängt, werden Ergebnisse der Orientierungsschätzung in den Abschnitten 7.8.4 bis 7.8.8 nicht erläutert. Es ist selbsterklärend, dass das Verfahren zur Orientierungsschätzung θ_{ori} jederzeit zur Kalibrierung eines gemeinsamen Rotationssystems verwendet werden kann, aber immer verwendet werden sollte, wenn verschiedene Rotationssysteme verwendet werden.

7.8.2 Metriken

Im Folgenden werden die Metriken zur Bewertung der Genauigkeit der Trajektorienrekonstruktion für alle Aktivitäten der ausgelassenen Probanden vorgestellt. Vor den Auswertungen werden alle Methoden auf V₃ bestmöglich trainiert. Tabelle 7.1 führt die Statistik der verwendeten Datensätze auf. Abbildung 7.2 bietet einen Überblick über mögliche Eingabeveriablen der Segmente und Fenster der Eingabesequenzen. Zu beachten ist, dass θ_{ref} nicht explizit eingezeichnet wird. Die Trajektorien werden basierend auf den Positionsschätzungen p_{radio} (siehe Abschnitt 4.7), den Geschwindigkeitsschätzungen v' oder v'' , da Hybrid einen LKF verwendet (siehe Abschnitt 5.7) und den Orientierungen θ_{ori} (siehe Abschnitt 6.8) bzw. θ_{radio} des RTLS (siehe Abschnitt 5.6) rekonstruiert. Die Orientierung wird in Abschnitt 7.8.3 zusätzlich mit einer kubischen SVM und den vorverarbeiteten Merkmalen und Parametern (bestmöglich auf V₃ trainiert) geschätzt, siehe Kapitel 6. Im Rahmen der großangelegten Parameterstudie werden die Geschwindigkeits- und Orientierungsschätzungen für jede Testkonfiguration (bspw. $f_s \in [50, 100, 200, 300, 400, 500]$ und $N_w \in [64, 128, 256, 512]$) explizit trainiert.

Für die beiden ausgelassenen Probanden, die nach den vier Aktivitäten (Gehen, Joggen, Laufen und Zufall) gruppiert werden, diskutiert jedes Experiment die Ergebnisse anhand der Posenfehlermetriken (MAE, MSE, RMSE und CEP) in [m] zwischen p_{ref} und \hat{p} .

7.8.3 Auswirkung von Eingabeveriationen

In einer Vorstudie wird die Auswirkung der Ausgabeveriationen (p_{ref} , Geschwindigkeit v_{ref} , SMV der Beschleunigung acc_{ref} , und Orientierung θ_{ref} bzw. jede Kombinationen dieser) der Referenzpose auf die Posengenauigkeit untersucht. Es ist zu beachten, dass die Referenzgeschwindigkeit v_{ref} und die -beschleunigung acc_{ref} von der Referenzposition p_{ref} abgeleitet und nicht geglättet werden. Die Ableitungen der Referenzdaten führen keine neuen Informationen (andere Sensorcharakteristiken) in die Modelle ein und ermöglichen es, den Effekt der Eingabetransformation allein zu bewerten. Keine der Ausgabeveriationen hatte einen Einfluss auf die Modellgenauigkeit.

Tabelle 7.3: Rekonstruktionsfehler [m] der FRFNN-Methode der Trajektorien der ausgelassenen Probanden mit variierenden Eingabedatenströmen.

Eingabe-variation	FRFNN			
	MAE	MSE	RMSE	CEP ₉₅
p_{radio}, v	0.0375	0.0141	0.0027	0.0991
p_{radio}, acc	0.2722	0.0286	0.0421	0.4356
p_{radio}, θ_{ori}	0.2156	0.0198	0.0375	0.2911
p_{radio}, v, acc	0.1144	0.0243	0.0324	0.2007
$p_{radio}, v, \theta_{ori}$	0.1377	0.0212	0.0371	0.2289
$p_{radio}, acc, \theta_{ori}$	0.3481	0.0463	0.1269	0.4108
$p_{radio}, v, acc, \theta_{ori}$	0.1962	0.0342	0.0452	0.2577

Daher werden hier nur die EingabevARIABLEN ($p_{radio}, v, acc, \theta_{ori}$ bzw. jede Kombinationen dieser) diskutiert. Um festzustellen, wie unterschiedliche EingabevARIATIONEN die Posengenauigkeit beeinflussen, werden alle Kombinationen von p_{radio}, v und acc in den Eingabesequenzen variiert. Abbildung 7.10 zeigt den Einfluss der verschiedenen Eingangsvariationen auf die Positionsgenauigkeit von FRFNN.

Konfiguration

Die Konfiguration ist in Abschnitt 7.8.1 definiert. Die Ergebnisse des optimierten KFs werden nur für das beste Modell (p_{radio}, v) berichtet.

Auswertung

Einfluss von v : In verschiedenen Modellen hat die Eingabekombination (siehe Abbildung 7.10 von links nach rechts) einen bemerkenswerten Einfluss auf die Genauigkeit. Die Modelle mit p_{radio} und v als Eingabeparameter übertreffen die ohne v ($p_{radio}, acc, p_{radio}, \theta_{ori}, p_{radio}, acc, \theta_{ori}$) deutlich. Einfluss von acc : Das Einbeziehen der Beschleunigung acc führt generell zu einer schlechteren Leistung. Ein Experiment mit entrauschten (gleitendes Mittelwertfilter SMA=10#) Beschleunigungen liefert ebenfalls ungenaue Posen. Die Eingabekombination p_{radio}, v erzielt die höchste Positionsgenauigkeit ($MAE = 0.0375$, $MSE = 0.0141$, $RMSE = 0.0027$, $CEP_{95} = 0.0991$ in [m]). Die Kombination der Funkposition mit dem SMV der Beschleunigung p_{radio}, acc ergibt das schlechteste Ergebnis ($MAE = 0.2722$, $MSE =$

0.0286 , $RMSE = 0.0421$, $CEP_{95} = 0.4356$ in [m]) aller Kombinationen ohne θ_{ori} . Die Kombination der Funkposition mit der Geschwindigkeitsschätzung und dem SMV der Beschleunigung p_{radio}, v, acc führt ebenfalls zu schlechteren Ergebnissen ($MAE = 0.1144$, $MSE = 0.0243$, $RMSE = 0.0324$, $CEP_{95} = 0.2007$ in [m]) als p_{radio}, v . Die Tatsache, dass das Hinzufügen der Beschleunigung zu ungenauen Posen führt, kann darauf zurückzuführen sein, dass die Beschleunigung neben der Geschwindigkeit kaum zusätzliche Informationen liefert und in Kombination mit der Position kein Zusammenhang zwischen beiden besteht, da die Geschwindigkeit fehlt.

Das KF mit Bewegungsmodell mit konstanter Geschwindigkeit liefert signifikant genauere Posen als das Bewegungsmodell mit konstanter Beschleunigung, selbst wenn acc explizit hinzugefügt wird. Die Ungenauigkeit wird möglicherweise durch das Messrauschen verursacht, das für acc für das KF nicht korrekt für unbekannte Daten definiert werden kann. KF liefert wie FRFNN die höchste Genauigkeit, wenn es auf p_{radio}, v optimiert wird ($MAE=0.3244$, $MSE=0.0411$, $RMSE=0.0573$, $CEP_{95} = 0.6019$ [m]). Die Genauigkeiten sind jedoch deutlich niedriger als bei FRFNN, jedoch genauer als bei der modellbasierten PDR Methode ($MAE=0.4900$, $MSE=0.5200$, $RMSE=0.6600$, $CEP_{95} = 1.2500$ [m]). Das KF ist ein optimaler Schätzer für Daten für die es optimiert wird. Da sich die Daten der ausgelassenen Probanden vom Datensatz V₃ unterscheiden, liefert das KF höhere Fehler. Das konstante Geschwindigkeitsbewegungsmodell liefert deutlich genauere Posen als das Bewegungsmodell mit konstanter Beschleunigung, auch wenn acc explizit hinzugefügt wird. Die Ungenauigkeit kann durch das Messrauschen von acc verursacht werden, das für den KF nicht definierbar ist.

Einfluss von θ_{ori} : Das Hinzufügen der Orientierung zur Position ergibt generell schlechtere Ergebnisse. Auch die Kombination ($p_{radio}, v, \theta_{ori}$) ergibt schlechtere Ergebnisse ($MAE=0.1377$, $MSE=0.0212$, $RMSE=0.0371$, $CEP_{95} = 0.2289$ in [m]). Dies könnte daran liegen, dass der Trägheitssensor (zum Schätzen und Kalibrieren von θ_{ori}) im selben Rotations- und Koordinatensystem wie der Funksensor installiert ist und die Funkposition daher bereits implizit eine Orientierung oder Ausrichtung angibt. Ein Schätzfehler von θ_{ori} verfälscht möglicherweise die Zusammenhänge zwischen Position und Orientierung und verwirrt das datengetriebene Modell. Es ist daher ratsam, die Orientierungsschätzung nur dann explizit in die Eingangsvariation eines da-

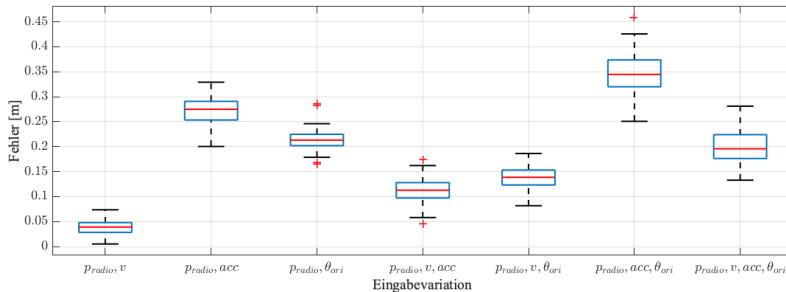


Abbildung 7.10: FRFNN-Poseabschätzgenauigkeit für verschiedene Eingabedaten. (MAE=rote Linie in der Box; Ausreißer: rotes +; und Standardabweichung).

tengetriebenen Modells zu integrieren, wenn ihre Fehlervarianz gering ist, d. h. wenn sie nur geringe Auswirkungen auf die Positionsgenauigkeit hat und der Positionsfehler aufgrund der Orientierung geringer ist als der des p_{radio} . Allerdings sollte die Orientierung explizit verwendet werden, wenn sich die Funk- und Trägheitssensoren in unterschiedlichen Rotations- und Koordinatensystemen befinden.

Zusammenfassung

Die Modelle mit p_{radio} und v als Eingabeparameter übertreffen die Genauigkeit aller anderen Eingabevereinstellungen signifikant ($MAE < 0.04$ m). Das Hinzufügen der Geschwindigkeit als explizite Eingabe führt also immer zu besseren Vorhersagen. Darüber hinaus ist es ratsam, die Orientierung nur dann in die Eingabesequenz aufzunehmen, wenn sie aus einem anderen Rotations- und Koordinatensystem als der Position stammt. Interessanterweise hat die Variation der Ausgabe, m.a.W. der Referenzen bzw. Zielvariablen Y , keine Auswirkung auf die Ergebnisse.

7.8.4 Auswirkung des Vorhersagehorizonts

Dieses Experiment variiert den Vorhersagehorizont zwischen 0.01 s und 3.0 s, um zu untersuchen, wie weit das FRFNN-Modell Posen in der Zukunft vorhersagen kann. Die Vorhersage wird erreicht, indem für jedes Schiebefenster für die Eingabe das zukünftige Fenster der zu untersuchenden Referenzpose ausgewählt wird, das die zukünftigen Re-

Tabelle 7.4: Rekonstruktionsfehler der FRFNN-Methode der Trajektorien [m] der ausgelassenen Probanden mit variierendem Prognosehorizont.

Konfiguration		FRFNN			
Vorhersagehorizont [s]	Sequenzlänge [s]	MAE	MSE	RMSE	CER ₉₅
0.0	0.64	0.0651	0.0227	0.0229	0.0919
0.0	1.28	0.0375	0.0141	0.0027	0.0991
0.0	2.56	0.1066	0.0341	0.0975	0.2356
1.0	0.64	0.0667	0.0243	0.0324	0.1319
1.0	1.28	0.0489	0.0076	0.0057	0.1074
1.0	2.56	0.1175	0.0546	0.0975	0.2356
2.0	0.64	0.4916	0.1966	0.2841	0.9835
2.0	1.28	0.3775	0.1745	0.2793	0.4745
2.0	2.56	0.5733	0.2981	0.4129	1.3213
3.0	0.64	8.4357	3.9453	-	-
3.0	1.28	3.8741	1.4765	-	-
3.0	2.56	10.3233	4.9989	-	-

ferenzwerte (zwischen 0.01 s und 3.0 s) enthält. Je weiter ein Modell die Pose in der Zukunft vorhersagen kann, desto mehr Rechenzeit und Systemlatenz kann es kompensieren. Während frühere Studien [15, 11] gezeigt haben, dass RNNs die Zukunft von Trajektorien in Bildverarbeitungsdomänen für mehrere Sekunden genau vorhersagen, wächst der Fehler dramatisch, wenn der Prognosehorizont 10 s [15] überschreitet.

Konfiguration

Die Konfiguration ist verändert, die Vorausschau (in s) wird variiert $\in [0.01, 0.1, 0.15, 0.25, 0.5, 0.75, 1.0, 2.0, 3.0]$. Jede Zielsequenz der Trainingdaten (V_3) und Testdaten der ausgelassenen Probanden wird relativ zur Eingabesequenz um die Zeit der Vorausschau verschoben.

Auswertung

Abbildung 7.11 zeigt die Posengenauigkeit in Bezug zum Vorhersagehorizont. Zur besseren Übersicht zeigt die Abbildung die Fehlervarianz nur für das beste FRFNN Modell und KF. Das KF hat grundsätzlich eine höhere Ungenauigkeit. KF verdreifacht den Fehler der Pose mit einer Vorhersage von 1 Sekunde in die Zukunft. Da das KF nur den letzten Zeitschritt zur Vorhersage des nächsten Zeitschritts nutzt, hat die Sequenzlänge keinen Einfluss auf die Genauigkeit

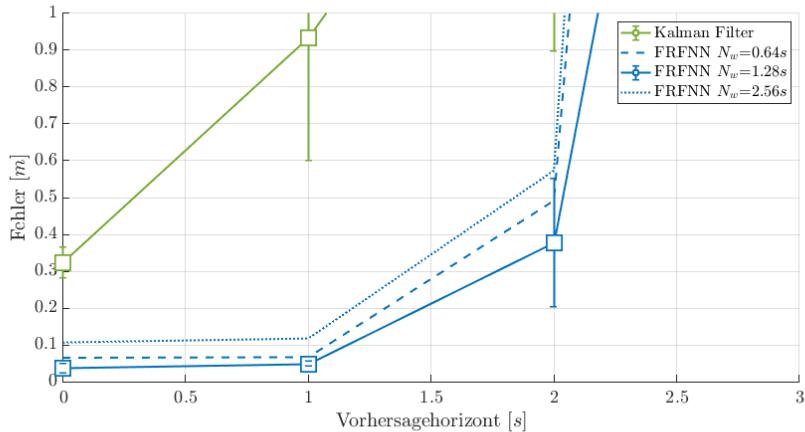


Abbildung 7.11: Posengenaugkeit von FRFNN (in [m]) über den Vorhersagehorizont (von 0 s bis 3 s) für verschiedene Sequenzlängen ($N_w \in [0.64, 1.28, 2.56] s$). Die Linien visualisieren den Fehler (MAE) und die Länge der vertikalen Linien der Boxen repräsentiert den Grad der Fehlervarianz.

des KF. Dies kann auch der Grund dafür sein, dass KF deutlich ungenauere zukünftige Posen vorhersagt, da die Informationen aus dem letzten Zeitschritt nur begrenztes Wissen über die Zukunft liefern.

Es ist deutlich erkennbar, dass FRFNN bei einer Sequenzlänge von 128 Zeitschritten und einem Prognosehorizont von 1 s die genauesten Vorhersagen liefert und mögliche Mess- und Systemverzögerungen bis zu einem Prognosehorizont von 1 s optimal kompensieren kann (1 s Prognosehorizont mit Sequenzlänge = 1.28 s liefert $MAE = 0.0489$, $MSE = 0.0076$, $RMSE = 0.0057$, $CEP_{95} = 0.1074$ in [m]). Kürzere (0.64 s) Sequenzlängen sind kaum ungenauer (1 s Prognosehorizont mit Sequenzlänge = 0.64 s liefert $MAE = 0.0667$, $MSE = 0.0243$, $RMSE = 0.0324$, $CEP_{95} = 0.1319$ in [m]). Im Gegensatz dazu wirken sich längere (> 1.28 s) Sequenzen negativ auf die Posengenaugkeit aus (1 s Prognosehorizont mit Sequenzlänge=2.56 s liefert $MAE = 0.1175$, $MSE = 0.0546$, $RMSE = 0.0975$, $CEP_{95} = 0.2356$ in [m]). Möglicherweise führen zu viele Informationen (z. B. mehrere Kreise und Spiralen) über die Vergangenheit, die in den langen Sequenzen enthalten sind, dazu, dass sich FRFNN zu sehr auf diese Details (z. B. Kreise statt Kurven

und Geraden) konzentriert und sich somit die Vorhersagegenauigkeit deutlich verschlechtert. Diese Erkenntnisse stimmen mit den Ergebnissen des Experiments in Abschnitt 7.8.5 überein, das die Auswirkung der Sequenzlänge auf die Kapazität des Kontextvektors der LSTM Zellen und deren Auswirkung auf die Posengenauigkeit überprüft. Es wird auch festgestellt, dass eine Prognose über 2 s zu einer signifikant schlechteren (2 s Prognosehorizont mit Sequenzlänge=1.28 s: $MAE = 0.3775$, $MSE = 0.1745$, $RMSE = 0.2793$, $CEP_{95} = 0.4745$ in [m]) und über 3 s zu keiner plausiblen Posengenauigkeit führt, weshalb Tests mit einem Prognosehorizont über 2 s verworfen werden. Prognosehorizonte <1 s zeigen keine Auffälligkeiten, der Fehler wächst fast linear mit der Breite des Prognosehorizonts.

In einem unabhängigen Experiment wird gezeigt, dass signifikant längere Eingabesequenzen (30 s) einfacher Kreisbahnen eine genaue Vorhersage der zukünftigen Pose von etwa 10 s in der Zukunft ermöglichen. Im Gegensatz dazu führen kurze (1.28 s) Eingabesequenzen einfacher Kreisbahnen zu erheblich größeren Fehlern, wenn sie Posen, die 10 s in der Zukunft liegen, prognostizieren müssen. Dies könnte daran liegen, dass die langen Eingabesequenzen die einfachen Kreisbewegungen vollständig enthalten und praktisch speichern können, während die kurzen Eingabesequenzen nur einen Bruchteil des Kreises lernen können. Das Weltwissen über Kreisbewegungen fehlt in den kurzen Eingabesequenzen, und daher kann FRFNN mit kurzen Eingabesequenzen nicht 10 s in die Zukunft schauen.

Interessanterweise scheint der Kontextvektor des LSTM über genügend Speicherkapazität für einfache Kreisbewegungsformen zu verfügen, um diese Formen mit langer Trajektorien zu lernen. Es besteht also möglicherweise eine starke Korrelation zwischen Genauigkeit, Sequenzlänge (Informationsdichte), Kapazität der LSTM Zellen und Vorhersagehorizont. Erkenntnisse über den Effekt der Sequenzlänge werden in Abschnitt 7.8.5 diskutiert.

Zusammenfassung

FRFNN sagt zukünftige Posen von komplexen Bewegungen mit einem Fehler von weniger als 0.05 m (1 s) voraus. Diese Ergebnisse sind vergleichbar mit denen von Altche et al. [15]. Da das Funksystem nur eine durchschnittliche Bewegung zu Photon (engl. Motion To Photon,

MTP) Latenz⁶ von durchschnittlich 246.34 ms (N = 87, min. 198.7 ms, max. 319.6 ms, SD 76.4 ms) und die Verzögerungen der Berechnungen der Position (FRFNN: 33 ms), Geschwindigkeit- (Hybrid: 46 ms)) und Orientierungsschätzungen (*SVM*³: 26 ms) noch niedriger sind, kann die Voraussicht (1000 ms) diese Verzögerungen leicht kompensieren und eine hohe Posengenauigkeit liefern.

7.8.5 Auswirkung der Sequenzlänge

Um die Speicherkapazität des Kontextvektors C zu untersuchen, wird FRFNN verwendet, das nur eine LSTM-Schicht aufweist, sodass der Kontextvektor nur von vorherigen (lateralen) Zeitschritten, jedoch nicht von niedrigeren Schichten beeinflusst wird, siehe Kapitel 3.7. Es ist zu beachten, dass C Informationen von jedem vorherigen Zeitschritt und jedem Zeitschritt der unteren Schicht sammelt, um eine endgültige Vorhersage zu treffen, d. h. bei längeren Sequenzen und tieferen Netzen muss er signifikant mehr Informationen sichern, d. h. die Anfangsinformation einer unteren Schicht verblasst immer mehr mit zunehmender Sequenzlänge und Modellgröße und kommt möglicherweise verblasst oder gar nicht im letzten Zeitschritt an.

Um zu untersuchen, was und wie das Modell zusätzlich zur Speicherkapazität lernt (im Folgenden Lernstrategien genannt), werden verschiedene Eingabe- und Ausgabesequenzlängen auf die Genauigkeit der Pose untersucht. Kurze Sequenzlängen decken somit nur Bruchteile einer Trajektorie ab, bspw. gerade Linien und Kurven einer Kreisbahn, und längere Sequenzen decken bspw. den gesamten Kreis ab. Abbildung 7.12 zeigt (links) eine kurze Sequenz und (rechts) eine längere Sequenz. Es ist offensichtlich, dass die längere Sequenz (komplexere Bewegung) selten oder gar nicht im gesamten Segment auftritt und daher nicht zur Rekonstruktion beiträgt. Im Gegensatz dazu können einige

⁶Bewegung zu Photon (MTP) ist die Latenz zwischen der physischen Translationsbewegung (z. B. die Kopfbewegung) und der entsprechenden visuellen Rückmeldung (Anzeige der Bewegung auf einem Bildschirm) [174, 275]. MTP beschreibt die Latenz (Verzögerung) der Positionsaktualisierungen. Die Positions-MTP wird durch Bildzählung gemessen [174]. Dazu werden gleichzeitig Sensorbewegungen und die daraus resultierende Bildschirmreaktion in wiederholten Messungen (n=100) mit einer Hochgeschwindigkeitskamera (1000 sihertz) aufgezeichnet und dann die Anzahl der Bilder zwischen den physischen Bewegungsspitzen und der entsprechenden Bildschirmreaktion bewertet.

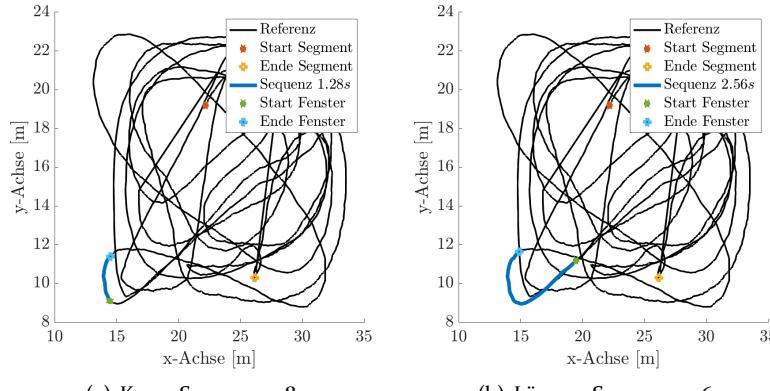


Abbildung 7.12: Einfluss der Sequenzlänge (blaue Linie) auf die Generalisierbarkeit. Kürzere Sequenzen (links) sind eher in komplexeren Trajektorien enthalten und können diese eher rekonstruieren. Längere Sequenzen (rechts) sind oft eindeutig und daher schwieriger zu verallgemeinern, d. h. die blaue Sequenz (rechts) wird nur einmal im gesamten Segment gefunden, sodass das Bewegungsmodell diese Bewegung nicht für die Rekonstruktion verwenden kann.

Abschnitte des Segments identifiziert werden, die der kurzen Sequenz sehr ähnlich sind und damit konstruiert werden können.

Konfiguration

Die Konfiguration ist verändert, die Sequenzlänge N_w wird variiert: $N_w \in [64, 128, 256, 512]$, d. h. 0.64, 1.28, 2.56 und 5.12s. Ein einschichtiges LSTM wird verwendet, um nur den Einfluss der Sequenzlänge und nicht der Tiefe (Anzahl der Schichten) auf die Speicherkapazität, die Lernstrategien und die Posengenauigkeit zu untersuchen.

Auswertung

Die Posengenauigkeit von FRFNN erhöht sich bei Sequenzlängen bis inklusive 128 Werten (Sequenzlänge=128: $MAE = 0.0375$, $MSE = 0.0141$, $RMSE = 0.0027$, $CEP_{95} = 0.0991$ in [m]), der Kontextvektor erreicht möglicherweise seine maximale Informationskapazität. Bei Sequenzlängen über 128 Werten nimmt die Posengenauigkeit ab (Sequenzlänge=256: $MAE = 0.1012$, $MSE = 0.0537$, $RMSE = 0.0646$,

$CEP_{95} = 0.1944$ in [m]). Dies könnte daran liegen, dass die Kapazität des Kontextvektors dann bereits zu begrenzt und überschritten ist. Kurze Sequenzen mit 64 Werten liefern auch eine schlechtere Positionsgenauigkeit als 128 Werte (Sequenzlänge=64: $MAE = 0.0615$, $MSE = 0.0034$, $RMSE = 0.0431$, $CEP_{95} = 0.1254$ in [m]), das könnte daran liegen, dass die Geschwindigkeitsschätzung von Hybrid bereits zu wenig Informationen in der kleinen Fensterlänge enthält, sodass v bereits deutlich ungenauer ist und sich die ungenaue Geschwindigkeitsschätzung negativ auf die Posenschätzung auswirkt. Darüber hinaus enthalten die kurzen Sequenzen viele kurze, kurvige und gerade Bewegungen, während die längeren Sequenzen komplexere Bewegungsformen enthalten.

Neue, komplexere längere Trajektorienformen können leichter aus verschiedenen kurzen Kurven und Geraden rekonstruiert werden, so dass ein Modell, das Kurven und Geraden mit kürzeren Sequenzen gelernt hat, die langen komplexen auch genauer vorhersagen kann. Im Gegensatz dazu sind komplexe Trajektorienformen in längeren Sequenzen sehr speziell und einschränkend und neue Bewegungsformen können nicht oder nur schwer daraus rekonstruiert werden (Sequenzlänge=512: $MAE = 0.2641$, $MSE = 0.1123$, $RMSE = 0.1435$, $CEP_{95} = 0.3546$ in [m]). Abbildung 7.12 visualisiert diese einfachen (links) und komplexeren (rechts) Bewegungsformen beispielhaft.

Zusammenfassung

FRFNN rekonstruiert Posen mit einem Fehler von weniger als 0.04 m bei einer optimalen Sequenzlänge von 128 Werten. Die einfacheren Bewegungsformen aufgrund kürzerer Sequenzen können komplexere Trajektorien besser rekonstruieren. Es ist daher ratsam, eine datengetriebene Architektur mit einem RNN gegen die Länge der Eingabesequenz zu optimieren. Dies ist wichtig, wenn mehrschichtige LSTM-Architekturen verwendet werden, da der Kontextvektor aufgrund der zusätzlichen vorderen oder unteren Schichten noch stärker beeinflusst wird. Nach sorgfältiger Daten- und Problemanalyse ist es ratsam, die Sequenzlänge so zu wählen, dass das Modell implizit die geeignete Lernstrategie (Abstraktionsebene der Bewegung: von Kurve und Gerade zu Kreis und Spirale) auswählt.

7.8.6 Auswirkung plötzlicher Bewegungsänderungen

Die Trajektorien der realen Welt sind normalerweise konsistent, da sich Menschen und Fahrzeuge normalerweise zielgerichtet bewegen, um ein Ziel zu erreichen, und sich die Bewegung nicht zufällig ändert. Dennoch in Sport oder in AR und VR Anwendungen ändern sich die Ausrichtung und Geschwindigkeit jedoch häufig plötzlich und abrupt. Ein optimaler Posenschätzer reagiert so schnell wie möglich mit einer möglichst kurzen Einschwingzeit und kehrt zur korrekten Trajektorie zurück. In ersten Experimenten wird deutlich, dass die zukünftige Pose für lineare Bewegungen mit konstanter Geschwindigkeit und Richtung sehr genau vorhergesagt werden kann. Plötzliche, zufällige Änderungen der Bewegung könnten jedoch eine genaue Vorhersage behindern, da die zukünftige Pose praktisch unabhängig von der vorherigen Trajektorie ist.

Ein gutes Modell reduziert den Fehler seiner Vorhersage so schnell wie möglich. Dies ist jedoch selbst für hochoptimierte KF-Methoden sehr schwierig, da das vorab definierte Bewegungsmodell des KFs starr sein muss, um starke und plötzliche Änderungen des Messrauschen auszugleichen. Abbildung 7.13(a) zeigt, wie ein KF von der korrekten Trajektorie durch eine alltägliche Bewegungsänderung abgelenkt wird und wie lange, m.a.W. wie unsicher das KF ist oder wie viele sichere Messungen nötig sind, die Methode benötigt, um wieder korrekte Posen entlang der Trajektorie zu schätzen. In diesem Experiment wird daher untersucht, wie FRFNN auf plötzliche Bewegungsänderungen reagiert.

Konfiguration

Die Konfiguration ist verändert. Der Testdatensatz der ausgelassenen Probanden wird verändert. Um das Verhalten und die Schätzgenauigkeit der Methoden während und nach Ereignissen mit abrupten und starken Bewegungsänderungen zu untersuchen, werden Fenster der ausgelassenen Probanden mit Bewegungsformen mit plötzlichen Änderungen ausgewählt. Die Fenster werden anhand starker Zunahme oder Abnahme der Beschleunigung ($> 2m/s^2$) und starken Änderungen der Orientierung ($> 75^\circ$) des Referenzsystems zwischen dem Start- und Endzeitschritt eines Fensters ausgewählt. Die ausgewählten Fenster werden von einem Studienleiter überprüft und falsch ausgewählte Fenster, die keine plötzliche Bewegungsänderung enthalten, werden verworfen. Die verbleibenden Fenster enthalten Trajektorien mit sich

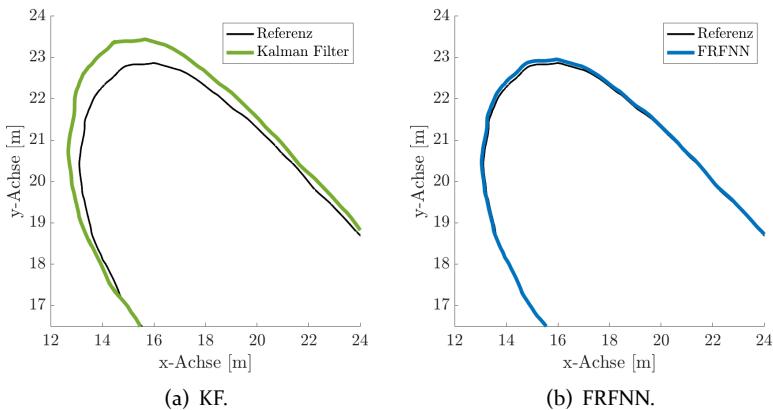


Abbildung 7.13: Posengenaugkeit einer Beispieltrajektorie des ausgelassenen Probanden A (zufällige Bewegung) mit abrupten Bewegungsänderungen. Vorhersagen des KF (links, grün) und des FRFNN (rechts, blau) beim Richtungswechsel. Zu Beginn ($x = 15$, $y = 17$ [m]) modelliert das KF die Position genau. Nach einem Richtungswechsel zeigen die Vorhersagen des KF signifikante Fehler ($x = 17$, $y = 22.5$ [m]), die sich nur langsam erholen (hier nach 3.8 s bei $x = 24$, $y = 19$ [m]). Im Gegensatz dazu ist FRFNN beim Richtungswechsel fehlerfrei.

Tabelle 7.5: Rekonstruktionsfehler (kleinste **fett**) des KF und FRFNN Verfahrens der Trajektorien in [m] der ausgelassenen Probanden.

Datensatz ausgelassene Prob.	KF				FRFNN			
	MAE	MSE	RMSE	CEP ₉₅	MAE	MSE	RMSE	CEP ₉₅
Gehen	0.1832	0.0536	0.2225	0.3813	0.0312	0.0013	0.0355	0.0588
Joggen	0.1917	0.0603	0.2339	0.4264	0.0378	0.0022	0.0454	0.0780
Rennen	0.1970	0.0569	0.2377	0.4378	0.0536	0.0042	0.0594	0.0925
Zufällig	0.2514	0.0895	0.2853	0.4595	0.0789	0.00072	0.0892	0.1367

ändernden Bewegungsabsichten und -formen. Abbildung 7.13 zeigt ein beispielhaftes Segment.

Auswertung

Die Vorhersagen der Modelle für diesen Datensatz werden ausgewertet und der MSE bei jedem Zeitschritt in jedem Fenster analysiert. Dieser ist ein Maß dafür, wie sich der Fehler nach einer plötzlichen Bewegungsänderung ändert. Neben dem MSE lässt sich das Einschwingverhalten sehr gut visualisieren und analysieren. Abbildung 7.13 zeigt das Einschwingverhalten von KF und FRFNN nach einer plötzlichen Kurvenbewegung. Es ist deutlich zu erkennen, dass das datengetriebene Verfahren deutlich schneller einschwingt und einen deutlich niedrigeren MSE ($MSE = 0.00072$ in [m]) erzielt, als das hochoptimierte KF ($MSE = 0.0895$ in [m]) selbst bei zufälligen Bewegungen.

Interessanterweise haben erste Experimente gezeigt, dass FRFNN auch kleine MSE Ergebnisse liefert ($MSE = 0.0088$ in [m]) und schneller ein schwingt, wenn es zukünftige Posen (Vorausschau=1s) vorher sagt. Tabelle 7.5 listet die Posengenauigkeiten der beiden Verfahren für die ausgelassenen Probanden auf. Um die Dauer der Einschwingphase zu evaluieren werden zehn Zufallsstichproben aus dem Datensatz der Bewegungsänderung entnommen und die Zeitschritte gezählt, die beide Methoden benötigen, um im Falle einer Abweichung zur Referenztrajektorie zurückzukehren. FRFNN reduziert Vorhersagefehler bei plötzlichen Bewegungsänderungen deutlich schneller als KF. Im Durchschnitt benötigt FRFNN 0.4s und KF 1.8s, um nach einer Abweichung zur Referenztrajektorie zurückzukehren. Abbildung 7.14 visualisiert die Posenschätzungen von KF (linke Spalte, grüne Linie) und FRFNN (rechte Spalte, blaue Linie) auf Daten des ausgelassenen Pro-

7.8 Auswertung der datengetriebenen Methode

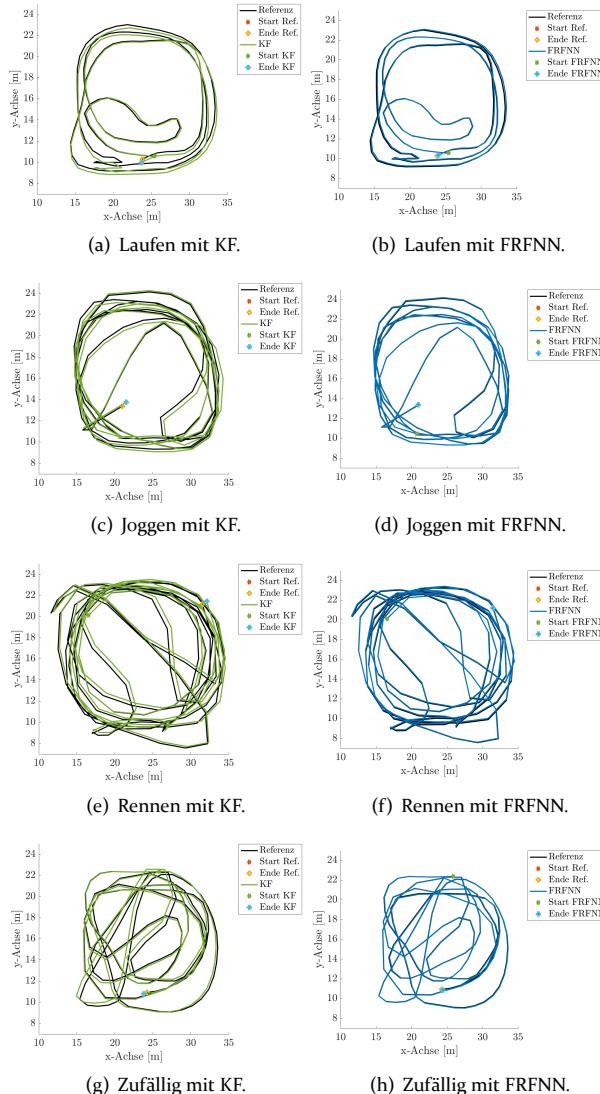


Abbildung 7.14: Draufsicht auf rekonstruierte Trajektorien; 3 min von *Gehen, Joggen, Rennen und Zufällig* des ausgelassenen Probanden A (linke Spalte: KF, rechte Spalte: FRFNN).

banden A (Aktivitäten Laufen, Joggen, Rennen und Zufällig). Während KF mit zunehmender Bewegungsgeschwindigkeit (von oben nach unten in der Spalte) und größeren Richtungsänderungen größere Abweichungen von der Referenztrajektorie aufweist, bietet FRFNN für alle Aktivitäten nahezu identische und sehr präzise Posen. KF und FRFNN werden auf V₃ optimiert. Beide könnten möglicherweise genauere Posen schätzen, wenn sie speziell für die jeweilige Aktivität optimiert sind. Die Ergebnisse unterstreichen, dass das zuvor definierte Bewegungsmodell des KF zu starr ist, um starke und plötzliche Änderungen des Messrauschens verschiedener Aktivitäten auszugleichen.

Zusammenfassung

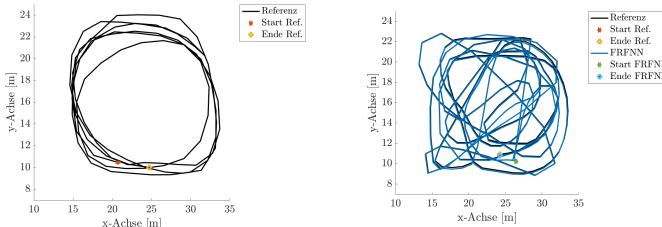
FRFNN reagiert deutlich schneller auf plötzliche Bewegungsänderungen als das hochoptimierte KF, sodass Posen deutlich weniger überschießen und Fehler auch für zukünftige Posen viel schneller minimiert werden können. Der Vorhersagefehler nach der Bewegungsänderung von FRFNN ist im Durchschnitt 60% niedriger als der von KF.

7.8.7 Auswirkung unbekannter Trajektorienformen

Dieses Experiment untersucht, ob FRFNN in der Lage ist, von bekannten Trajektorienformen auf unbekannte Trajektorienformen zu verallgemeinern. Zu diesem Zweck werden die Methoden mit kreisförmigen Trajektorienformen des Datensatzes V₃ trainiert und die Positionsschätzgenauigkeit mit zufälligen Trajektorienformen mit plötzlichen Bewegungsänderungen der ausgelassenen Probanden überprüft.

Konfiguration

Die Konfiguration wird geändert. Die Sequenzlängen werden variiert und Trainings- und Testdatensätze werden geändert. Für dieses Experiment werden explizit Trajektorienformen für den Trainingsdatensatz V₃ (Aktivitäten *Gehen*, *Joggen* und *Rennen*) ausgewählt, die Kreisbewegungen während des Trainings und zufällige Bewegungen (der ausgelassenen Probanden) für den Testdatensatz. Zu diesem Zweck werden eindeutig kreisförmige Trajektorien explizit aus Datensatz V₃ (Laufen, Joggen und Laufen) ausgewählt und Ausreißer aus dem Datensatz entfernt. Für den Testdatensatz der beiden ausgelassenen Probanden werden die Trajektorien des Zufallsdatensatzes explizit ausgewählt und diejenigen Ausreißer entfernt, die sich auf einer Kreisbahn bewegen. Abbil-



(a) Beispiel-Trajektorie der Trainingsdaten. (b) Beispiel-Trajektorie der Testdaten.

Abbildung 7.15: Draufsicht auf typische Trajektorienformen der Trainingsdaten (links, ca. 3 min) und Testdaten (rechts, ca. 3 min) zur Bewertung der Generalisierbarkeit von FRFNN mit verschiedenen Bewegungsformen; Trainingsdaten der Aktivitäten *Gehen*, *Joggen* und *Rennen* des Datensatzes V3 enthalten kreisförmige Bewegungen; Testdaten der Aktivität *Zufällig* der ausgelassenen Testperson A enthalten komplexe Bewegungsformen.

dung 7.15 zeigt eine beispielhafte Referenztrajektorie des Trainingsdatensatzes (links) und eine beispielhafte Trajektorie des Testdatensatzes des ausgelassenen Probanden A (rechts).

Auswertung

Das KF hat kaum Schwierigkeiten (außer Über- und Unterschießen), da das Bewegungsmodell nicht von Formen, sondern von den physikalischen Zuständen p_{radio} , v und acc abhängt. Somit erzielt KF ähnliche Posengenauigkeiten, als ob es explizit auf der zufälligen Aktivität optimiert wurde, da es in der Lage ist, die physikalischen Beziehungen weitestgehend aus den Daten zu bestimmen ($MAE = 0.4427$, $MSE = 0.2580$, $RMSE = 0.5001$, $CEP_{95} = 0.8331$ in [m]). Im Gegensatz dazu liefert FRFNN, wenn es nur auf kreisförmige Bewegungsformen der Aktivitäten Laufen, Joggen und Rennen trainiert wird, keine plausiblen Ergebnisse. Dies liegt daran, dass die Positionen in der Mitte der kreisförmigen Trainingsdaten unbekannt sind und daran, dass viele der kleinsten gängigen Bewegungsformen fehlen (siehe Abbildung 7.15(a)), die nötig sind, um komplexe Bewegungen zu rekonstruieren.

Daher wird FRFNN, in einem weiteren Experiment auf alle Daten der drei Aktivitäten aus Datensatz V3 trainiert, da diese auch Trajektorien enthalten, wenn Athleten die Richtung der Strecke umkehren

oder den Kreis überqueren. Dieses FRFNN-Modell schätzt dann sehr genaue Posen auf den Daten der zufälligen Aktivität der ausgelassenen Probanden, siehe Abbildung 7.15(b). Dies bedeutet, dass die kleinsten gängigen Bewegungsformen von den Trainingsdaten abgedeckt werden müssen, um sie später zur Rekonstruktion unbekannter komplexer Bewegungen verwenden zu können. Kürzere Sequenzlängen von 128-Werten führen zu den niedrigsten Fehlern ($MAE = 0.0417$, $MSE = 0.0041$, $RMSE = 0.0821$, $CEP_{95} = 0.0612$ in [m]). Längere Sequenzen von 256-Werten verschlechtern die Posengenaugkeit drastisch ($MAE = 0.3675$, $MSE = 0.2168$, $RMSE = 0.3273$, $CEP_{95} = 0.6578$ in [m]).

Die Tatsache, dass längere Sequenzen signifikant schlechtere Ergebnisse erzielen als kürzere Sequenzen, könnte darauf zurückzuführen sein, dass signifikant mehr Fraktionen der zufälligen Bewegungen mit kürzeren Sequenzen abgedeckt sind. Z. B. zeigen kurze Sequenzen einer kreisförmigen Bewegung viele kurvige und gerade Bewegungen. Längerer Sequenzen, die bspw. ganze Kreise in den Eingangssequenzen erlernen, können natürlich die zufällig wechselnden Formen nicht abdecken.

Abbildung 7.15(b) visualisiert eine rekonstruierte Trajektorie mit vielen zufälligen Bewegungen. Es ist ersichtlich, dass FRFNN nahezu kongruente, hochpräzise Trajektorien rekonstruiert, selbst wenn es mit weniger komplexen Bewegungsformen (siehe Abbildung 7.14(a-f)) trainiert wird. In Zukunft muss untersucht werden, wie feinkörnig die Fläche zur Lokalisierung in den Trainingsdaten abgedeckt sein muss, um optimale Posen zu schätzen. Die Experimente zeigten, dass eine grobe Abdeckung wie die in Abbildung 7.14 gezeigte (die Pfade, die die Kreise kreuzen) ausreichend ist, um optimale Posen zu schätzen.

Zusammenfassung

Die Wahl der Sequenzlänge hat direkten Einfluss auf die erlernten Bewegungsformen, mit denen komplexere unbekannte Bewegungsformen verallgemeinert werden können. Eine optimale Sequenzlänge enthält jede kleinste gemeinsame Bewegungsform, die jede komplexe unbekannte Bewegung darstellen kann. Ein positiver Nebeneffekt ist, dass kürzere Sequenzen gleichzeitig zu mehr Trainingsdaten führen, ohne dass zusätzliche Daten generiert werden müssen und als kostenlose Datenerweiterung des Datensatzes angesehen werden können. Da alle Ge-

schwindigkeitsschwankungen sowohl im Trainings- als auch im Testdatensatz auftreten, konzentrieren sich die Modelle auf die Abbildung der Form der Trajektorien.

7.8.8 Auswirkung von Datenlücken

In diesem Experiment wird untersucht, ob FRFNN niederfrequente, d. h. mit niedriger Aktualisierungsrate, Funkpositionen mithilfe hochfrequenter, d. h. mit hoher Aktualisierungsrate, Trägheitsmessungen interpolieren kann und wie die Posengenaugigkeit variiert. Positionsausfälle stellen Datenlücken dar, wie sie in typischen Funklokalisierungsanwendungen aufgrund von Empfangsstörungen und Mehrwegeausbreitungen (z. B. Absorption des Signals) auftreten können.

Konfiguration

Die Konfiguration ist verändert. Die Trainings- (V_3) und Testdatensätze (ausgelassene Probanden) werden verändert. Die künstlichen Datenlücken werden durch Unterabtastung des Funkpositionssignals erzeugt. Die Funkpositionen werden folgendermaßen in der Eingangssequenz verteilt: Die aktuelle Funkposition wird nach mindestens 0,5 s gesetzt, jedes Fenster enthält mindestens einen und maximal 2 Positions-werte pro Sekunde mit einem Mindestabstand von 0,5 s, beginnend bei Zeitschritt 0. Die verbleibenden Zeitschritte der Eingabesequenz des Positionsvektors werden mit dem Wert 0 gefüllt.

Auswertung

Eine anfängliche Funkposition wird immer im Zeitschritt 0 bereitgestellt, und neue Funkpositionen werden in Intervallen von 0,5 s bereitgestellt; minimale Abweichungen, $SD = 50$ ms, werden durch die variable Aktualisierungszeit der Funkposition verursacht, da bei einer beispielhaften Aktualisierungsrate von 20 Hz ungefähr alle 50 ms eine neue Position geliefert wird. Wenn mindestens 2 Funkpositionen (entspricht ca. 1 Hz Aktualisierungsrate der Funkposition) pro Eingabesequenz verfügbar sind (1,28 s), liefert FRFNN sehr ähnliche Ergebnisse für die Unterabtastung (Unterabtastung 1 Hz: $MAE = 0.0512$, $MSE = 0.0023$, $RMSE = 0.0031$, $CEP_{95} = 0.1156$ in [m]) wie für den normal abgetasteten Datensatz (normal abgetastet, 100 Hz: $MAE = 0.0375$, $MSE = 0.0008$, $RMSE = 0.0027$, $CEP_{95} = 0.0991$ in [m]). Experimente mit einer niedrigeren Aktualisierungsrate (<1 Hz Aktualisierungsrate

der Funkposition) konvergieren nicht. Experimente mit einer signifikant höheren Aktualisierungsrate (> 10 Hz Aktualisierungsrate der Funkposition) führen zu keiner höheren Positionsgenauigkeit.

Selbst wenn die Funkposition ausfällt, liefert FRFNN eine hohe Posengenauigkeit, solange es mindestens 2 aktualisierte Positionen in jeder Eingabesequenz empfängt. Das heißt, bei 128 Zeitschritten einer Sequenz mit einer Abtastrate von 100 Hz ist eine Aktualisierungsrate der Funkposition von mindestens 1 Hz erforderlich. Ein geringeres Abtasten der Funkposition mit nur einer Funkposition pro Eingabesequenz führt dazu, dass FRFNN nicht konvergiert. Dies könnte daran liegen, dass FRFNN dann nicht in der Lage ist, die Beziehungen zwischen Geschwindigkeiten und Positionen innerhalb einer Sequenz abzuleiten, da jede Position für sich als Markierung im Raum verfügbar ist. Sobald es mehr als 2 Funkpositionen pro Eingabesequenz gibt, kann das Modell die fehlenden Positionen in jedem Zeitschritt interpolieren und liefert identische Posengenauigkeiten wie FRFNN bei normaler Abtastung der Position. Darüber hinaus liefert ein Modell, das auf Daten ohne Positionsausfälle (normale Abtastung) trainiert wird, unplausible Ergebnisse bei Positionsausfällen in den Testdaten (unterabgetastet). Im Gegensatz dazu liefert das Modell, das mit moderierten Positionsfehlern trainiert wird, auch sehr genaue Posen für Testdaten mit Positionsausfällen.

Experimente mit Positionsausfälle und Verwerfungsschichten (mit 50%) vor der LSTM-Schicht führten zu ungenauen Posen. Obwohl der Effekt der Verwerfungsschicht einem Positionsausfall ähnlich ist, werden durch die Verwerfungsschicht auch Geschwindigkeiten verworfen, was es für das Modell schwieriger macht, die Fehler zu kompensieren. Es wird daher nur empfohlen, die Positionsfehler manuell zu moderieren, wenn keine Verwerfungsschicht verwendet wird. Zukünftige Forschungen müssen untersuchen, wie Verwerfungsschichten kontrolliert werden können, während Positionen händisch kontrolliert werden, ohne versehentlich alle Positionsinformationen zu verwerfen.

Zusammenfassung

Es sind mindestens 2 Funkpositionen erforderlich, um fehlende Positionen mit Geschwindigkeiten zu interpolieren. Mit einer Sequenzlänge von 1.28 s kann die Positionsrate von 100 Hz auf 1 Hz reduziert werden

und die entsprechende Kommunikationslast kann um den Faktor 100 reduziert werden. Erste Experimente zeigen, dass eine aktuelle Funkposition pro Eingabesequenz oder Fenster genaue Posen liefert, wenn nur der Änderungsbetrag, der gerichtete Abstand zwischen zwei aufeinanderfolgenden Positionen anstelle der Position in die Eingabesequenz integriert wird. Anstatt eine Position vorherzusagen, wird dann eine gerichtete Entfernung zur nächsten Position vorhergesagt. Die geschätzte Entfernung wird an eine Startposition oder an jede neu geschätzte Position angehängt. Diese Methode hat eine ähnliche Genauigkeit wie die Methode mit 2 Funkpositionen pro Fenster erzielt ($MAE = 0.0686$, $MSE = 0.0034$, $RMSE = 0.0066$, $CEP_{95} = 0.1546$ in [m]).

7.8.9 Zusammenfassung

FRFNN liefert die höchste Genauigkeit mit der Eingangsvariation p, v ($MAE < 0.038$ m). Das Hinzufügen von Geschwindigkeit zur Eingabesequenz führt zu genaueren Vorhersagen in allen Experimenten. Die Orientierung sollte nur explizit in die Eingabesequenz aufgenommen werden, wenn sich das System zur Orientierungsschätzung in einem anderen Rotationssystem als das Funksystem befindet oder die Rotationssysteme für Funkposition und Orientierung unterschiedlich sind, z. B. Funksender in der Hosentasche am Rumpf des Körpers und Trägheitssensor eines VR-Systems am Kopf. Andernfalls entstehen ungenaue Posen. Das Hinzufügen der Beschleunigung führt zu ungennauen Posenschätzungen mit FRFNN, da die Architektur von FRFNN, der vollständig verbundenen Schicht vor dem LSTM, möglicherweise nicht genügend Rauschunterdrückungskapazität (Tiefe) für die stark verrauschte Beschleunigung aufweist. Zukünftige Arbeiten sollten die Architektur (Hybrid) aus Kapitel 5 untersuchen, ob sie das Rauschen explizit aus der Beschleunigung entfernt und optimale Posen schätzt.

FRFNN sagt Posen zukünftiger (1 s) Bewegungen mit einem Fehler von weniger als 0.05 m am genauesten voraus. Auf diese Weise können system- und verarbeitungsbedingte Latenzen kompensiert werden.

Um Posen genau vorherzusagen, hängen rekurrente Verfahren von der Länge oder Menge der Eingabeinformationen ab, die sie verarbeiten müssen. Die Sequenzlänge ist daher ein Parameter, der im Rahmen einer Vorstudie immer für eine bestimmte Anwendung optimiert werden muss. Es ist wichtig zu überprüfen, welcher Abstraktionsgrad

der Information mit der Sequenzlänge ausgewählt wird, z. B. kurze Sequenzen bilden Kurven und gerade Linien ab, während lange Sequenzen Kreise und Rechtecke abbilden. Kürzere Sequenzen erfordern möglicherweise weniger Speicherkapazität des Kontextvektors in einer LSTM-Architektur. Es bleibt unklar, ob die vor- und nachgelagerten FF-Schichten die Speicherkapazität des Kontextvektors überlasten, wenn sie ungenaue Posen für längere Sequenzen schätzen. Die Sequenzlänge hat auch direkten Einfluss auf die erlernten Bewegungsformen und sollte kleinste gemeinsame Bewegungsformen enthalten, um jede komplexe, unbekannte Bewegung zu rekonstruieren.

FRFNN reagiert viel schneller auf abrupte Änderungen der Geschwindigkeit und Orientierung als hochoptimierte KF. Auf diese Weise schwingen die Posen von FRFNN viel weniger ($MSE=0.00072 [m]$) und Fehler werden schneller minimiert (0.4 s). Diese Beziehungen gelten auch für die Rekonstruktion zukünftiger Posen.

Um Datenlücken in den Funkpositionen zu schließen, sind in einer Eingabesequenz mindestens 2 Funkpositionen und eine lückenlose Geschwindigkeitssequenz erforderlich. Die leeren Zeitschritte der Sequenz müssen mit 0 Werten gefüllt sein. Alternativ können vergleichbare Ergebnisse mit dem gerichteten Abstand zwischen zwei Funkpositionen erzielt werden. Eine einzelne Entfernung wird in die Eingabesequenz eingefügt, um eine neue Entfernung zu schätzen. Dann muss die Position nacheinander zur aktuellen Entfernungsschätzung außerhalb des Netzes hinzugefügt werden.

7.9 Diskussion

In diesem Abschnitt werden die Ergebnisse der Auswertungen der modellbasierten PDR Methode (Abschnitt 7.7) und der datengetriebenen Methode (Abschnitt 7.8) verglichen und diskutiert. Es ist zu beachten, dass beide Methoden unter Verwendung der Daten der beiden ausgelassenen Probanden diskutiert werden.

Vor- und Nachteile der Methoden

Aus Sicht der Anwendung hat das modellbasierte PDR Verfahren den Vorteil, dass jede Komponente ausgetauscht werden kann. Auf diese Weise können spezifische Änderungen berücksichtigt und

einzelne Komponenten der Verarbeitungskette ausgetauscht werden. Im Gegensatz dazu erfordert FRFNN eine zeitaufwendige Umschaltung. Die Varianten FRFNN ($MAE=0.0789\text{ m}$) und KF sind präziser ($MAE=0.2514\text{ m}$) als die modellbasierte PDR Variante ($MAE=0.4900\text{ m}$), da sie die verschiedenen Messungen implizit optimal kombinieren.

Während der Kalibrierung der modellbasierten PDR-Methode hilft die Neukalibrierung am meisten, wenn die Geschwindigkeit langsam ist. Das optimierte KF hat Schwierigkeiten mit schnellen Änderungen von Bewegung und Richtung. Die datengetriebene Methode liefert für alle Aktivitäten die gleichen Ergebnisse und bleibt daher von Geschwindigkeits- und Richtungsschwankungen unberührt. Es ist jedoch zu beachten, dass sich die Testdaten auch für das datengetriebene Verfahren innerhalb einer Verteilung oder zwischen zwei Verteilungen befinden müssen, die von den Trainingsdaten überspannt werden. Auch hier schlägt eine Extrapolation über die gelernte Verteilung hinaus fehl: Positionen, die über das gelernte Koordinatensystem hinausgehen, können nicht zu einer neuen Position verarbeitet werden.

Einfluss der Funkposition

Insgesamt liefert die datengetriebene Methode in allen Tests höhere Posengenauigkeiten als die modellbasierte PDR und die KF-Methode. Dies legt nahe, dass FRFNN eine optimale Fusion für Funk- und Trägheitssensoren finden kann. Die modellbasierte PDR-Methode erfordert jedoch deutlich weniger häufige Positionsaktualisierungen (alle 60 s) als die KF und die datengetriebene Methode (alle 0.5 s).

Die Positionssprünge, das Wackeln und die Jitter sind bei den modellbasierten PDR und KF-Methoden mit vielen Richtungsänderungen stärker ausgeprägt als bei der datengetriebenen Methode. Die datengetriebene Variante zeigt keine Sprünge in den Trajektorien. Dies ist hauptsächlich auf die implizite optimale Fusion aller Informationen zurückzuführen, insbesondere wenn die angegebenen Positions- und Geschwindigkeitsschätzungen bereits sehr genau sind.

Weitere Experimente haben gezeigt, dass bspw. die Positionsschätzung präzise sein muss, um ungenaue Geschwindigkeitsschätzungen mit der datengetriebenen Methode implizit zu korrigieren. Wenn beide falsch schätzen, liefert die datengetriebene Methode auch weniger

genaue Posen. Das Einschwingverhalten der datengetriebenen Methode ist deutlich schneller als die Korrektur der modellbasierten PDR-Methode (alle 30 s) und verhindert so Positionssprünge.

Auswirkung der Geschwindigkeit

Je höher die Geschwindigkeit ist, desto mehr Fehler häufen sich in der rekonstruierten Trajektorie der modellbasierten PDR und der KF-Methode. Das FRFNN-Verfahren liefert jedoch genaue Positionen, solange die Positionsschätzungen des Funksystems genau sind. Mit zunehmender Geschwindigkeit weichen die rekonstruierten Trajektorien der modellbasierten PDR Methode stärker von der Referenz ab. Die datengetriebene Methode wird nicht beeinflusst.

Auswirkung von Sensorrauschen

Erste Studien haben gezeigt, dass das Rauschen der Positions- und Geschwindigkeitsschätzungen die Posengenaugkeit von FRFNN nicht beeinflusst. Das datengetriebene Modell liefert genaue Posen, wenn die Daten mit einem gleitenden Mittelwertfilter oder Bandpassfilter geglättet wurden, bevor sie in das datengetriebene Modell eingeführt werden oder wenn stattdessen das Rohsignal verwendet wird. Dies zeigt, dass FRFNN das Rauschen der Sensoren und die Fehlervarianz der Positions- und Geschwindigkeitsschätzungen sehr gut verarbeiten, Rauschen aus den Datenströmen entfernen und sich trotzdem auf die wesentlichen Merkmale und Informationen konzentrieren kann.

Auswirkung von Datenausfall

Da die Positions-, Geschwindigkeits- und Orientierungsschätzer unterschiedliche Fensterüberlappungen, Aktualisierungsraten und Übertragungsverzögerungen aufweisen können, können ihre Datenraten und zeitlichen Beziehungen variieren. Die Eingabedatenströme dieser Schätzer werden daher im Trainingsprozess linear abgetastet, um eine einheitliche Datenrate und Sequenzlänge zu erhalten. Dies ist zur Laufzeit nicht möglich, da zukünftige Punkte für die Interpolation fehlen. Die Vorhersagbarkeit der datengetriebenen Methode löst dieses Problem, wenn die Daten Verzögerungen von weniger als 1 Sekunde aufweisen, da sonst die Genauigkeit der Methode erheblich abnimmt. Das modellbasierte PDR-Verfahren leidet andererseits nicht unter Systemverzögerungen, arbeitet mit den aktuellen Daten und liefert Posen auf

Daten, wenn diese verfügbar sind. Das Verfahren kann jedoch keine zukünftigen Posen abschätzen oder Systemverzögerungen kompensieren, d. h. es arbeitet in der Vergangenheit. Im Gegensatz dazu, bricht das KF direkt zusammen, da es nur den aktuellen und letzten Zustand zur Vorhersage nutzen kann.

Auswirkung der Sequenzlänge

Im Gegensatz zu datengetriebenen Methoden kann die modellbasierte PDR-Methode Funkpositionsfehler von über 60 s überbrücken. Dies ist mit dem vorgeschlagenen FRFNN-Modell nicht möglich, da dies eine Sequenzlänge von 6,000 Werten ($60 \text{ s} \cdot 100 \text{ Hz}$) erfordern würde, um mindestens 2 Funkpositionen in einer Sequenz zu enthalten. Diese Annahme widerspricht jedoch der maximal möglichen Kapazität (zur Erinnerung: eine Sequenzlänge von mehr als 1.28 s hat die Speicherkapazität des Kontextvektors des Einzelschichtmodells überlastet). Darüber hinaus widerspricht diese Annahme der Lernstrategie der kleinsten gängigen Bewegungsformen (128 Werte führen zu ausreichenden atomaren Bewegungsformen, z. B. geraden Linien, um komplexere zufällige Bewegungsmuster wie Spiralen zu rekonstruieren). Zukünftige Arbeiten müssen hier einen Kompromiss untersuchen.

Die aktuelle Architektur von FRFNN erlaubt keine ultralangen Sequenzen (mehr als 128 Werte für einschichtige LSTM-Architekturen), da diese möglicherweise bereits die Speicherkapazität des internen Kontextvektors voll ausnutzen. Der letzte Zeitschritt der Architektur liefert die endgültige Position, die durch den letzten Kontextvektor bestimmt wird, der sich an alle vorherigen Zustände erinnern muss. Eine längere Sequenz kann dazu führen, dass die wichtigen Informationen zu Beginn der Sequenz verblassen.

Skalierung der Daten

Durch die Neukalibrierungen werden die geschätzten Trajektorien der modellbasierten PDR Methode an die Referenztrajektorie angeähert. Das FRFNN-Verfahren kalibriert sich andererseits implizit neu, solange es zwei aktuelle Funkpositionen in einer Eingabesequenz empfängt. Die Daten müssen jedoch a priori skaliert werden, und die Skalierung muss auch für die unbekannten Daten einer Live-Phase gelten. Darüber hinaus müssen die Eingabesequenzen so skaliert werden, dass

die Werte für Geschwindigkeit, Beschleunigung und Position im gleichen Wertebereich liegen. Es muss auch beachtet werden, dass Positionen und Beschleunigungen ähnliche Wertebereiche haben, um realistische Werte darzustellen. Wenn die Werte der Position (z. B. $x = 100.000.000$ m und $y = 900.000.000$ m) signifikant größer sind als die der Geschwindigkeit (0.0001 m/s), ist der Einfluss der Position auf den Lernerfolg signifikant höher als der der Geschwindigkeit. Durch eine einfache Skalierung der maximal möglichen Werte, bspw. anhand der Sensormessbereiche, der maximal möglichen Bewegungsgeschwindigkeit und der maximalen Entfernung der Funkverbindung, können die Wertebereiche optimal zueinander skaliert werden. Dies war für die hier verwendeten Daten nicht erforderlich.

Optimales Training

Die Variation der Stapelverarbeitung hat in allen Experimenten einen positiven Einfluss auf die Posengenaugkeit. Keine Variation der Sequenzen in der Stapelverarbeitung führte zu unkontrollierten Trainingsverläufen (Abbruch aufgrund eines vorzeitigen Stopps, wenn sich Trainings- und Validierungsfehler 5 mal überschneiden) und liefert vergleichbare oder etwas weniger genaue Posenschätzungen.

Die Experimente in Abschnitt 7.8 zeigen, dass es sinnvoll ist, künstliche Datensätze zu generieren, um alle realistischen Geschwindigkeiten und Bewegungsformen mit einer Sequenzlänge abzudecken, die lang genug ist, um Formen und Geschwindigkeiten aufzunehmen, aber kurz genug, um alle Bewegungsformen rekonstruieren zu können. Bspw. kann ein Kreis (Kreisverkehr) aus Kurven und ein Rechteck (Ampelkreuzung) aus geraden Linien rekonstruiert werden.

Generalisierbarkeit der zufälligen Aktivität

Unerwarteterweise behandeln alle Methoden modellbasiert PDR ($CEP_{95} = 1.2500$ m), KF ($CEP_{95} = 0.4595$ m) und datengetrieben ($CEP_{95} = 0.1367$ m) das kompliziertere Muster der Aktivität *Zufällig* sehr gut. Die datengetriebene Methode liefert jedoch deutlich genauere Ergebnisse erfordert jedoch viel häufiger eine aktuelle Funkposition.

7.10 Zusammenfassung

Die modernsten Ansätze auf dem neuesten Stand der Technik können Funk- und Trägheitssensormessungen verwenden, um die Position menschlicher Bewegung in einigen alltäglichen Situationen zu schätzen. Sie haben jedoch alle Schätzfehler, wenn die Bewegung dynamisch ist, z. B. in Sport- und VR-Anwendungen, wenn abrupte und unvorhersehbare Bewegungsänderungen auftreten oder wenn Funk- oder Trägheitssensoren nicht starr am Körper angebracht sind. Eine große Schwierigkeit besteht darin, diese multimodalen Sensoren zu einer optimalen Pose zu kombinieren. Das Bewegungsmodell des KFs ist typischerweise sehr allgemein, um so viele Bewegungsformen wie möglich abzudecken, während das Mess- und Prozessrauschen sehr anwendungsspezifisch definiert wird, um Sensorwerte und -zustände so genau wie möglich zu verfolgen. Daher sind diese Filter für Sportanwendungen mit hoher Bewegungsdynamik eher ungeeignet. PF leiden zur Laufzeit unter einem sehr hohen Rechenaufwand, haben ernsthafte Schwierigkeiten bei der Korrektur von Messfehlern und verfügen nicht über ein optimiertes Bewegungsmodell. Daher sind sie auch für Echtzeitanwendungen ungeeignet. Beide Filter können nur mit großem Aufwand parametriert werden und sind dann auf die jeweilige Anwendung beschränkt. Wie multimodale Sensordaten optimal zusammengeführt werden können, ist noch nicht geklärt.

Daher wurde in dieser Dissertation eine neue datengetriebene Methode zum Fusionieren verschiedener Sensordatenströme vorgestellt. Die Architektur von FRFNN kombiniert vorwärtsgekoppelte und rekurrente Schichten, um die aktuelle Pose einer Person aus Signalströmen von Funkpositionen, rotationsinvarianten Geschwindigkeiten und Orientierungen zu schätzen. Die Kombination von vor- und nachgeschalteten Vorwärtsschichten und LSTM-Zellen extrahiert Merkmale aus den Sensorsignalen, verfolgt ihre zeitlichen Beziehungen und übersetzt diese in eine andere Domäne, um Posen oder Trajektorien zu rekonstruieren. Ein hochgenaues Referenzsystem wird verwendet, um Funkpositionen, Geschwindigkeiten und Orientierungen von Athleten aufzuzeichnen und aus den Daten auf überwachte Weise zu lernen, wie die Pose und Trajektorien einer gehenden Person, eines Joggers oder eines Sprinters genau geschätzt werden kann. Die aufgezeichneten Funk- und Trägheitsmessungen werden mit den Methoden aus den vorherigen Kapiteln zu Funkpositionen, Geschwindigkeiten und

Orientierungen verfeinert. Diese können dann implizit mit FRFNN zusammengeführt werden, um eine aktuelle Pose zu bestimmen. Das Kapitel vergleicht auch quantitativ Verfahren des Stands der Technik, nämlich ein modellbasiertes PDR und ein KF-Verfahren mit FRFNN.

Die Ergebnisse der Experimente sind eine Referenz für konzeptuelle Entscheidungen für Regressionsproblemen anderer Domänen und stimmen mit den Ergebnissen früherer Untersuchungen überein [15, 127]. Die Ergebnisse der Architekturoptimierung zeigen, dass die FRFNN-Architektur mit einfachen Vorwärtsschichten vor und nach einer LSTM-Schicht andere Architekturen übertrifft, da sie die gleiche Posengenauigkeit wie gestapelte und dichte LSTM-Schichten erzielt, wesentlich weniger rechenintensiv ist als gestapelte LSTMs und erheblich zuverlässigere Trainingsprozesse, wie dichte LSTMs zeigt. Modelle mit mehr Parametern erzeugen weniger Vorhersagefehler als kleinere, und 100.000 Parameter bieten optimale Posengenauigkeiten. Weitere Schichten verbessern den Vorhersagefehler nicht. Für genaue Trajektorienvorhersagen sind mindestens 14 verborgene Zustände der Verdichtungs- und der LSTM-Schichten erforderlich, 128 Zustände pro Schicht liefern die höchste Positionsgenauigkeit. Modelle mit mehr als 128 verborgenen Zuständen schneiden nur unwesentlich besser ab. Durch Hinzufügen von Geschwindigkeit als Eingabe wird der Vorhersagefehler um bis zu 90% reduziert.

Im Vergleich zu FRFNN bieten die modellbasierte PDR und die KF Methode geringere Genauigkeiten und weniger Vorteile, selbst wenn sie stark optimiert sind. Während die Rechenkosten für beide niedriger sind, haben sie Schwierigkeiten, auf unbekannte Trajektorien zu verallgemeinern, können zukünftige Posen nicht vorhersagen und können daher die Mess- und Systemverzögerungen trotz der geringen Rechenkosten nicht kompensieren. Darüber hinaus leidet das KF unter der Tatsache, dass es bei starken und abrupten Änderungen in Bewegung und Richtung hohe Fehler erzeugt und lange braucht, um diese zu kompensieren. Die modellbasierte PDR-Methode akkumuliert noch größere Fehler und kompensiert sie durch eine Kalibrierung der Position die zu Positionssprüngen führen. Dank der impliziten Fusion im FRFNN-Modell ist das Einschwingverhalten viel präziser und damit schneller als bei den Verfahren des Stands der Technik. Im Durchschnitt benötigt FRFNN 0.8s und KF 7.8s, um nach einer

Abweichung zur Referenztrajektorie zurückzukehren.

Gegenüber der modellbasierten PDR und der KF-Methode bietet FRFNN weitere Vorteile: Die Anhäufung von Fehlern in den Geschwindigkeitsschätzungen von Hybrid können mit FRFNN beseitigt werden und glatte Trajektorien ohne Ausreißer können rekonstruiert werden. FRFNN verallgemeinert aus verschiedenen Bewegungsformen, wenn die Sequenzlänge Bruchteile der komplexen Trajektorienformen abdeckt und diese rekonstruieren kann. FRFNN kann gut auf verschiedene Bewegungsarten verallgemeinert werden, wenn die Trajektorienformen und Geschwindigkeiten des Trainingsdatensatzes den Bereich des Testsatzes abdecken. Darüber hinaus kann FRFNN zukünftige Posen vorhersagen und so Systemverzögerungen und den eigenen Rechenaufwand kompensieren. Der Fehler nimmt jedoch mit zunehmendem Prognosehorizont zu. FRFNN ist auf realistischere Probleme als frühere Ansätze anwendbar, da die Posengenauigkeit hoch ist, selbst wenn die eingegebenen Funk- und Trägheitsmessungen von unbekannten Sensoren und Benutzern stammen und wenn sich die Trainings- und Testdaten nicht in derselben Domäne befinden. Darüber hinaus kann FRFNN herausfordernde Bewegungen und unterschiedliche Gehgewohnheiten unbekannter Benutzer effizient bewältigen und ist robust gegenüber Ausreißern, hauptsächlich aufgrund des gemeinsamen Trainings aller Bewegungen und Geschwindigkeiten in einem Modell. So verarbeitet die mit LSTM-Zellen verwendete FRFNN-Architektur Eingabesequenzen, um von bekannten elliptischen Trajektorien auf unbekannte zufällige Trajektorien zu verallgemeinern. Gleichzeitig werden zukünftige Bewegungsdaten erfolgreich vorhergesagt, Trägheitssensordaten werden implizit von Sensorrauschen gereinigt und unterabgetastete Funksignale werden interpoliert.

8 Zusammenfassung, Diskussion, Anwendungen und Ausblick

In diesem Kapitel werden einige wesentliche Ergebnisse der Dissertation zusammengefasst, Forschungsfragen diskutiert und beantwortet, Anwendungen der untersuchten Methoden demonstriert und ein Ausblick auf das zukünftige Forschungspotential gegeben. Das Kapitel ist folgendermaßen strukturiert: Abschnitt 8.1 fasst die Arbeit zusammen. Abschnitt 8.2 diskutiert Limitierungen. Abschnitt 8.3 geht auf die Beiträge der Arbeit ein und beantwortet die Forschungsfragen. Abschnitt 8.4 beschreibt Anwendungen, die im Rahmen der Dissertation durch die vorgeschlagenen Methoden ermöglicht und demonstriert wurden. Abschnitt 8.5 berichtet über offene Fragestellungen und mögliche zukünftige Arbeiten. Abschließend werden in Abschnitt 8.6 letzte Worte zur Dissertation an den Leser gerichtet.

8.1 Zusammenfassung

Dieser Abschnitt fasst die Erkenntnisse der Methoden zur Positionsschätzung (Abschnitt 8.1.1), Geschwindigkeitsschätzung (Abschnitt 8.1.2), Orientierungsschätzung (Abschnitt 8.1.3) und Fusion (Abschnitt 8.1.4) zusammen.

In dieser Arbeit wurden die Auswirkungen von datengetriebenen Verfahren entlang einer typischen Verarbeitungskette zur Lokalisierung mit Funk- und Trägheitssensorik untersucht. Die Verarbeitungskette ist lose gekoppelt, sodass jedes datengetriebene Verfahren problemlos ausgetauscht werden kann. Zunächst werden Positionen aus Ankunftszeiten eines Funksystems in Innenräumen geschätzt. Diese werden an eine parallele Orientierungsschätzung und eine nachgelagerte Posenschätzung weitergeleitet. Die Orientierungsschätzung klassifiziert zuverlässig Bewegungsmuster, um eine langzeitstabile Orientierung auf Basis des Richtungsvektors zweier Funkpositionen für die Posenschätzung bereitzustellen. Parallel werden aus rotationsinvarianten, verrauschten Trägheitsmessungen genaue Geschwindigkeiten geschätzt, die ebenfalls an die Posenschätzung weitergeleitet werden. Die Posenschätzung fusioniert Positionen, Geschwindigkeiten und Orientierungen, um eine genaue zukünftige Trajektorie zu rekonstrui-

ren. Diese neuartige datengetriebene Verarbeitungskette liefert genauere und robustere Schätzungen als bekannte Verfahren des Stands der Technik, auch in dynamischen Bewegungen, mit verrauschten Trägheitssensoren und Funkumgebungen mit Mehrwegeausbreitung.

8.1.1 Datengetriebene Positionsschätzung

Kapitel 4 stellt eine neue Verarbeitungskette vor, die verrauschte ToA-Werte geeignet bündelt und ein überwachtes Regressionsverfahren formuliert, das die ToA-Bündel mit einem datengetriebenen Verfahren auf hochgenaue Referenzpositionen abbildet. Die vorgestellte gestapelte LSTM-Architektur kann aufeinanderfolgende Positionen aus ToF-Messungen, die von realen Mehrwegepfaden gestört sind, deutlich genauer vorhersagen ($MAE < 0.024 \text{ m}$) als Methoden des Stands der Technik ($MAE < 1.024 \text{ m}$). Kapitel 4 zeigt auch, dass das LSTM-Modell lernt, mit zeitabhängigen Fehlern wie realer Mehrwegeausbreitung und synthetischer langfristiger zufällige (stochastische) Irrfahrt umzugehen ($MAE < 0.036 \text{ m}$), das bislang für Methoden des Stands der Technik unmöglich war (keine plausiblen Ergebnisse möglich). Weitere Experimente haben gezeigt, dass die Tiefe (Anzahl der Schichten) und die Sequenzlänge wichtige Parameter sind, die direkt aus den anwendungsspezifischen Daten konfiguriert werden können. Das vorgestellte Verfahren übertrifft sowohl bei synthetisch erzeugten als auch bei realen Trajektorien, die driftende ToF-Messfehler enthalten, ein optimiertes KF aus dem Stand der Technik. Die resultierenden Positionsschätzungen werden dann zur Orientierungsschätzung und zur Fusion weitergeleitet, um Posen und Trajektorien zu rekonstruieren.

8.1.2 Datengetriebene Geschwindigkeitsschätzung

Alle bekannten Verfahren des Stands der Technik Schätzfehler weisen auf, wenn die Bewegung des Sensorträgers dynamisch ist, z. B. bei VR- oder Sportanwendungen, wenn abrupte und unvorhersehbare Bewegungsänderungen auftreten oder wenn IMU-Sensoren nicht starr am Körper angebracht sind. Kapitel 5 stellt daher eine neue Verarbeitungskette vor, die aus verrauschten Messungen von Trägheitssensoren Geschwindigkeiten schätzt. Dazu werden zwei überwachte Regressionsverfahren formuliert, die Bündel von Signalgrößenvektoren von Beschleunigungen und Drehraten auf Referenzgeschwindigkeiten abbilden. Eine Variante, C/RNN, zielt darauf ab, möglichst

schnell und ressourcenschonend (über 160 Hz) genaue Geschwindigkeiten ($MAE < 0.30 \text{ m/s}$ bei unbekannten Personen) zu schätzen. Die andere Variante, Hybrid, zielt darauf ab, möglichst robuste und hochgenaue Geschwindigkeiten zu schätzen ($MAE < 0.19 \text{ m/s}$ bei unbekannten Personen). Die neuartigen Architekturen kombinieren tiefes Lernen mit einem Bayesschen Filter, um die Geschwindigkeit einer Person in rotationsinvarianten Signalströmen zu schätzen. Ein hochgenaues Referenzsystem wird verwendet, um Geschwindigkeitsdaten von Athleten zu sammeln und aus den Daten auf überwachte Weise zu lernen, wie die Geschwindigkeit einer gehenden Person, eines Joggers oder eines Sprinters genau geschätzt werden kann. Ein Vergleich der Methoden mit aktuellen Verfahren aus dem Stand der Technik hat in zahlreichen Experimenten belegt, dass Hybrid deutlich robuster gegen sensorspezifisches Rauschen ist und die genauesten Geschwindigkeiten (0.16 m/s auf Testdaten) auch über lange Zeiträume schätzt. Untersuchungen haben gezeigt, dass Hybrid auch realistische Probleme sehr genau löst, selbst wenn die eingegebenen IMU-Messungen von unbekannten Sensoren oder unbekannten Benutzern stammen und wenn sich die Trainings- und Testdaten nicht in derselben Domäne befinden. Die resultierenden Geschwindigkeitsschätzungen werden dann zur Fusion zur Posenabschätzung und zur Trajektorienrekonstruktion genutzt.

8.1.3 Datengetriebene Orientierungsschätzung

Kapitel 6 stellt eine neue Verarbeitungskette vor, um aus verrauschten Beschleunigungs- und Drehratenmessungen eines Trägheitssensors und aus Positionen, des Positionsschätzers aus Kapitel 4, eine langzeitstabile Orientierung zu schätzen. Dazu wird ein überwachtes Klassifikationsverfahren formuliert, das statistische Merkmale nutzt, um spezifische Bewegungsmomente zu detektieren, die die Pose von Kopf zu Körper beschreiben. Es wird gezeigt, wie langfristig stabile absolute Orientierungen (auch mit Sensoren aus unterschiedlichen Koordinatensystemen, bspw. Kopf-Körper-Orientierungen) aus ungenauen Positionen und verrauschten Trägheitssensoren (bspw. am Kopf) abgeschätzt werden. Es wurde eine Reihe von Merkmalen vorgestellt, die aus gefilterten Sensordaten extrahiert werden können, um genau die Momente zu erfassen, in denen Benutzer mit dem Kopf nach vorne gerichtet gehen und die akkumulierte Drift aus dem Richtungsvektor zweier konsekutiver Positionen ableiten kann. Experimente in typischen Mehrbenutzer- und großflächigen VR-Szenarien, nämlich einem Mu-

seum, haben gezeigt, dass diese Technik die Kopf-zu-Körper-Haltung eines Benutzers mehrmals pro Minute zuverlässig (F1-Maß, engl. F1-Score=95% mit einem maximalen Fehler von $\pm 15^\circ$ in 5% aller Fälle) bestimmt, wenn der Benutzer in die Bewegungsrichtung blickt, auch bei natürlicher und entspannter Bewegung (einschließlich des Kopfes). Die resultierenden Orientierungsschätzungen werden dann verwendet, um die Orientierung der Pose und der rekonstruierten Trajektorie in der Fusion zu kalibrieren und zu stabilisieren, um Sport- und VR-Anwendungen zu ermöglichen.

8.1.4 Datengetriebene Fusion

Das Kapitel 7 stellt eine neue Verarbeitungskette vor, die die Positions-, Geschwindigkeits- und Orientierungsschätzungen der vorgegangenen Kapitel 4-6 aufnimmt, um auf deren Basis Posen und Trajektorien von Sportlern zu rekonstruieren. Die modernsten Ansätze auf dem neuesten Stand der Technik haben Posenschätzfehler, wenn abrupte und unvorhersehbare Bewegungsänderungen auftreten, Trägheitssensoren nicht starr am Körper angebracht sind oder Funkumgebungen viele störenden Mehrwegepfade enthalten. Neben dem Entfernen dieser Störquellen und (nicht)deterministischem Rauschen, besteht die größte Schwierigkeit darin, diese multimodalen Sensoren zu einer optimalen Pose zu kombinieren. Herkömmliche Bayes-Filter können nur mit großem Aufwand parametriert werden und sind dann auf die jeweilige Anwendung beschränkt. Deshalb wird ein überwachtes Regressionsverfahren, FRFNN, formuliert, das Bündel von Positionen, Geschwindigkeiten und (optional) Orientierungen auf Referenzposen und -trajektorien abbildet. Quantitative Vergleiche der Verfahren des Stands der Technik wie eine modellbasierte PDR und ein optimiertes KF mit FRFNN zeigen, dass die FRFNN Architektur mit einfachen Vorwärtskopplungsschichten vor und nach einer rekurrenten LSTM-Schicht den Stand der Technik übertrifft ($MAE=0.037\text{ m}$). Die höheren Rechenkosten von FRFNN können durch seine Vorhersagefähigkeit kompensiert werden. Zahlreiche Experimente haben gezeigt, dass FRFNN genauer und schneller auf plötzliche Bewegungsänderungen reagiert ($MSE=0.00024\text{ m}$) und in der Lage ist, die Eingabesequenzen zu verarbeiten, um von bekannten elliptischen Trajektorien auf unbekannte zufällige Trajektorien zu verallgemeinern ($MAE=0.0417\text{ m}$). Gleichzeitig werden zukünftige Bewegungsdaten erfolgreich vorhergesagt (Prognose für 1 s in der Zukunft ergibt einen $MAE=0.0489\text{ m}$), Träg-

heitssensordaten intern mit Sensorrauschen geglättet und unterabgetastete Funksignale interpoliert ($MAE=0.0512\text{ m}$).

8.2 Diskussion

Dieser Abschnitt diskutiert die Limitierungen und das Zusammenspiel der Verfahren der Kapitel 4 bis 7. Abschnitt 8.2.1 diskutiert die Abhängigkeit der Methoden von der Sequenzlänge der Eingabedaten. Abschnitt 8.2.2 diskutiert Limitierungen der Datensätze, die zur Auswertung der Methoden verwendet wurden. Abschnitt 8.2.3 bis 8.2.6 diskutieren die Methoden zur Positionsschätzung (Abschnitt 8.2.3), der Geschwindigkeitsschätzung (Abschnitt 8.2.4), der Orientierungsschätzung (Abschnitt 8.2.5) und der Fusion (Abschnitt 8.2.6).

8.2.1 Abhängigkeit von der Sequenzlänge

Der in dieser Dissertation vorgeschlagene Ansatz basiert auf der Verfügbarkeit der Sensordaten unterschiedlicher Sensorfamilien, die konsistent Datenströme verschiedener Bewegungsinformationen liefern müssen. Bspw. erfordert die Orientierungs- und Geschwindigkeitsschätzung Bruchteile oder einen vollständigen Gangzyklus in einem Schiebefenster mit fester Größe sowie einen absoluten Positionsvektor (zurückgelegte Strecke pro Fenster). In der Praxis kann dies zu Einschränkungen führen. Es ist möglich, dass nicht jedes Schiebefenster genügend Informationen enthält, um geeignete Momente zur Kalibrierung zu finden, oder dass Beschleunigungswerte fehlen, die zu einer falschen Geschwindigkeitsschätzung führen. Obwohl in Experimenten im Abschnitt 6.8.5 gezeigt wird, dass variable Bewegungsgeschwindigkeiten die Genauigkeit der Klassifizierung nicht beeinflussen, gibt es möglicherweise Bewegungen, die die Genauigkeit der Klassifizierung oder Regression beeinflussen. Ein vorgelagerter Prozess könnte die Bewegungsgeschwindigkeit klassifizieren, um eine geeignete Sequenzlänge zu wählen, die die verschiedenen Bewegungsformen in jeder Sequenz abhängig von der Geschwindigkeit optimal abdeckt. Obwohl datengetriebene Methoden versuchen, diese Probleme implizit zu lösen, sind erheblich mehr Daten erforderlich, um einen perfekten Datensatz zu erstellen, der diese Korrelation enthält. Es ist daher ratsam, die Sequenzlänge mithilfe einer Rastersuche zu optimieren.

8.2.2 Datensatz Limitierungen

Aus Sicht des maschinellen und des tiefen Lernens würde ein *perfekter* Datensatz identisch verteilte Daten (exakt äquivalente statistische Eigenschaften jeder einzelnen individuellen Kategorie) einbetten, sodass jeder Proband in der gleichen möglichen Variation von Beschleunigungen und Geschwindigkeiten läuft, joggt und rennt mit gleicher Dauer auf der gleichen Trajektorie und mit jeder möglichen Gangart und Geschwindigkeit auf jeder möglichen Trajektorie. Ein solcher Datensatz würde auch alle möglichen Gang- und Bewegungsverhalten abdecken. Dieser Datensatz würde erfordern, dass Probanden dieselben Bewegungsmuster bereitstellen, die über die gesamte Zeitspanne einer Aktivität zu denselben Geschwindigkeiten führen. Die Erzeugung eines solchen Datensatzes war unmöglich. Aufgrund der unterschiedlichen Körperphysik lieferten die Probanden keine deterministischen Geschwindigkeiten, z. B. joggten einige mit einer Geschwindigkeit, die andere nur erreichten, wenn sie rannten. Daher mischen sich sowohl die Bewegungsmuster als auch die Geschwindigkeiten innerhalb und zwischen den Aktivitäten. Obwohl dies durch die Bildung von Gruppen mit nahezu gleichmäßig verteilten Geschwindigkeiten näherungsweise kompensiert wurde, gibt es noch Raum für Verbesserungen.

8.2.3 Funklokalisierung

Ein Grundproblem der funkbasierten Lokalisierung ist die korrekte Auflösung von Mehrwegekomponenten in einem Funksignal, um das Kanalmodell einer Sender-Empfänger-Verbindung genau zu beschreiben, ToA-Werte korrekt zu identifizieren, um eine Position genau zu multilaterieren. Einerseits ist die Bandbreite der Funksignale oft zu schmal, um die räumliche Verteilung der Mehrwegekomponenten hoch aufzulösen, um genauere Ankunftszeiten des direkten ersten Signalpfades zu identifizieren. Andererseits sind die Mehrwegekomponenten sehr umgebungsspezifisch. Antennengeometrie, Objekte in der Umgebung und die Dynamik der zu lokalisierenden Objekte führen zu völlig neuen Funkausbreitungsumgebungen. Diese Umweltveränderungen spiegeln sich auch in den ToA-Werten wieder, die in Kapitel 4 verwendet werden, um sie mit einem datengetriebenen Modell auf Referenzpositionen abzubilden. Das Modell arbeitet in statischen Umgebungen mit LoS, OLoS und NLoS Komponenten und mit dynamischen Sendern und funktioniert, solange sich die Funkaus-

breitungsumgebung nicht dynamisch ändert, z. B. solange keine neuen Störquellen hinzugefügt oder verschoben werden. Die Ergebnisse der Experimente in Kapitel 7 deuten darauf hin, dass das FRFNN-Modell die Geschwindigkeitsschätzungen verwenden kann, um Störquellen der Funkausbreitung innerhalb einer Eingabesequenz zu identifizieren, da es die korrekten Beziehungen zwischen Geschwindigkeit und Position kennt und durch Mehrwege gestörte Positionsänderungen in einer Sequenz kompensieren kann, um genaue Posen zu schätzen.

Die Korrelationen zwischen Funk- und Trägheitssensor müssen jedoch genauer analysiert werden. Erste Experimente haben gezeigt, dass nur wenige Datenpunkte aus einer neuen Funkumgebung an möglichst weit voneinander entfernten Positionen, die den größtmöglichen Bereich abdecken, ausreichen, um ein vorab trainiertes Modell an eine neue Umgebung anzupassen. Ein anderes Experiment hat gezeigt, dass das Vortraining einer einfachen CNN-Architektur auf synthetischen, deterministischen ToA-Werten mit LoS-Mehrwegkomponenten und die anschließende Anpassung des Modells durch Umschulung des Modells auf einige wenige (im Verhältnis 1:10, real zu synthetisch) echte ToA-Werte mit LoS-, OLoS- und NLoS-Komponenten den Positionsfehler, in einer realen Umgebung von $CEP_{95} = 1.35 \text{ m}$ auf unter $CEP_{95} = 0.12 \text{ m}$, reduziert. Eine Veröffentlichung wurde eingereicht [134].

Die Verwendung von Verwerfungsschichten in datengetriebenen Verfahren ermöglicht es, die Unsicherheit des Modells über die aktuelle Positionsschätzung zu bewerten. Der Positionsschätzer könnte somit störende Mehrwege aufgrund von Unsicherheiten in der Vorhersage erkennen. In Kombination mit dem sequentiellen Ansatz aus Kapitel 7 können diese Störungen über einen sehr langen Zeitraum sehr genau durch Posenschätzungen überbrückt werden, die im Störfall nur auf den Geschwindigkeiten basieren. Darüber hinaus könnte der Ansatz kontinuierlich neue Funkdaten in Bereichen der Funkumgebung sammeln, in denen der Positionsschätzer hohe Unsicherheiten aufweist. Die auf der Grundlage der Geschwindigkeit geschätzten Positionen könnten als Referenzpositionen verwendet werden, um Unsicherheiten des Positionsschätzers in unbekannten Funkgebieten zu korrigieren.

Eine weitere offene Frage ist, wie genau datengetriebene Verfahren die Höhe schätzen können. Die Experimente in den Kapiteln 4 und 7 nutzen ein Funksystem, das die Höhe des Funkraums nur mit wenigen

Antennen abdeckt, sodass die Auflösung der Höhe nicht hoch ist. Die Funkdaten wurden auch nur im Bereich von 0.56 m bis 2.01 m aufgezeichnet, sodass nicht klar ist, welchen Effekt die vertikale Achse auf die Posengenauigkeit hat. Erste Experimente mit synthetischen Daten haben gezeigt, dass es möglich ist, 3D-Posen aus ToA-Werten zu schätzen. Die Höhe der Posen variiert zwischen 0.10 m und 5.0 m in Schritten von 0.10 m. Im Bereich von 0.10 m bis 0.50 m treten starke Bodenreflexionen auf. Dennoch, die Fehlervarianz in der horizontalen Ebene ($CEP_{95} = 0.14$ m) ist geringer als im sphärischen Raum ($SEP_{95} = 0.27$ m). Die niedrigere Genauigkeit wird durch zu geringes Abtasten der vertikalen Achse durch zu wenige Antennen verursacht.

8.2.4 Geschwindigkeits- und Schrittängenschätzung

Einige Probleme mit Trägheitssensor-basierter Lokalisierung wurden behoben. Es ist jedoch möglich, dass es Situationen gibt, in denen Daten ausfallen, Daten verzögert werden, Messbereiche verschoben werden, Skalierungsänderungen vorgenommen werden müssen und Datensätze nicht jede mögliche Bewegung aufzeichnen, durch die falsche Geschwindigkeiten oder höhere Fehlervarianzen auftreten können. Der hier vorgeschlagene Datensatz enthält viele dieser Fehler, enthält jedoch nicht das Wissen der Welt. Daher ist unklar, wie Hybrid (in Kapitel 5.5) mit unbekannten Situationen umgeht. Diese Unsicherheiten werden bereits von einer Verwerfungsschicht bewertet und können verwendet werden, um herausfordernde Situationen zu identifizieren. Methoden des halbüberwachten Lernens könnten aufgrund dieser Unsicherheiten neue Daten und Beziehungen sammeln. Mithilfe einer Positionsschätzung mit geringer Unsicherheit können die unsicheren Geschwindigkeiten benannt und neu gelernt werden. Auf diese Weise könnte der Geschwindigkeitsschätzer in unbekannten Situationen kontinuierlich genauere Schätzungen erzielen.

Die Dissertation vergleicht aktuelle Methoden von ML und DL mit klassischen Methoden zur impliziten Kompensation von Schwerkraft-, Rausch- und Signalartefakten, um eine hohe und robuste Geschwindigkeitsschätzung sowohl für die kurze als auch für die lange Dauer menschlicher Bewegungen bereitzustellen. Die datengetriebenen Ansätze lernen, einen Geschwindigkeitsvektor mit unterschiedlichen Geschwindigkeiten in dynamischen Szenarien auf eine genaue Referenzgeschwindigkeit abzubilden und sind unabhängig von der

Ausrichtung und Platzierung eines Sensors. Und obwohl dynamische Bewegungen und ausgeprägte Gehgewohnheiten unbekannter Testpersonen die Genauigkeit der Hybrid Methode nicht beeinflussen ($SD \leq 0.01 \text{ m/s}$ über alle Fehlermetriken), kann es dennoch unvorhersehbare reale Situationen geben, die in der Studie nicht behandelt werden. Daher können hier mehr Daten hilfreich sein, um die Grenzen der Generalisierbarkeit zu erweitern, z. B. mit vielen kostengünstigen synthetischen Daten und Umschulungen mit wenigen realen Daten. In unveröffentlichten Experimenten wurde festgestellt, dass die tiefe Kopplung von KF und LSTM-Zellen die Fehlervarianz eines solchen Hybridfilters verringert, Positionen glättet und die Genauigkeit der Unsicherheitsbestimmung erhöht. Die LSTM-Zellen ersetzen die Mess-, Prozess- und Übergangskovarianzmatrizen des Bayesschen Filters und können ein größeres Spektrum der Sensor- und Bewegungsdynamik direkt aus den Daten erfassen.

Im Rahmen der Experimente in Kapitel 7 wurde gezeigt, dass eine Vorhersage zukünftiger Posen möglich ist, um Systemverzögerungen auszugleichen. Um die Aktualisierungsrate der Positionsschätzung stark zu reduzieren, um mehr Kommunikationskapazität für das Funksystem zu ermöglichen, wäre eine fehlerfreie Geschwindigkeits schätzung nötig. Je weiter zukünftige Posen genau geschätzt werden können, desto seltener wird eine aktuelle Position benötigt. Obwohl die Methode in der Trainingsphase Messwerte aus der Vergangenheit auf zukünftigen Referenzen abbilden können, bleibt unklar, wie zukünftige Geschwindigkeiten zur Laufzeit vorhergesagt werden können.

Das Entfernen der Lage des Sensors im Raum (Gravitation) aus den Trägheitsdaten unter Verwendung des Signalgrößenvektors hat gezeigt, dass datengetriebene Verfahren dann wesentlich genauere Geschwindigkeiten schätzen können, da die Ausrichtung in jedem Fenster nicht länger von der Lage des Sensors im Rotationssystem ab hängt. Platzieren des Sensors in einem dynamischen Rotationssystem, z. B. eine Tasche, war ohne den SMV nicht möglich. Die Experimente zeigen, dass es mehrdeutige Orientierungen gibt, die zu falschen Richtungsvektoren der Geschwindigkeit führen und die Ergebnisse verfälschen können. Es ist daher ratsam, vorhandene Methoden mit einem Verwerfungs-Mechanismus neu zu trainieren, um die Unsicherheit bestehender Methoden zu überprüfen. Selbst Methoden, die die Ausrichtung der Trägheitssensoren verwenden und zuverlässig genaue

Schrittängen oder Geschwindigkeiten schätzen, müssen die Sensorik ausnahmslos starr und konsistent an dem zu lokalisierenden Objekt befestigen. Selbst minimale Abweichungen in der Position des Sensors im Rotationssystem führen zu schwerwiegenden Fehlern bei der Schätzung der Richtung des Geschwindigkeitsvektors. Wie Kapitel 5 gezeigt hat, ist es ausreichend, nur den Betrag der Beschleunigung zu verwenden, um eine genaue Geschwindigkeit zu schätzen.

Obwohl viele Experimente zur Geschwindigkeitsschätzung im Rahmen der Arbeit durchgeführt wurden, ist unklar, wie die Methoden mit anderen Arten von Bewegungen funktionieren, z. B. beim Springen, Treppensteigen oder anderen Sportanwendungen. Ein erstes Experiment hat gezeigt, dass Geschwindigkeiten mit Sensoren, die an den Händen eines Benutzers angebracht sind, plötzliche und schnelle Bewegungen während des Sports erfassen können, z. B. Schläge im Boxsport mit einer Genauigkeit von $RMSE = \pm 1.74$ km/h und einer Fehlervarianz von 0.87 km/h konnten anhand von Trägheitssensordaten mit geeigneten statischen Eigenschaften und logistischer Regression geschätzt werden. Auch hier musste die Schwerkraft entfernt werden, die nur zur Klassifizierung der Schlagart herangezogen wird.

8.2.5 Orientierungsschätzung

Es ist keine Einschränkung des Ansatzes, dass die Orientierung nur basierend auf einem einzelnen Bewegungsmuster geschätzt wird. Das in der Dissertation vorgeschlagene Verfahren zur Posenschätzung erfordert keine vollständige (Neu-)Kalibrierung der 3D-Pose, da Madgwick et al. [287] die Nick- und Rollbewegungen mithilfe von Beschleunigungsmesser- und Gyroskopinformationen präzise bestimmen können. Tatsächlich kann der Gier-Winkel langfristig nur fehlerhaft über das Magnetfeld stabilisiert werden, sodass sie durch den vorgeschlagenen Ansatz in Kapitel 6 in Störungsszenarien mit der Position und Klassifizierung von Bewegungsmustern kalibriert werden muss. Die Kalibrierung der Orientierung ermöglicht die Stabilisierung der Orientierung der Trajektorie über lange Zeiträume und ermöglicht auch das Zusammenführen von Sensoren über verschiedene und hochdynamische Koordinatensysteme hinaus, wie z. B. Kopf und Rumpf in Sport- und VR-Anwendungen. Im Vergleich zu modernen KF [266], die die nichtlineare Kopf-Körper-Pose über die Zeit nicht korrekt abschätzen können, akzeptiert die Methode

unmerkliche Drift in den Sensorsignalen und lernt Momente, um verrauschte Sensordaten zu abstrahieren, die nichtlineare Bewegungen darstellen, und lernt zuverlässig diese zur Laufzeit erneut zu erkennen und verwendet sie gelegentlich, um akkumulierte Fehler zu korrigieren.

Die Genauigkeit der Klassifizierung zur Orientierungsschätzung nimmt mit der Vielzahl von Daten zu, auf die das Modell trainiert wird, z. B. mehr Teilnehmer, mehr Variationen und verschiedene Arten von Bewegungen. Das Sammeln ausgewogenerer Daten und das (erneute) Trainieren des Klassifizierungsmodells zur Laufzeit verbessert daher die Effektivität des Klassifizierers. In einem kontinuierlichen Lernansatz mit Positionsschätzungen mit geringer Unsicherheit könnten Trainingsdaten für bekannte Bewegungsszenarien gesammelt und für das (Nach-)Training der Orientierungsschätzung verwendet werden. Die Auswertung im Abschnitt 6.8.5 zeigt jedoch, dass die Methode im täglichen Gebrauch auch bei unbekannten Bewegungen unbekannter VR-Benutzer bereits gut funktioniert. Es hängt daher von der Anwendung, nämlich den Bewegungsmustern und Sensorplatzierungen ab, ob zusätzliche Daten erforderlich sind.

Die Transformation der Position und der Orientierung zwischen den Koordinatensystemen ist nicht fehlerfrei möglich. Das auf fehlerhaften absoluten Positionen basierende absolute Koordinatensystem ist falsch (siehe Kapitel 4) und wird zur Kalibrierung der Orientierung verwendet. In den Anwendungen der Dissertation ist diese Beziehung jedoch nicht kritisch, da der Fehler mit zunehmendem Abstand zwischen zwei Positionen linear abnimmt. Im Durchschnitt wurde ein Abstand von 1.25 m zwischen zwei aufeinanderfolgenden Referenzpositionen pro Fenster (Fenstergröße = 1.28 s) bestimmt. Die Orientierungsschätzung hat dann einen durchschnittlichen Fehler von 2.6 °. Der maximale Fehler beträgt 4.1 °, wenn die Start- und Endpositionen den größtmöglichen Fehler von ± 0.20 m aufweisen und den größten Richtungswinkel zueinander überspannen. Es ist jedoch ratsam, im Voraus Grenzwerte für die Verwendung der Orientierungsschätzung festzulegen. Eine Unsicherheitsschätzung der Orientierung könnte dazu beitragen, Fehler in der Modellkalibrierung in die Vorhersageergebnisse einzubeziehen, um zu beurteilen, ob die Orientierungsschätzung zuverlässig ist und ob sie zur Kalibrierung einer falschen Position verwendet werden sollte. Das Verfahren kann auch sehr ungenaue GPS- oder Wi-Fi-basierte absolute Positionen

verwenden, wenn der Positionsvektor so gewählt wird, dass er ausreichend groß ist, d. h. längere Positionsbeobachtungsintervalle. Da der Positionsvektor den tatsächlichen absoluten Gier-Winkel darstellt, muss sichergestellt werden, dass sein Fehler gering ist, z. B. liefert GPS mit einem $CEP_{95} = 1.9$ m nach einer zurückgelegten Strecke eine durchschnittliche Positionsgenauigkeit von 0.38 m ab 10 m. Die Merkmale müssen jedoch für so lange Eingabesequenzen angepasst werden, da bspw. Spitzen in einem Mittelwertmerkmal keinen signifikanten Einfluss mehr haben. Methoden wie LSTM oder Attention, siehe FRFNN in Kapitel 7, könnten hier hilfreich sein.

Typischerweise werden Varianten der SVM so auf die Daten kalibriert, dass sie sehr geringe Unsicherheiten für Vorhersagen mit einem Vertrauen von mehr als 0.75 % und sehr hohe Unsicherheiten für Vorhersagen mit einem Vertrauen von $\leq 50\%$ liefern. Diese Beziehung wurde in Kapitel 6 ausgenutzt. Es wurde eine SVM verwendet, deren Klassifizierungsergebnisse nur zur Orientierungsschätzung mit sehr hohen Konfidenzwerten verwendet werden. Wenn das Modell quasi ein Klassifizierungsergebnis errät (Konfidenz um 50%), ist die Unsicherheit sehr hoch und die Ergebnisse werden nicht verwendet. Im Gegenteil, wenn das Klassifizierungsergebnis sehr zuverlässig ist, ist die Unsicherheit sehr niedrig. Trotzdem könnte eine sequentielle logistische Regression verwendet werden, um das SVM-Modell über das gesamte Konfidenzintervall zu kalibrieren. Die SVM-Klassifizierungsergebnisse werden dann verwendet, um einen logistischen Regressor zu trainieren, um Schätzungen mit hoher Sicherheit und geringer Unsicherheit aus der kalibrierten logistischen Regression zu erhalten.

In ersten Experimenten wurde gezeigt, dass die Klassifizierung der Kalibriermomente auch mit einer datengetriebenen Methode durchgeführt werden kann. Hierfür wurde die FRFNN-Methode aus Kapitel 7 verwendet. Anstelle der Regressionsschicht am Ende der FRFNN-Architektur wurde eine Softmax-Schicht zur Klassifizierung verwendet. Ein Vergleich der Klassifizierungsgenauigkeit von FRFNN für die in Kapitel 6 verwendeten Merkmale mit den Ergebnissen bei direkter Verwendung der Rohdaten für Beschleunigung und Gierrate hat gezeigt, dass FRFNN mit Merkmalen ($F_1: 94.8\%$) nahezu identische Genauigkeiten erreichen wie mit Rohdaten ($F_1: 96.3\%$). Dies deutet an, dass die Merkmale die wesentlichen Informationen für eine korrekte Klassifizierung enthalten. Der Aufwand einer Methode des tiefen Lernens (Skalierung)

rung, Datenzusammensetzung) ist daher nicht zwingend erforderlich. FRFNN reduziert jedoch die Verzögerung, die die SG- oder IIR-Filter zu den Signalströmen hinzufügen (Verzögerung von maximal 125 ms , d. h. 1/8 eines Schiebefensters), da FRFNN implizit die Filterung von Eingabesequenzen mit Trägheitsrohdaten durchführt.

8.2.6 Fusion multimodaler Sensordaten

Herkömmliche Verfahren aus dem Stand der Technik haben Schwierigkeiten, Sensordatenströme von verschiedenen Sensoren zusammenzuführen, um bspw. eine Pose oder Trajektorie zu rekonstruieren. Kapitel 7 zeigt, dass die Verarbeitungskette dieser Dissertation die Stärken datengetriebener Methoden nutzt, um optimale Posen zu schätzen. Herkömmliche sequentielle Verfahren haben jedoch den Vorteil, dass jede Komponente ausgetauscht werden kann und die Positionsschätzung langfristig fehlschlagen kann und dennoch eine ziemlich genaue Pose geschätzt werden kann. Auf diese Weise können spezifische Änderungen berücksichtigt und einzelne Komponenten der Verarbeitungskette ausgetauscht werden. Im Gegensatz dazu erfordert die Anpassung von FRFNN eine zeitaufwändige Umschulung. FRFNN ist aber präziser als die modellbasierte PDR Version, da es die verschiedenen Messungen implizit optimal kombiniert.

Die Experimente in Abschnitt 7.8 zeigen, dass es sinnvoll ist, künstliche Datensätze zu generieren, um alle realistischen Geschwindigkeiten und Bewegungsformen mit einer Sequenzlänge abzudecken, die lang genug ist, um Formen und Geschwindigkeiten aufzunehmen aber kurz genug, um alle Bewegungsformen zu rekonstruieren. Bspw. kann ein Kreis (Kreisverkehr) aus Kurven und ein Rechteck (Ampelkreuzung) aus geraden Linien rekonstruiert werden. Die verschiedenen Daten müssen jedoch a priori skaliert werden, und die Skalierung muss auch für die unbekannten Daten einer Live-Phase gelten. Die Eingabesequenzen sind idealerweise so skaliert, dass bspw. die Werte für Geschwindigkeit, Beschleunigung und Position im gleichen Wertebereich liegen. Positionen und Beschleunigungen müssen ähnliche Wertebereiche und Werte haben, um die Verlust- und Kostenfunktion gleichermaßen zu beeinflussen. Erste Versuche haben gezeigt, dass durch eine einfache Skalierung der maximal möglichen Werte, anhand der Sensormessbereiche, der maximal möglichen Bewegungsgeschwindigkeit und der maximalen Entfernung der Funkverbindung, die

Wertebereiche optimal skaliert werden. Trägheitssensoren werden jedoch beim Einschalten speziell für die Umgebung kalibriert und der Wertebereich wird somit zufällig variiert. Hier könnten Methoden des Transferlernens verwendet werden, um datengetriebenen Modell auf neue Sensormerkmale innerhalb einer Domäne fein zu justieren.

Da die Positions-, Geschwindigkeits- und Orientierungsschätzer unterschiedliche Fensterüberlappungen, Aktualisierungsraten und Übertragungsverzögerungen aufweisen können, können ihre Datenraten und zeitlichen Beziehungen variieren. Erste Experimente haben gezeigt, dass datengetriebene Methoden diese Datenlücken implizit schließen können. Es ist jedoch unklar, wie viel Kapazität eines Netzes dafür benötigt wird und ob wertvolle Funktionen dadurch verloren gehen. Eine mögliche Lösung könnte sein, die Abtastrate aller Informationen signifikant niedriger zu halten und eine Extrapolationsfunktion zum Erzeugen von Eingabesequenzen zu verwenden. Selbst wenn von einigen Sensoren viele Daten verfügbar sind, könnten die Sensoren mit wenigen Daten ausgeglichen werden. Im Gegensatz zu datengetriebene Methoden kann die sequentielle Methode Funkpositionsfehler von über 60 s überbrücken. Dies ist mit dem vorgeschlagenen FRFNN-Modell nicht möglich, da dies eine Sequenzlänge von 6,000 Werten erfordern würde, um mindestens 2 Funkpositionen in einer Sequenz zu enthalten. Experimente haben gezeigt, dass Sequenzlängen von mehr als 128 Werten bereits die Speicherkapazität des Kontextvektors des Einzelschichtmodells überlasten. Darüber hinaus sollte eine optimale Lernstrategie die kleinsten gängigen Bewegungsformen lernen, um komplexere zufällige Bewegungsmuster zu rekonstruieren.

Um die Rekonstruktion von Trajektorien in jeder Raumgröße zu ermöglichen, ist es nicht erforderlich, mit absoluten Positionen in den Eingabesequenzen zu arbeiten. Erste vorläufige Studien haben gezeigt, dass die Methode nur dann eine Aktualisierung der Funkposition pro Fenster erfordert, wenn die Position nicht direkt in die Eingabesequenz des Modells integriert ist, sondern nur das Delta, der gerichtete Abstand zwischen zwei Positionen, zwischen aufeinanderfolgenden Positionen. Anstatt eine Position vorherzusagen, wird dann das Delta zur nächsten Position vorhergesagt, das an eine Startposition und an jede neu geschätzte Position angehängt wird. Auf diese Weise kann auch der notwendige Skalierungsaufwand reduziert werden, da die pro Fenster zurückgelegte Strecke keine großen Werte erzeugt.

8.3 Beiträge

Dieser Abschnitt diskutiert die Beiträge der Dissertation. Zunächst erörtert Abschnitt 8.3.1 allgemeine Beiträge. Abschnitt 8.3.2 diskutiert die Beiträge der Funkpositionsschätzung. Dann diskutiert Abschnitt 8.3.3 die Beiträge der Geschwindigkeitsschätzung. Abschnitt 8.3.4 erörtert die Beiträge der Orientierungsschätzung und Abschnitt 8.3.5 diskutiert die Beiträge der Fusion. Abschließend diskutiert Abschnitt 8.3.6 die methodischen Beiträge.

Diese Arbeit hat gezeigt, dass es möglich ist, die absolute Trajektorie eines Fußgängers in Innenräumen zu rekonstruieren, selbst wenn die lose gekoppelte Funk- und Trägheitssensorik dynamischen und abrupten Bewegungsänderungen ausgesetzt sind und unter Mehrwegepfaden in der Funkumgebung und Störungen des lokalen Magnetfeldes leiden. Dies wurde erreicht, indem Funklokalisierungs- und PDR-Techniken mit Ideen aus dem Bereich des maschinellen und tiefen Lernens kombiniert wurden. Das in dieser Arbeit entwickelte Lokalisierungssystem für Fußgänger besteht aus vier wesentlichen Komponenten: der Positionsschätzung aus Ankunftszeiten eines Funksystems, der Geschwindigkeitsschätzung aus Betragssgrößenvektoren eines Beschleunigungssensors, der Orientierungsschätzung aus Drehraten und Beschleunigungssignalen und der Fusion der drei Schätzer zur Rekonstruktion der Trajektorie von Benutzern der virtuellen Welt, Sportlern, Fußgängern und autonomen Fahrzeugen.

Probleme des Stands der Technik

Die Methoden zur Positions- (Kapitel 4), Geschwindigkeits- (Kapitel 5), Orientierungs- (Kapitel 6) und Posenschätzung (Kapitel 7) befassen sich mit den folgenden Problemen des Stands der Technik.

SINS akkumulieren Sensorrauschen, indem sie Beschleunigungen zweimal integrieren, um eine Entfernung zu erhalten. PDR-Methoden erfordern Sensoren an den Füßen, um diese akkumulierten Fehler zu kalibrieren und zurückzusetzen, wenn die Füße mit dem Boden in Kontakt kommen. Daher fordert eine genaue Geschwindigkeitsschätzung Sensoren an den Füßen des Benutzers. Der Benutzer darf nicht laufen, da sich der Gang ändert, kein Bodenkontakt für die Schritterkennung vorhanden ist und integrale Verfahren mit ZUPT nicht

korrigiert werden können. Dies führt zu schwerwiegenden Fehlern bei der Geschwindigkeitsschätzung.

Zusätzlich zur Schrittänge kann die Schrittrichtung langfristig nicht richtig geschätzt werden. Herkömmliche Verfahren liefern ungenaue Orientierungen, insbesondere bei sich schnell ändernden Bewegungen, elektromagnetischen Störquellen oder lose platzierten Sensoren. Zusammen mit der falschen Schrittänge werden Trajektorien rekonstruiert, die nach kurzer Zeit von der Wahrheit abweichen.

Heute, 2021, tragen die meisten Menschen unbemerkt viele Sensoren an ihrem Körper, die die Bewegung in verschiedenen Rotations- und Koordinatensystemen aufzeichnen. Um die Informationen zusammenzuführen, müssen die Systeme kalibriert werden. Dies ist für herkömmliche Methoden in dynamischen Situationen unmöglich, da natürliche Bewegungsformen nicht deterministisch beschrieben werden können. Die Zusammenführung mehrerer Sensordatenströme von einem oder mehreren Sensoren ist ebenso schwierig, da jeder Sensordatenstrom aufwendig deterministisch beschrieben werden muss. Viele Sensorkonfigurationen können nur angenähert werden und entsprechen nicht der Realität. Bereits kleinste Abweichungen führen über lange Zeiträume zu schwerwiegenden Fehlern.

Ein weiterer Schwachpunkt liegt im Umgang mit Stör- und Rauschquellen. Herkömmliche Methoden können nichtdeterministisches, stochastisches Rauschen nicht beschreiben, da zufälliges Rauschen theoretisch nur angenähert werden kann. Darüber hinaus können zufällige Irrfahrten aufgrund temperaturabhängiger Schwankungen der Messungen auch bei herkömmlichen Methoden nur annähernd beschrieben werden. Herkömmliche modellgetriebene Verfahren haben Schwierigkeiten bei der Beschreibung nichtlinearer Beziehungen, und herkömmliche datengetriebene Verfahren haben Probleme bei der Beschreibung nichtlinearer Beziehungen aufgrund fehlender Daten.

Vorgeschlagene Lösungen. In dieser Arbeit wurde eine Reihe von Techniken vorgestellt, die die Leistung von Funkortungs- und Trägheitsverfolgungssystemen verbessern. Die wichtigste davon ist die Vorhersagbarkeit zukünftiger Posen mit rekurrenten datengetriebenen Methoden aus niederfrequenten Positionen und höherfrequenten Geschwindigkeiten (Kapitel 7). Infolgedessen können die höheren

Rechenanforderungen datengetriebener Verarbeitungsketten sowie Mess- und Systemverzögerungen reduziert werden und dennoch die bestmögliche Genauigkeit erzielt werden (Kapitel 7). Es wurde auch gezeigt, dass die Kombination von Positionen mit Bewegungsklassifizierung selbst in stark magnetisch gestörten Umgebungen zu stabilen Orientierungsschätzungen führen kann (Kapitel 6). Darüber hinaus konnte gezeigt werden, dass exakte Positionen vor Ankunftszeiten von Funksignalen geschützt werden können, selbst wenn diese unter dem Einfluss der Mehrwegeausbreitung leiden (Kapitel 4). Ein zentraler Teil der Arbeit hat gezeigt, dass die Trägheitsnavigation über den Stand der Technik hinaus erheblich verbessert werden kann, wenn die Referenzgeschwindigkeit mit aktuellen datengetriebenen Methoden direkt aus den Daten abgebildet wird (Kapitel 5). Schließlich wurde gezeigt, dass das datengetriebenen Zusammenführen verschiedener Sensordatenströme zu einer genaueren Rekonstruktion von Trajektorien führt als der Stand der Technik (Kapitel 7).

Der Zweck der Entwicklung des beschriebenen Lokalisierungssystems für Fußgänger bestand darin, die im Abschnitt 1.3 gestellten Forschungsfragen zu beantworten. Diese Fragen werden in den folgenden Abschnitten 8.3.1 bis 8.3.6 wiederholt, diskutiert und die wichtigsten Beiträge und Schlussfolgerungen dieser Arbeit zusammengefasst sind.

8.3.1 Allgemeine Forschungsfragen

RQ1: Wie schnell und wie groß häufen sich Orientierungs- und Abstandsfehler in einem Trägheitsnavigationssystem an, wenn aktuelle MEMS-Trägheitssensoren nach dem Stand der Technik (modell- und datengetrieben) verwendet werden?

Diese Forschungsfrage wird von T. Feigl in einer bereits veröffentlichten Arbeit [125] behandelt.

Eine der wichtigen Schlussfolgerungen dieser Arbeit ist, dass die uneingeschränkte Trägheitsnavigation (nur auf der Grundlage von Beschleunigungs- und Gierraten) eine ungeeignete Technik ist, um die relativen Bewegungen eines Fußgängers mit aktuellen modell- und datengetriebenen Trägheitssensor technologien zu verfolgen, wenn eine zentimetergenaue Lokalisierung gefordert ist.

Kapitel 6.3 zeigt, dass bei einer aktuellen MEMS-IMU nach dem Stand der Technik kaum Orientierungsfehler bei linearer Bewegung, bspw. langsames statisches Gehen, auftreten. Im Gegensatz dazu akkumulieren die Sensoren nach mehr als 5 min eine signifikante Fehlausrichtung von 25° , wenn die Sensoren am Kopf getragen werden, während der Träger spazieren geht. Wenn Menschen ihre Bewegungsrichtung schnell und abrupt ändern, häufen sich Orientierungsfehler schneller (5.5° pro 1 min).

Kapitel 5.7 zeigt, dass sich mit einer aktuellen MEMS-IMU nach dem Stand der Technik auch bei linearen Bewegungen (Distanzfehler, DET: PDR=81.51, RoNIN=5.67 und Hybrid=2.26 [m]) signifikante Distanzfehler ansammeln. Bei dynamischen Bewegungen mit vielen Richtungs- und Geschwindigkeitsänderungen akkumulieren selbst die datengetriebenen Methoden signifikante Distanzfehler (DET: PDR=141.02, RoNIN=75.11 und Hybrid=11.47 [m]). Die Fehler entstehen aufgrund von Messfehlern, die bei jedem der einzelnen Beschleunigungsmesser und Gyroskope auftreten, die in einer MEMS-IMU enthalten sind. Diese einzelnen Fehler breiten sich bei klassischen Verfahren durch die Trägheitsnavigationsgleichungen und bei datengetriebenen Methoden durch epistemische¹ und aleatorische² Unsicherheiten der Modelle aus. Die Fehler verursachen Orientierungs- und Positionsverschiebungen, die kurzfristig proportional zur Bewegungsperiode (wenn die Beschleunigungsmesser die Hauptursache für die Drift sind) und langfristig (wenn die Gyroskope die dominante Fehlerquelle sind) proportional zur Bewegungsperiode zunehmen.

Angesichts der Tatsache, dass viele Lokalisierungsanwendungen in Innenräumen über einen längeren Zeitraum eine Genauigkeit von wenigen Millimetern erfordern, ist eine uneingeschränkte MEMS-basierte Trägheitsnavigation von Fußgängern in Innenräumen selbst mit den vorgeschlagenen neuen datengetriebenen Verfahren (CNN-RNN und

¹Epistemische Unsicherheiten ergeben sich aus Wissenslücken, bspw. aus ungenauen Modellparametern, die mit Methoden der Fuzzy-Arithmetik beschrieben werden können. Z. B. Abweichungen zwischen dem gelernten Modell und der Referenz der Realität. Die epistemische Unsicherheit kann normalerweise systematisch durch geeignete Daten sowie durch Training- und Testzusammensetzung beseitigt werden. [2]

²Aleatorische Unsicherheiten sind zufällige und natürliche Schwankungen, die durch stochastische Verteilungsfunktionen beschrieben werden können. Z. B. Messunsicherheit und -abweichung in physikalischen Messdaten. Die aleatorische Unsicherheit kann normalerweise nur minimiert, aber nicht vollständig beseitigt werden. [2]

Hybrid, siehe Kapitel 5) unmöglich. Neue Technologien zur Kompensation lokaler Gravitationsschwankungen und Sensoren mit höherer Genauigkeit müssen daher entwickelt werden.

8.3.2 Forschungsfragen zur Funkpositionsschätzung

Um möglichst genaue Positionen aus Funksignalen zu schätzen, wird die funk basierte Verarbeitungskette in kleine atomare Komponenten aufgeteilt. Statt Ende-zu-Ende-Lernen von der Kanalimpulsantwort direkt zur Position werden die wesentlichen Komponenten, die zur Beschreibung des Funkkanals erforderlich sind, durch einzelne Komponenten der Verarbeitungskette modelliert und somit besser kontrolliert. So können zunächst Ankunftszeiten aus den Kanalimpulsantworten extrahiert werden, um dann Positionen aus diesen Ankunftszeiten der Funksignale zu ermitteln.

RQ2: *Können datengetriebene Verfahren die Positionsschätzung eines dynamisch bewegten Funksenders aus verrauschten Ankunftszeitmessungen verbessern?*

Diese Forschungsfrage wird von T. Feigl in bereits veröffentlichten Arbeiten [127, 247] und einer eingereichten Arbeit [134] behandelt. Im Rahmen dieser Arbeiten wurde auch ein Patent eingereicht [132].

Kapitel 4 zeigt eine datengetriebene Methode, die Positionen von synthetischen und realen Ankunftszeiten bestimmt, die geringe (Bodenreflexionen und Bewegungen des Senders) bis zu starken (gezielte räumliche Störungen) Verzögerungen aufgrund von Mehrwegeausbreitung enthalten. Es wurde gezeigt, dass rekurrente neuronale Netze, insbesondere LSTM-Zellen, Positionen aus vorverarbeiteten Funksignalen bestimmen können, insbesondere die synchronisierten Ankunftszeiten. Die Wendepunktmethode eines Funkortungssystems stellt ToA-Werte mit geringer Genauigkeit ($MAE = \pm 0.27 \text{ m}$) bereit (siehe Abschnitt 3). Das vorgestellte RNN mit einem gestapelten LSTM-Modell kann (nach dem Training) genauere aufeinanderfolgende Positionen (Rechtecke: $MAE = 0.024$, $MSE = 0.001 \text{ [m]}$) aus diesen rohen und driftenden ToF-Messungen vorhersagen als der Stand der Technik ($MAE = 1.042$, $MSE = 2.448 \text{ [m]}$). Datengetriebene Methoden verbessern daher die ToA-basierte Positionsschätzung erheblich.

Die Ergebnisse zeigen, dass datengetriebene Verfahren eine Abbildungsfunktion direkt aus den Daten lernen können, die Ankunftszeiten mit und ohne Rauschen genauer auf eine Position abbilden können als Verfahren des Stands der Technik. Dies funktioniert auch bei Bewegungen, die sich abrupt und plötzlich ändern (Zickzack: MAE=0.058, MSE=0.006 [m]). Erste Experimente haben gezeigt, dass Ankunftszeiten, die andere als die erlernten Bewegungsformen darstellen, mit hoher Genauigkeit (Spaziergang: MAE=0.078, MSE=0.011 [m]) auf unbekannte Positionen durch LSTMs abgebildet werden können.

RQ2a: Kann eine datengetriebene Methode Mehrwegepfade in Ankunftszeiten in Funksystemen auflösen, um die Positionsgenauigkeit in gestörten Sichtverbindungen zu erhöhen?

Kapitel 4 zeigt, dass RNN, insbesondere LSTM-Zellen, Positionen auch aus synchronisierten Ankunftszeiten mit synthetischen und realen Mehrwegekomponenten (mit NLoS Komponenten) bestimmen. Experimente haben gezeigt, dass gestapelte LSTMs (zwei Schichten) sogar vorübergehend (<10 s) vollständige Ausfälle von Empfängern (Ausfall von bis zu 6 von 12 Antennen), die durch 0-Werte in den Eingabesequenzen dargestellt werden, eine unveränderte Schätzgenauigkeit aufweisen ($SD = 0.0017 \text{ m}$). Das vorgestellte LSTM übertrifft ein optimiertes KF aus dem Stand der Technik (LSTM: MAE=0.091, MSE=0.002 gegenüber KF: MAE=0.163, MSE=0.042 [m]).

Die Ergebnisse zeigen, dass LSTMs robust sind sowohl gegen eine schwache (aufgrund von Bewegung und Bodenreflexionen) als auch eine starke (aufgrund der Abdeckung des Senders) Mehrwegeausbreitung. LSTMs liefern selbst bei einem vollständigen Ausfall einzelner Sender-Empfänger-Verbindungen genaue Positionsschätzungen. Im Gegensatz zu KF korrigieren sie Positionsfehler oder erhöhen die Positionsgenauigkeit in Mehrwegesituationen. Allerdings generalisieren sie nicht auf unbekannte Funkumgebungen, Signal- und Kanalparameter und Bewegungsformen.

RQ2b: Kann eine datengetriebene Methode langzeitabhängiges stochastisches Rauschen in synthetischen Ankunftszeiten auflösen, um die Positionsgenauigkeit zu erhöhen?

Kapitel 4 zeigt, dass die rekurrente datengetriebene Methode stochastisches Rauschen interpretieren oder entfernen kann (zufällige

Verzögerungen bei den Ankunftszeiten aufgrund nicht deterministischer Systemfehler und Mehrwegeausbreitung). Das optimierte LSTM liefert präzisere und robustere Positionen als typische Verfahren aus dem Stand der Technik (LSTM: MAE=0.036 gegenüber KF: MAE=0.163 [m]). Es kann stochastisches Irrfahrten Rauschen aus Funksignalen, insbesondere den synchronisierten Ankunftszeiten mit Mehrwegekomponenten entfernen, die NLoS Verbindungen enthalten (LSTM: MAE=0.024, MSE=0.001 gegenüber KF: MAE=1.042, MSE=2.448 [m]).

Die Ergebnisse aus Abschnitt 7.7 zeigen, dass LSTM-Zellen zufälliges Rauschen entfernen können, da die Positionsgenauigkeit mit und ohne Mehrwegekomponenten und mit und ohne zufälliges Rauschen quasi identisch ist. Es könnte also sein, dass sich die LSTM-Zellen auf die wesentlichen Informationen in den Signalströmen konzentrieren, das Rauschen ignorieren, um genaue Positionen zu schätzen. Die Auswertung der Experimente zeigt, dass die Modelle lernen können, mit zufälliger Signaldrift umzugehen und optimierte KF-Filter aus dem Stand der Technik zu übertreffen.

RQ2c: Welche Auswirkung hat die Architektur einer rekurrenten datengeriebene Methode auf Positionsgenauigkeit?

Die Experimente in Kapitel 4 zeigten, dass flachere Modelle (mit weniger Schichten) das hochkomplexe Langzeitrauschen schlechter lernen. D. h., sie sind kaum in der Lage, das Rauschen von der Bewegung zu trennen und liefern somit schlechtere Ergebnisse (LSTM-1L: Synthetisch D_{rd} : MAE=0.091 [m]) als vergleichbare tiefere Modelle (LSTM-4L: Synthetisch D_{rd} : MAE=0.036 [m]). Außerdem verallgemeinern sie das Verhalten bei körperlicher Aktivität schlechter, d. h. die Zickzack-Kurvenfahrt mit plötzlichen Richtungsänderungen war signifikant ungenauer (LSTM-1L: MAE=0.127, MSE=0.024 gegenüber LSTM-4L: MAE=0.058, MSE=0.006 [m]). Stattdessen ordnen flache Modelle die ToF-Eingabedaten nur den Ausgabedaten (Verzögerungen oder Positionen) zu, ohne tiefere Wechselbeziehungen (z. B. Beschleunigung, Geschwindigkeit und Richtung) verwandter Messungen zu berücksichtigen. Andererseits konnten flache Modelle das Langzeitrauschen nicht von den Bewegungsinformationen trennen. Im Gegensatz dazu wurde die hier vorgeschlagene optimale LSTM-Architektur erfolgreich auf rechteckige, sinusförmige, elliptische und Trajektorien mit zufälligen Bewegungsformen angewendet und validiert.

Die Ergebnisse aus Abschnitt 4.7 zeigen, dass die Architektur einen direkten Einfluss auf die Genauigkeit und Robustheit der Positions schätzung. Es ist ratsam, die Architektur anhand der Anwendung und der zugrunde liegenden Daten zu optimieren. Die Kapitel 5 und 7 zeigen, dass eine besonders tiefe Architektur mit vielen Parametern nicht unbedingt erforderlich ist und einfache Architekturen den Rechenaufwand verringern. Der vorgeschlagene datengetriebene Ansatz adressiert und löst die grundlegenden Probleme der Funklokalisierung: Phasen- und Amplitudenverschiebungen aufgrund von Mehrwegekomponenten und beweglichen Funksendern oder gestörten Betrachtungs bedingungen, die zu langfristigem Signaldrift führen.

8.3.3 Forschungsfragen zur Geschwindigkeitsschätzung

RQ3: Können datengetriebene Verfahren die Geschwindigkeits schätzung eines Fußgängers aus verrauschten MEMS Trägheitsmessungen verbessern?

Diese Forschungsfrage wird von T. Feigl in bereits veröffentlichten Arbeiten [128, 131] behandelt.

Kapitel 5 zeigt, dass datengetriebene Methoden die Geschwindigkeit von Fußgängern genauer und zuverlässiger bestimmen können als optimierte herkömmliche Methoden. Insbesondere in Situationen, in denen das Sensorsystem nicht fest mit dem Fußgänger verbunden ist, liefern datengetriebene Verfahren wesentlich präzisere Geschwindigkeiten als modellgetriebene Verfahren (unbekannte Testperson: Hybrid MAE=0.19, MSE=0.09 gegenüber PDR: MAE=2.10, MSE=6.95 [m]). Die datengetriebenen Verfahren verbessern auch die relativen Fehler und liefern über einen längeren Zeitraum signifikant geringere Distanzfehler (unbekannte Testperson DET: Hybrid = 1.54, PDR = 914.55 [m]). Datengetriebene Verfahren können die Geschwindigkeit und Entfernung von Beschleunigungssensoren genauer und robuster schätzen als die Verfahren des Stands der Technik (unbekannte Test person DEPM: PDR = 0.374, Hybrid = 0.024 [m]).

Die Ergebnisse aus Abschnitt 5.7 zeigen, dass die Kombination von CNN- und RNN-Methoden (C/RNN) die Geschwindigkeit viel genauer schätzt als Netze mit Resteinheiten (RoNIN). Die Kombination von

Resteinheiten und BLSTM (Hybrid) kann die Genauigkeit noch weiter erhöhen (unbekannte Testperson, MAE=C/RNN = 0.30, RoNIN = 0.58, Hybrid = 0.19 [m]). Es ist daher ratsam, immer Zustandsverfolger wie rekurrente Netze zu verwenden, wenn die Informationen in einer Eingabesequenz zeitliche Verbindungen aufweisen.

RQ3a: *Kann die Gravitationskraft eines lose befestigten Beschleunigungsmessers so eliminiert werden, um in Sport- und VR-Anwendungen mit hochdynamischen Bewegungsänderungen die Geschwindigkeit genau zu bestimmen?*

Kapitel 5 zeigt, dass SMV (der Betrag der gemessenen Rohbeschleunigungswerte über alle Sensorachsen) den Einfluss der Gravitationskraft auf die Geschwindigkeit eliminieren kann. SMV ermöglicht das Entfernen der vorzeichenbehafteten Gravitationskraft und das lose Platzieren der Sensoren auf einem Fußgänger. Selbst Experimente mit Fußgängern, die die MEMS-IMU in der Tasche tragen, führen zu sehr genauen Geschwindigkeitsschätzungen (unbekannte Testperson CEP_{95} : Hybrid = 0.56 [m]), während nicht normalisierte Eingabesequenzen keine plausiblen Ergebnisse liefern konnten. Die Experimente zeigen, dass die Gravitationskraft bis zu einem gewissen Grad eliminiert werden kann, da die genaue Gravitationskraft für jeden Sensor variiert und a priori gemessen werden müsste, um sie genau zu subtrahieren. Das Entfernen der theoretischen Schwerkraft von 9.81 m/s^2 in Deutschland führte jedoch zu sehr genauen Geschwindigkeitsschätzungen.

Die Ergebnisse aus Abschnitt 5.7 zeigen, dass der vorgeschlagene datengetriebene Ansatz die grundlegenden Probleme der trägebasierter Lokalisierung adressiert und löst: Kurz- und Langzeitfehler aufgrund der Integration von Sensorrauschen und falscher Bestimmung der Gravitationskraft. Diese Arbeit löst durch den SMV diese Probleme auch bei hochdynamischen Bewegungen.

8.3.4 Forschungsfragen zur Orientierungsschätzung

RQ4: *Können datengetriebene Verfahren die Orientierungsschätzung eines Fußgängers aus verrauschten MEMS-Trägheitsmessungen verbessern?*

Diese Forschungsfrage wird von T. Feigl in bereits veröffentlichten Arbeiten [125, 126, 124, 122] behandelt. Im Rahmen dieser Arbeiten

wurden drei Patente eingereicht [119, 123, 121].

Kapitel 6 zeigt, dass die Kombination eines Funkpositionsdatenstroms und hoher Trägheitsmessungen eine Orientierung langfristig stabilisieren kann, wenn Bewegungsmuster erkannt werden können, die die aktuelle Bewegungsform darstellen, z. B. gehe geradeaus und schaue nach vorne, gehe seitwärts und schaue seitwärts. Datengetriebene Verfahren, z. B. eine Stützvektormaschine, können die aktuelle Bewegungsform mit hoher Genauigkeit (F1-Punktzahl: 95 %) anhand statistischer Merkmale (Mittelwert, Varianz und PCA) für Beschleunigungs- und Rotationsgeschwindigkeitsmessungen klassifizieren. Erste Experimente haben auch gezeigt, dass BLSTM Netze in 9.97 von 10 Fällen die aktuelle Gangphase einer Person richtig klassifizieren können. Mit einer Fehlervarianz, die von der Positionsgenauigkeit abhängt (siehe Kapitel 6), kann die Orientierung genau geschätzt werden ($MAE=2.6^\circ$) und zur Kalibrierung verwendet werden.

Die Ergebnisse in verschiedenen Anwendungen (siehe Kapitel 8.4) bekräftigen, dass das vorgeschlagene Verfahren die Orientierung aus verrauschten MEMS-Trägheitsmessungen langfristig genau bestimmen kann, wenn die Bewegungsrichtung eines verbundenen Rotationssystems (z. B. Körper) und die Beziehung zwischen den Rotationssystemen (z. B. Kopf und Körper) bekannt ist.

RQ4a: *Mit welcher Genauigkeit können datengetriebene Methoden die Ausrichtung von Kopf-zu-Rumpf mit am Kopf platzierten Beschleunigungs- und Rotationsratensensoren abschätzen?*

Die Genauigkeit der Orientierungsschätzung hängt von zwei Faktoren ab: Zum einen von der Genauigkeit der Positionsbestimmung zwischen dem Rotationssystem des Kopfes und dem Oberkörper. In Kapitel 6 wurde die Genauigkeit der Klassifizierung bestimmt, die die Beziehung zwischen Kopf und Körper darstellt (F1-Punktzahl: 95 %). Andererseits liefert der Vektor, der von zwei aufeinanderfolgenden Positionen überspannt wird, die Referenzorientierung für die Korrektur der Orientierungsschätzung. Je größer die Fehlervarianz in der Positionsschätzung ist, desto größer ist der Fehler in der Referenzorientierung. Die Orientierungsschätzung hat dann einen durchschnittlichen Fehler von 2.6° . Der maximale Fehler beträgt 4.1° , wenn die Start- und Endpositionen den größtmöglichen Fehler von ± 0.20 m aufweisen und den größten Richtungswinkel zueinander überspannen. In Kapitel 6 wurde

gezeigt, dass die Fehler auch bei nicht optimierten Lokalisierungssystemen vernachlässigt werden können, sodass auch Anwendungen mit GPS-Systemen möglich sind.

RQ4b: *Kann eine datengetriebene Klassifizierung Momente in der menschlichen Bewegung finden, um die Ausrichtung verschiedener Körperteile zu bestimmen?*

Kapitel 6 zeigt, dass das SVM-Verfahren Momente in der Bewegung des Kopfes erfassen kann (F_1 -Punktzahl = 95 %), die die genaue Beziehung zwischen Kopf und Oberkörper bestimmen können. Ein weiteres Experiment hat gezeigt, dass eine BLSTM-Methode (F_1 -Punktzahl = 98.4 %) Gangphasen (bekannte Momente in einem Gangzyklus) eines Menschen direkt aus den Daten eines am Kopf getragenen Sensors vorhersagen kann (durchschnittlich 140 ms) [130]. Es somit möglich, dass datengetriebene Klassifizierer Momente in der menschlichen Bewegung finden, um die Ausrichtung verschiedener Körperteile zu bestimmen. Es ist sogar mögliche zukünftige Momente vorherzusagen [130].

8.3.5 Forschungsfragen zur Fusion

RQ5: *Können datengetriebene Verfahren die Fusion verschiedener Sensordatenströme (MEMS-Trägheitssensor und Funksystem) verbessern?*

Diese Forschungsfrage wird von T. Feigl in bereits veröffentlichten Arbeiten [133, 130, 323, 129] behandelt. Im Rahmen dieser Arbeiten wurde ein Patent eingereicht [120].

Zahlreiche Experimente in Kapitel 7 zeigen, dass die Fusion von Datenströmen von verschiedenen Sensoren mit datengetriebenen Methoden zu genaueren Posen und rekonstruierten Trajektorien führt als herkömmliche Methoden, wie sequentielle PDR und KF. Datengetriebene Methoden bieten die Möglichkeit, weit in die Zukunft zu extrapoliieren, bei plötzlichen Bewegungsänderungen so schnell wie möglich zur richtigen Trajektorie zurückzukehren und Ausfälle am genauesten zu kompensieren.

Die Ergebnisse der Experimente in Kapitel 7 zeigen daher, dass datengetriebene Methoden klare Vorteile, bspw. höhere Genauigkeit, gegenüber herkömmlichen Methoden haben, wenn sie verschiedene Sen-

sordatenströme fusionieren, aber es gibt auch einige Dinge zu beachten die in den folgenden Fragen diskutiert werden.

RQ5a: *Wie genau können datengetriebene Verfahren zur Fusion verschiedener Sensordatenströme die Pose eines Fußgängers in dynamischen Situationen schätzen?*

Kapitel 7 zeigt, dass bestimmte Kombinationen von Eingabeveriablen zu wesentlich genaueren Posen führen als andere. Die Kombination der Funkposition mit der Geschwindigkeit führte zu den genauesten Posen (FRFNN: MAE = 0.0375, MSE = 0.0008 [m]). Das Hinzufügen der verrauschten Beschleunigung führte zu den schlechtesten Ergebnissen (FRFNN: MAE=0.2722, MSE=0.0248 [m]). Im Allgemeinen lieferte die FRFNN-Methode in allen Experimenten immer die genauesten Posen und zeigte das kürzeste Einschwingverhalten und die kleinsten Fehler bei plötzlichen Bewegungsänderungen. Tests an dynamischen Daten mit abrupten und schnellen Bewegungsänderungen (z. B. Spiralen) haben gezeigt, dass FRFNN sehr genau ist, wenn es mit Bruchteilen der komplexen Bewegungsformen (gerade Linien und Kurven) und einem größeren Geschwindigkeitsintervall (FRFNN: MSE=0.00024 [m] gegenüber KF: MSE=0.65413 [m]) trainiert wurde.

Die Ergebnisse aus Abschnitt 7.8 zeigen, dass der vorgeschlagene datengetriebene Ansatz zur Fusion verschiedener Sensordatenströme die Pose eines Fußgängers in dynamischen Situationen deutlich genauer und zuverlässiger schätzen kann als bisherige (optimierte) Verfahren.

RQ5b: *Wie weit voraus können datengetriebene Verfahren zur Fusion verschiedener Sensordatenströme die Pose eines Fußgängers in dynamischen Situationen vorhersagen?*

Kapitel 7 zeigt, dass die Vorhersagegenauigkeit von FRFNN am höchsten ist und Posen, die in der Zukunft weit entfernt sind (1000 ms), genau vorhergesagt werden können (FRFNN: MAE=0.0489, MSE=0.0016 [m]). Erste Ergebnisse zeigen, dass die Breite des Prognosehorizonts nicht unbedingt mit der Sequenzlänge korreliert, sondern mit der Komplexität der Bewegungsdynamik. Je variabler die Dynamik der Daten ist, desto kürzer ist der Horizont mit zunehmender Sequenzlänge. Da die Daten sehr dynamische Bewegungen zeigen, kann davon ausgegangen werden, dass der Prognosehorizont für statische Bewegungen erheblich breiter ist. Erste Ergebnisse haben gezeigt, dass

einfache Kreisbewegungen (10 s) mit langen Eingabesequenzen (30 s) fast vollständig und sehr genau vorhergesagt werden können.

Die Ergebnisse aus Abschnitt 7.8 zeigen, dass der vorgeschlagene datengetriebene Ansatz zur Fusion verschiedener Sensordatenströme die Pose eines Fußgängers in dynamischen Situationen bis zu 2 s ohne signifikanten Genauigkeitsverlust vorhersagen kann.

RQ5c: Welche Vorteile bieten datengetriebene Verfahren zur Fusion verschiedener Sensordatenströme eines Fußgängers in dynamischen Situationen?

Kapitel 7 zeigt, dass die datengetriebene Methode FRFNN Posen in allen Experimenten genauer und zuverlässiger schätzt (unbekannte Testpersonen mit zufälliger Bewegung: $MAE=0.0375$, $MSE=0.0008$ [m]) als die sequentielle Methode mit Hybrid als Basis und einem naiven Kalibrierungsmechanismus (unbekannte Testpersonen mit zufälliger Bewegung: $MAE=0.49$, $MSE=0.52$ [m]). Im Gegensatz zu FRFNN hängt die sequentielle Methode jedoch nicht von der Verfügbarkeit der Positionen ab und kann trotzdem genaue Trajektorien rekonstruieren ($MAE=0.92$, $MSE=1.37$ [m]). In Bezug auf dynamische Bewegungen sind beide Methoden geeignet, da beide auf der Geschwindigkeitsschätzung von Hybrid basieren. Es hängt von der Anwendung ab: Wenn keine hohe absolute Genauigkeit erforderlich ist, eine hohe Genauigkeit für relative Änderungen genügt, kann auch eine einzige Ausgangsposition ausreichen, um eine stabile Pose über einen längeren Zeitraum mit der sequentiellen Methode zu schätzen. Allerdings, wenn die absolute Position sehr genau sein muss, z. B. um Kollisionen zu vermeiden, sollte FRFNN ausgewählt und sichergestellt werden, dass mindestens zwei Positionen pro Sekunde verfügbar sind.

Die Ergebnisse aus Abschnitt 7.8 zeigen, dass der vorgeschlagene datengetriebene Ansatz zur Fusion verschiedene Vorteile bietet (höhere Genauigkeit, schnelleres Einschwingverhalten, Vorhersagbarkeit, optimale Gewichtung, Robustheit gegenüber Datenlücken und implizite Sensorfusion) als herkömmliche Verfahren des Standes der Technik. Diese Vorteile kosten jedoch viele Daten.

RQ5d: Gibt es einen Einfluss verschiedener Kombinationen von Eingabevervariablen, z. B. Position, Geschwindigkeit und Beschleunigung, auf die Schätzgenauigkeit von rekurrenten datengetriebene Modellen?

Kapitel 7 zeigt, dass die Eingabeveriablen einen starken Einfluss auf die Posengenauigkeit in datengetriebenen Verfahren haben. Das Hinzufügen der Beschleunigung und Orientierung zur Position führte zu deutlich schlechteren Ergebnissen (FRFNN: MAE=0.2722, MSE=0.0248 [m]). Das Hinzufügen von Geschwindigkeit zur Position ergab jedoch die besten Ergebnisse (FRFNN: MAE=0.0375, MSE=0.0008 [m]). Es wurde auch gezeigt, dass es ratsam ist, die Ausrichtung explizit in die Eingabesequenz aufzunehmen, wenn die Ausrichtung für ein anderes Rotationssystem als die Position bestimmt wird, z. B. Ausrichtung des Kopfes und Position des Rumpfes. Es wurde auch gezeigt, dass die Sequenzlänge die Speicherkapazität von LSTM-Zellen möglicherweise beeinflusst, sodass längere Eingabesequenzen zu einer Abnahme der Genauigkeit führen. 128 Werte bei FRFNN lieferten die höchste Genauigkeit (FRFNN: $CEP_{95} = 0.0991$ [m]). Durch die Auswahl der richtigen Sequenzlänge basierend auf dem Informationsgehalt der Daten in jedem Zeitschritt können datengetriebene Methoden auf alle komplexen Bewegungsformen verallgemeinert werden. Eine höhere Dimension bedeutet höheren Informationsgehalt, bspw. Position, Geschwindigkeit, Beschleunigung und Orientierung pro Zeitschritt und erhöht die Komplexität pro Zeitschritt und kann zu Chaos führen. FRFNN liefert die genauesten Posen, wenn die Sequenzlänge so gewählt wird, dass sie gerade Linien und Kurven darstellt, um auf dieser Grundlage komplexe zufällige Bewegungsformen wie Spiralen usw. darzustellen (MAE=0.0417, MSE=0.0041 [m]).

Die Experimente in Abschnitt 7.8 zeigen, dass datengetriebene Verfahren die Rekonstruktion von Pose und Trajektorien signifikant verbessern (unbekannte Testpersonen mit zufälliger Bewegung: FRFNN: MAE=0.0375, Sequentiell-Hybrid rekalibriert: MAE=0.49 [m]). Die einheitliche Skalierbarkeit (z. B. maximale Entfernung oder maximale Geschwindigkeit) aller Daten, einschließlich unbekannter Daten, muss jedoch möglich sein, damit die unbekannten Daten mit den im Modell gelernten Daten übereinstimmen. Die Robustheit des datengetriebenen Modells ist auch im Falle eines Fehlers im Datenstrom begrenzt, da exogene Informationen wie mindestens zwei aktuelle Positionen oder eine Entfernung in der Eingabesequenz vorhanden sein müssen, um eine plausible Vorhersage der Pose zu treffen.

8.3.6 Methodische Forschungsfragen

RQ6: Welche Vorteile bieten datengetriebene Verfahren zur Verarbeitung von verrauschten Zeitreihendaten?

Datengetriebene Verfahren bringen eine Reihe von Vorteilen. Die korrekte Konfiguration eines modellgetriebenen Ansatzes ist selbst bei einer perfekten synthetischen Datengrundlage mühsam. Kapitel 4 zeigt, dass es für modellgetriebene Ansätze mit natürlichen Daten unmöglich ist, Messprozesse und Übergangsfunktionen zu finden, die für alle möglichen unbekannten Daten funktionieren. Selbst wenn die Funktionen anhand eines bekannten Datensatzes optimiert werden, führt das vordefinierte Bewegungsmodell dazu, dass die Wahrheit überschritten oder unterschritten wird, da die Modelle entweder zu starr sind, um das Rauschen zu kompensieren, oder zu dynamisch, um der Bewegung zu folgen. Daher sind die modellgetriebenen Verfahren häufig sehr allgemein konzipiert, um so viele Situationen wie möglich zu bewältigen, und führen häufig zu größeren Ungenauigkeiten. Datengetriebene Verfahren hingegen lernen direkt aus den Daten und sind daher so allgemein oder spezifisch wie Ihre Daten. Eine zeitaufwändige Parametrisierung ist nicht erforderlich, obwohl die (optionale) Möglichkeit der Modellparameteroptimierung wesentlich umfangreicher ist als die eines Bayes-Filters. Ein weiterer Vorteil ist das automatische Entfernen oder Umgehen von Rauschen in den Daten. Die tiefen Modelle in den Experimenten waren alle in der Lage, zusätzlich zum Rauschen zusätzliche Informationen zu extrahieren, um sinnvolle Abbildungsfunktionen zu lernen. Herkömmliche Verfahren verwenden dagegen umständliche Signalverarbeitungsketten, wie sie zum Vorfiltern in Kapitel 6 verwendet werden. Ein Vorteil ist es, vorhandene Modelle an eine neue Situation anzupassen. Vorab trainierte Modelle können wiederverwendet werden, indem Teile der Architektur durch nicht geschulte Komponenten ersetzt und für die neue Situation mit den entsprechenden Daten (nach)trainiert werden.

Die Ergebnisse dieser Arbeit zeigen, dass es ratsam ist, datengetriebene Methoden zur Verarbeitung datenbasierter Probleme zu verwenden. Die nichtlinearen Funktionen, die mit der Tiefe und Breite der datengetriebenen Modelle einhergehen, bieten unerwartete Analyse- und Interpretationseigenschaften in hochdimensionalen Daten mit nichtlinearen Eigenschaften. Bspw. war es möglich, Gangparameter [129] und

Symptome der Simulatorkrankheit [125] zu korrelieren, die mit herkömmlichen linearen Operationen nicht gefunden werden konnten.

RQ6a: *Kann die Extraktion räumlicher Merkmale mithilfe von faltenden neuronalen Netzen und die Verfolgung dieser Merkmale mithilfe rekurrenter Netze die Genauigkeit und Robustheit der Schätzungen verbessern?*

Kapitel 5 zeigt, dass die Verwendung von Schichten mit Faltungskernen vor Schichten mit rekurrenten Zellen signifikant genauere Ergebnisse erzielt als Schichten, die nur aus Faltungskernen oder Schichten mit Faltungskernen und Sprungverbindungen bestehen (Datensatz ausgelassene Probanden mit zufälliger Bewegung, MAE=RoNIN = 0.83, C/RNN = 0.41 [m]). Weitere Experimente in Kapitel 5 haben gezeigt, dass Sprungverbindungen in Resteinheiten tiefere und komplexere räumliche Merkmale lernen und dass nachfolgende rekurrente Zellen noch genauere Geschwindigkeiten vorhersagen können (Datensatz ausgelassene Probanden mit zufälliger Bewegung, MAE=C/RNN = 0.41, Hybrid = 0.21 [m]). Dies weist darauf hin, dass es abhängig von der Komplexität der Informationen in den Eingabesequenzen (z. B. hochdimensionale Daten wie Rohbeschleunigungen oder niedrigdimensionale Daten wie die Position) wichtig ist, wie tief ein Netz ist, um notwendige Merkmalsextraktoren oder Dimensionsreduzierungen bereitzustellen, um die Kapazität der rekurrenten Schichten nicht zu überlasten. Kapitel 7 zeigt, dass eine einfache dichte Schicht vor der rekurrenten Schicht bereits veredelte Informationen (Positionen und Geschwindigkeiten) optimal in eine geeignete Dimension umwandeln kann, um optimale Posen mit nachgeschalteten LSTM-Zellen vorherzusagen (FRFNN: MAE=0.0375, MSE=0.0008 [m]). Im Gegensatz dazu konnte diese Architektur die zusätzliche Rohbeschleunigung in der Eingabesequenz nicht angemessen umwandeln, sodass die LSTM-Zellen keine genaue Pose schätzen konnte (FRFNN: MAE=0.2722, MSE=0.0248 [m]). Diese Tatsache zeigt an, dass die zusätzliche Beschleunigung zu komplex war, sodass die einfache Dichteschicht nicht ausreichte, um die Abmessungen zu verringern. Eine (Vorab)Extraktion räumlicher Merkmale ist daher ratsam.

Die Ergebnisse zeigen, dass die Extraktion von Merkmalen oder die Reduzierung oder Transformation der Eingabesequenz in eine Dimension, die für nachfolgende zeitliche Merkmalsverfolger tolerierbar ist, zu signifikant besseren Vorhersagen und Rekonstruktionsge-

nauigkeiten führt. Es ist daher ratsam, eine anwendungsspezifische Parameter- und Architekturoptimierung durchzuführen, um eine optimale (Räumliche-Zeitliche-Merkmalesverfolger-)Architektur zu finden.

RQ6b: *Können rekurrente datengetriebene Modelle niederfrequente Positionsinformationen implizit mittels hochfrequenten Geschwindigkeitsinformationen interpolieren?*

Kapitel 7 zeigt, dass mehrdimensionale Eingabesequenzen so strukturiert sein müssen, dass sie für jeden Zeitschritt in einer Sequenz gültige Informationen enthalten. Für eine Eingabesequenz bspw. mit einer Länge von 100 stehen für jeden Zeitschritt 100 Positionen und Geschwindigkeiten zur Verfügung. Fehlen einzelne Datenpunkte, können diese entweder durch Interpolation ergänzt oder durch den Wert 0 ersetzt werden. Die Experimente in Kapitel 7 haben gezeigt, dass mindestens zwei verschiedene aktuelle Positionen mit einem Mindestabstand von 0.5 s erforderlich sind, um genaue Posen vorherzusagen. Sequenzen mit nur einer Position lieferten dagegen keine plausiblen Ergebnisse.

Dies legt nahe, dass datengetriebene Verfahren niederfrequente Daten (zwei Positionen pro Eingabesequenz) mit lückenlosen Geschwindigkeiten kompensieren können. Möglicherweise lernen diese Verfahren implizit eine Interpolationsfunktion, um fehlende Informationen aufzufüllen. Diese Aussage scheint gültig zu sein, da auch längere dynamische Sequenzen mit nur zwei Positionen problemlos rekonstruiert werden konnten. Ohne implizit erlernte Interpolationsfunktion wären die Ergebnisse erheblich schlechter, da selbst die genauesten datengetriebenen Geschwindigkeitsschätzer in dynamischen Situationen ungenaue Ergebnisse liefern, die zu ungenauen Trajektorien führen.

Die Ergebnisse in Abschnitt 7.2.1 deuten darauf hin, dass datengetriebene Modelle niederfrequente Positionsinformationen implizit mittels hochfrequenten Geschwindigkeitsinformationen interpolieren können, allerdings mit Einschränkungen.

RQ6c: *Kann ein nachgelagertes Bayes-Filter (Kalman) die Fehlervarianz eines datengetriebenen Modells reduzieren?*

Kapitel 5 zeigt, dass ein datengetriebenes Modell und ein nachgeschaltetes KF, die Geschwindigkeitsgenauigkeit erhöhen kann (V3 Testdatensatz: Hybrid ohne KF: MAE=0.17; Hybrid mit KF: MAE=0.16

[m]). Die Fehlervarianz des KFs korreliert jedoch mit der Fehlervarianz des Lieferprozesses, der die Eingabeveriablen (fehlerhaft) schätzt. In Experimenten wurde gezeigt, dass eine PDR-Methode mit einer hohen Fehlervarianz am meisten von einem nachgeschalteten KF profitiert (V₂ Testdatensatz: PDR ohne KF: MAE=1.93; PDR mit KF: MAE=0.99 [m]), während die datengetriebenen Methoden mit zunehmender Schätzgenauigkeit signifikant weniger profitierten (V₂ Testdatensatz: Hybrid ohne KF: MAE=0.22; Hybrid mit KF: MAE=0.19 [m]).

Die Ergebnisse in Abschnitt 5.7 zeigen, dass alle Verfahren von einem nachgeschalteten KF profitieren, da dieser die Fehlervarianz reduziert und die Geschwindigkeitsvektoren deutlich sanftere Trajektorien rekonstruieren. Dennoch, die Optimierung durch das KF kostet aufwendiges Expertenwissen, zusätzliche Trainingszeit und zusätzliche (Validierungs)Daten.

RQ6d: *Können rekurrente datengetriebene Methoden natürliches Rauschen von Trägheitssensoren in dynamischen Bewegungen entfernen, um Zielvariablen genau und zuverlässig zu schätzen?*

In einem naiven Experiment in Kapitel 7 wurde gezeigt, dass ein datengetriebenes Modell die gleichen Funktionen wie ein Bandpassfilter erlernen kann: In zwei unabhängigen Tests wurden einmal verrauschte Beschleunigungen und vorgefilterte Beschleunigungen, in zwei datengetriebenen Modellen auf Referenzgeschwindigkeiten abgebildet. Beide Modelle boten nahezu identische Genauigkeiten. Das Modell, das die verrauschten Eingabedaten abgebildet hat, liefert unwesentlich ($SD = 0.007 \text{ m/s}$) genauere Geschwindigkeiten. Dies deutet darauf hin, dass datengetriebene Verfahren implizit komplexe Filterfunktionen lernen können. Es ist unklar, wie tief die Architektur sein muss, um sowohl die Filterfunktion als auch andere Funktionen wie das Zusammenführen multimodaler Signale zu ermöglichen. Ein Hinweis auf die Komplexität wird in Kapitel 5 gegeben, in dem gezeigt wird, dass für die Abbildung von Rohbeschleunigungen auf die Geschwindigkeit niedrigerer Modelle höhere Genauigkeiten liefern. Während der Experimente in Kapitel 4 wurde festgestellt, dass flachere Modelle (mit weniger Schichten) das hochkomplexe Langzeitrauschen schlechter lernen und nicht in der Lage sind, das Rauschen von der Bewegung zu trennen und daher schlechtere Ergebnisse liefern als vergleichbare tiefere Modelle. Allerdings wurde in Kapitel 7 auch gezeigt, dass das Hinzufügen der Rohbeschleunigung zur Fusion zu

schlechten Posengenauigkeiten führt.

Die Ergebnisse zeigen, dass datengetriebene Modelle natürliches Rauschen entfernen oder kompensieren und gleichzeitig genaue Ergebnisse schätzen können. Allerdings dürfen die Eingabedaten eine bestimmte (unbekannte) Komplexität nicht überschreiten.

RQ6e: Können rekurrente datengetriebene Methoden die Fehlervarianz bei plötzlichen Bewegungsänderungen schneller reduzieren, um das Einschwingverhalten zu verbessern?

Die Ergebnisse in Kapitel 7 zeigen, dass die FRFNN-Methode präzise Posen und ein kurzes Einschwingverhalten (FRFNN = 0.4 und KF = 1.8 s) sowie die kleinsten relativen Fehler (FRFNN: MSE=0.00024; KF: MSE=0.65413 [m]) bei plötzlichen Bewegungsänderungen liefert. Tests an unbekannten dynamischen Daten mit abrupten und schnellen Bewegungsänderungen (z. B. Kreise, Spiralen) haben gezeigt, dass FRFNN sehr genau ist, wenn es mit Bruchteilen der komplexen Bewegungsformen (gerade Linien und Kurven) und einem größeren Geschwindigkeitsintervall (FRFNN) trainiert wird (MAE=0.0417, MSE=0.0041 [m]). Die Ergebnisse in Kapitel 4 zeigen, dass datengetriebene Methoden eine Zuordnungsfunktion direkt aus den Daten lernen können, selbst bei Bewegungen, die sich abrupt und plötzlich ändern. Die Experimente haben gezeigt, dass LSTMs die Ankunftszeiten, die Bewegungsformen mit schnellen Richtungsänderungen oder Beschleunigungs- und Bremsvorgängen darstellen, mit hoher Genauigkeit auf unbekannte Positionen abbilden können (Zickzack Datensatz: MAE=0.058, MSE=0.006 [m]) und Kurven deutlich weniger überschießt oder sich schneller einpendelt als herkömmliche Methoden. Während der Experimente in Kapitel 4 wurde auch festgestellt, dass flachere Modelle (mit weniger Schichten) bei Kurvenfahrten mit plötzlichen Richtungsänderungen signifikant ungenauer sind.

Die Ergebnisse zeigen, dass datengetriebene Modelle signifikant schneller (0.4 s) und zuverlässiger ($MSE < 0.0003 \text{ m}$) auf plötzliche Bewegungsänderungen reagieren als modellgetriebene Ansätze (1.8 s).

RQ6f: Können rekurrente datengetriebene Methoden Bewegungseigenschaften, z.B. Trajektorienformen, verallgemeinern, die nicht Teil des Trainingssatz waren?

In Kapitel 7 wurden eine sequentielle modellgetriebene und ein datengetriebenes Verfahren vorgestellt. Der modellgetriebene Ansatz kann Trajektorienformen gut rekonstruieren, wenn die angegebenen Positionen und Geschwindigkeiten genau sind (Datensatz ausgelassene Probanden, Hybrid: MAE=0.49, MSE=0.52 [m]). Es werden jedoch Abstandsfehler akkumuliert, die zu Sprüngen führen, wenn eine aktuelle Position für die Kalibrierung verwendet wird (Datensatz ausgelassene Probanden: Hybrid pur RMSE=1.12, Hybrid rekalibriert RMSE=0.66 [m]). Sequentielle Prozeduren (Seq.) verallgemeinern, wenn die gelieferten Daten korrekt sind. Im Gegensatz dazu liefern datengetriebene Methoden präzise Ergebnisse sowohl für die Geschwindigkeitsschätzung als auch für die Position, Orientierung und Pose, wenn die Trainingsdaten Informationen so verteilen, dass die Informationen in den Testdaten Teil der Trainingsdaten sind (zufällige Bewegung einer unbekannten Person: Seq. Hybrid: MAE=0.49, FRFNN: MAE=0.0375 [m]). Kapitel 5 zeigt, dass das datengetriebene Modell während des Trainings Bewegungen lernt, die sowohl die minimale als auch die maximale Geschwindigkeit abdecken. Wenn der Testdatensatz eine höhere Geschwindigkeit darstellt, liefert das Verfahren unplausibel Ergebnisse. Ebenso zeigt Kapitel 7, dass die Modelle alle kleinsten gängigen Bewegungsformen (Kurven und gerade Linien) gelernt haben müssen, um komplexe zufällige Bewegungen darzustellen. Dies ist jedoch nur möglich, wenn die Positionen und Geschwindigkeiten auch in den Informationsgrenzen der Trainingdaten liegen.

Die Ergebnisse zeigen, dass datengetriebene Methoden unbekannte Datenpunkte sehr gut interpolieren können, wenn die Trainingsdaten die Obermenge und die Grenzbereiche abstecken. Im Gegensatz dazu können die Modelle nicht extrapolieren, siehe Kapitel 5. Interessanterweise haben erste Experimente gezeigt, dass es oft ausreicht, die letzte dichte Schicht auf neue Daten zu trainieren, z. B., um höhere Geschwindigkeiten oder weiter entfernte Positionen in einer anderen Umgebung außerhalb der Trainingsdaten zu erlernen. Diese Experimente haben auch gezeigt, dass es ratsam ist, mit Datensätzen zu trainieren, die keine Fehler enthalten (keine Mehrwegkomponenten und keine Ausfälle in den Daten), um die ersten Schichten im Modell für die Dimensionsreduzierung optimal zu trainieren. Die Schichten zum Entfernen des Rauschens können dann mit schmutzigen Referenzen und verrauschten Daten nachbehandelt werden, um optimale Ergebnisse zu erzielen.

8.4 Anwendungen in der virtuellen Realität

In diesem Abschnitt werden Anwendungen aufgeführt, die durch die Methoden dieser Dissertation ermöglicht wurden. Die Gangrekonstruktion in VR (Abschnitt 8.4.1), die Ermittlung der Simulatorkrankheit in VR (Abschnitt 8.4.2) und das weltgrößte VR Museum (Abschnitt 8.4.3). Die Hintergründe der Anwendungen werden vorgestellt und der Zusammenhang mit der Dissertation sowie der Beitrag und Einsatz der Methoden diskutiert. Da Anwendungen in Robotik und Sport bereits in den Kapiteln 4 und 5 vorgestellt wurden, werden im Folgenden Anwendungen der virtuellen Realität beschrieben.

8.4.1 Gangrekonstruktion

Diese Forschungsarbeit wurde bereits veröffentlicht [130, 129].

Hintergrund. Das Verkörpern von Benutzern durch Avatare, die auf Bewegungsverfolgung und Rekonstruktion basieren, ist eine ständige Herausforderung für VR-Anwendungsentwickler. Hochwertige VR-Systeme verwenden Systeme, die den gesamten menschlichen Körper erfassen oder inverse Kinematik, um die Bewegung der unteren Extremitäten zu rekonstruieren und die Animation des virtuellen Avatars steuern. Mobile Systeme für großflächige VR Erlebnisse mit vielen Benutzern beschränken sich auf die Bewegungserkennung die Sensoren von Head-Mounted Displays (HMDs) ermöglichen und können weder die untere Extremität noch den gesamten Körper animieren.

In dieser Forschungsarbeit wird ein Ansatz zur Rekonstruktion von Gangbewegungen mit einem einzigen am Kopf montierten Beschleunigungsmesser vorgeschlagen, siehe Abbildung 8.1. Es werden verschiedene modell- und datengetriebene Modelle entwickelt, um Kopfbewegungen auf entsprechende Gangphasen eines Referenzsystems abzubilden. Um die Beinbewegung zu rekonstruieren, sagen die Modelle Gangphasen voraus, um äquivalente synthetische Animationen auszulösen. Die Experimente zeigen, dass das bidirektionale Langzeit-Kurzzeitgedächtnis (BLSTM)-basierte Modell am genauesten zukünftige Gangphasen vorhersagen kann. Eine Anwenderstudie mit 21 Testpersonen untersuchte die Auswirkungen des Ansatzes auf die Simulatorkrankheit und zeigte signifikant weniger negative Auswir-

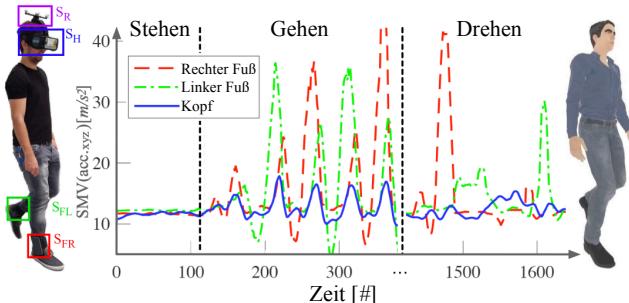


Abbildung 8.1: Gangrekonstruktion in VR: BLSTM prognostiziert Gangphasen (blaues Signal) mit einem Beschleunigungssensor (SH) zur Echtzeit-Gangrekonstruktion in VR (rechts im Bild).

kungen auf die Desorientierung.

Zusammenhang. Der Ansatz aus Kapitel 6 wurde zur Erstkalibrierung und Langzeitstabilisierung der Orientierung verwendet. Das in Kapitel 5 vorgeschlagene CNN-BLSTM-Modell wurde in dieser Forschungsarbeit angepasst, um die Bewegung des Menschen vorherzusagen, nämlich den Gangzyklus und die Gangphasen. Sensordaten von einem am Kopf montierten Trägheitssensor wurden verwendet, um einen virtuellen Avatar in VR in Echtzeit zu visualisieren. Die Erkenntnisse der Dissertation konnten genutzt werden, um zu zeigen, dass ein BLSTM-Modell signifikant geringere Latenzen aufweist als der Stand der Technik, da es Gangphasen früher erkennen bzw. zukünftige genau vorhersagen kann. Im Durchschnitt sagt BLSTM eine Gangphase mit einem F1-Score = 98% um 140.0 ms in der Zukunft voraus; min = 280.0 ms; max = 0.0 ms; SD = 30.0 ms.

8.4.2 Simulatorkrankheit

Diese Forschungsarbeit wurde bereits veröffentlicht [129, 125].

Hintergrund. In dieser Forschungsarbeit werden Bewegungsparameter, insbesondere Gangparameter, als objektive Indikatoren zur Beurteilung der Simulatorkrankheit in VR untersucht. Es werden zwei verschiedene Schätzmethoden für die Kamerapose (Position und Orientierung) zur Bewertung von Bewegungsaufgaben in einer

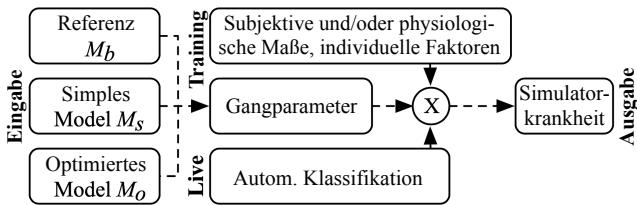


Abbildung 8.2: Verarbeitungskette zur Klassifikation von Simulatorkrankheit.

großen VR-Umgebung verwendet: Ein einfache und eine optimierte Methode. Die optimierte Methode ermöglicht eine genauere und natürlichere Abbildung der menschlichen Sinne. Die Teilnehmer führten mehrere Bewegungsaufgaben (Gehen, Balancieren, Laufen) unter drei Bedingungen aus: Einer Grundbedingung der physischen Realität, einer VR-Bedingung mit dem einfachen Modell und einer VR-Bedingung mit dem optimierten Modell. Diese Zustände werden im Hinblick auf die daraus resultierende Krankheit und Gangart sowie die wahrgenommene Präsenz in den VR-Zuständen verglichen.

Die Ergebnisse zeigen, dass das optimierte Posenschätzungsmodell die Simulatorkrankheit reduziert und die wahrgenommene Präsenz in VR erhöht. Beide Modelle beeinflussen die Gangparameter und die Simulatorkrankheit, weshalb ein Klassifizierungsansatz (siehe Abbildung 8.2) untersucht wurde, der die nichtlinearen Korrelationsabhängigkeiten zwischen Gangparametern und Simulatorkrankheit analysiert. Der Klassifikator könnte verwendet werden, um die Simulatorkrankheit auf der Grundlage menschlicher Gangparameter zu bewerten und vorherzusagen.

Zusammenhang. Die einfache Methode basiert auf einem hochoptimierten KF, das deterministische Verzögerungen und Positionssprünge verursacht. Diese führen zur Simulatorkrankheit. Die langsame Aktualisierungsfrequenz der Positionen (um die VR-Ansicht darzustellen) führt zu einer variierenden Verzögerung von Bewegung zu Photon (MTP) von durchschnittlich 246.34 ms. ($N = 87$, min. 198.7 ms, max. 319.6 ms, SD 76.4 ms) und einem Präzisionsfehler von <0.026 m. Das verwendete KF leidet unter plötzlichen Bewegungsänderungen, die zu einem Überschwingen und Unterschwingen der geschätzten

Trajektorie führen. Das einfache Modell bewegt sich somit im Stillstand und macht sich im Stillstand bemerkbar. Daher ist die Assoziation zwischen realer und virtueller Bewegung falsch, was zur Simulatorkrankheit führt.

Im Gegensatz dazu verwendet die optimierte Methode Erkenntnisse aus der Positionsschätzung dieser Dissertation, um die Sprünge und Verzögerungen der Position zu reduzieren. Zu diesem Zweck werden Bewegungszustände zur Laufzeit mit einer kubischen Stützvektormaschine klassifiziert und falsche Schätzungen, bspw. Bewegungen im Stillstand, des KFs herausgefiltert. Die Orientierungsschätzung aus Kapitel 6 kalibriert die Orientierung des Benutzers initial und stabilisiert sie zur Laufzeit. Das Wissen über die Signalverarbeitung und Klassifizierung von Zeitreihendaten aus Kapitel 6 wurde angewendet, um gesammelte Trägheitssensormessungen in bekannte Gangparameter umzuwandeln und verschiedene Intensitätsniveaus der Simulatorkrankheit auf der Grundlage der Gangparameter mit einer kubischen SVM zur Laufzeit zu klassifizieren.

8.4.3 Museum

Diese Forschungsarbeit wurde bereits durch kooperative Forschungsaktivitäten veröffentlicht [285, 284, 363, 362].

Hintergrund. In dieser Forschungsarbeit wird ein neuartiger Ansatz zur Verbesserung des Sozialverhaltens in der virtuellen Realität vorgestellt. Drei visuelle Transformationen wurden für Verhaltensphänomene entwickelt, die für alltägliche soziale Interaktionen von entscheidender Bedeutung sind: Augenkontakt, gegenseitige Aufmerksamkeit und Gruppierung. Um den Ansatz zu bewerten, interagieren Benutzer in einem virtuellen Museum mithilfe einer groß angelegten Mehrbenutzer-Lokalisierungsumgebung. Die Ergebnisse zeigen, dass der Ansatz die soziale Präsenz in Mehrbenutzerumgebungen erheblich steigern kann. Darüber hinaus konnte gezeigt werden, dass das zur Posenschätzung verwendete Lokalisierungssystem es ermöglicht, dass VR-Erlebnisse für viele Benutzer auf großen Flächen gleichzeitig bereitgestellt werden können, ohne den Tragekomfort zu beeinträchtigen und ohne Simulatorstörungen zu verursachen.

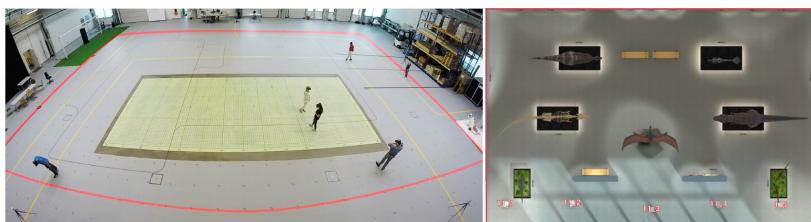


Abbildung 8.3: Reale und virtuelle Ansicht eines virtuellen Museums. (Die eingesetzte Posenschätzung basiert auf den Ergebnissen dieser Dissertation). [284]

Zusammenhang. Das optimierte Verfahren aus Kapitel 4 wurde verwendet, um die reale Kopfpose der Nutzer in der virtuellen Welt darzustellen. Die Kombination eines KFs mit einer datengetriebenen Klassifizierung ermöglicht Positionsschätzungen, die keine Simulatorerkrankung verursachen. In Kombination mit der Orientierungsschätzung aus Kapitel 6 können reale Posen auf virtuelle abgebildet werden. Die Posenschätzung ermöglicht einen langen Aufenthalt in einer virtuellen Erfahrung wie dem Museum, ohne Simulatorerkrankheit zu verursachen. Diese Anwendung wurde auch verwendet, um die Anwendbarkeit der Orientierungsschätzung aus Kapitel 6 zu bewerten.

8.5 Zukünftige Arbeiten

Dieser Abschnitt beschreibt die wesentlichen zukünftigen Arbeiten, die im Verlauf der Dissertation aufgekommen sind. In den Abschnitten zur Bewertung, Diskussion und Zusammenfassung der einzelnen Kapitel 4 bis 7 werden weitere zukünftige Arbeiten vorgeschlagen. In den Abschnitten 8.5.1 bis 8.5.4 werden zukünftige Methoden zur Analyse und Interpretation (Abschnitt 8.5.1), Optimierung der Verfahren (Abschnitt 8.5.2) und Generalisierbarkeit (Abschnitt 8.5.3 und 8.5.4) der datengetriebenen Methoden vorgeschlagen. Abschnitt 8.5.5 schlägt zukünftige Arbeiten zur weiteren Verbesserung der Lokalisierung vor.

Die Lokalisierung und Verfolgen von Fußgängern in Innenräumen wirft weiterhin unbeantwortete Fragen auf. Es gibt viele Möglichkeiten für zukünftige Arbeiten, die untersucht werden müssen, bevor Endverbrauchersysteme die in dieser Arbeit entwickelten Algorithmen verwenden können. Die Methoden dieser Arbeit können natürlich auch

auf andere Sensoren ausgedehnt werden, z. B. Kameras. Um zuverlässige, erklärbare und steuerbare neuronale Netze zu erhalten, sollten vier Hauptfaktoren berücksichtigt werden: die zuverlässige Bestimmung der Unsicherheit der datengetriebenen Methode, die optimale Modellarchitektur für die zugrunde liegende Anwendung, kontinuierliche Weiterbildung in neuen Umgebungen, bzw. die Übertragung von einer bekannten Domäne in eine neue, unbekannte Domäne und der Transfer der datengetriebene Algorithmen in praktische Anwendungen.

8.5.1 Unsicherheit und Erklärbarkeit

Eine besonders vielversprechende Richtung besteht darin, ein tiefes Wahrscheinlichkeitsmodell für die Zustandsschätzung zu erstellen, anstatt Zustände deterministisch aus Daten zu lernen. Ein Wahrscheinlichkeitsmodell kann die stochastische Messung und das Prozessrauschen eines Systems erfassen. Es bietet auch eine Unsicherheitsschätzung der Vorhersagen, die mit datengetriebenen Methoden getroffen wurden, um die katastrophalen Folgen von Fehlvorhersagen zu vermeiden. Erste Arbeiten in dieser Richtung wurden bereits im Rahmen der Dissertation eingeleitet. Die Mess-, Prozess- und Übergangskovarianzmatrizen eines KFs wurden durch LSTM-Zellen ersetzt, um Unsicherheiten eng in ein tiefes Modell zu integrieren, um eine Überanpassung zu verhindern, robustere Vorhersagen zu erzielen und eine effektive Fusion der multimodalen Sensorinformationen für die Erfassung des Systemzustands zu erreichen.

Mithilfe der probabilistischen Struktur des KF können Vorkenntnisse oder strukturelle Einschränkungen *a priori* als Verteilung formuliert und Fehlervarianzen der datengetriebenen Verfahren gesteuert werden. Die Hoffnung ist, dass dies die Menge an Trainingsdaten und die Konvergenzzeit reduziert und gleichzeitig die Generalisierungsfähigkeit in neuen Domänen verbessert. Darüber hinaus können mit ein wenig Bayes die einzelnen Komponenten, insbesondere die datengetriebenen Komponenten, in einer modularen Prozesskette besser interpretiert und ein mögliches falsches Verhalten direkt beeinflusst werden.

In den Methoden der Kapitel 4 bis 7 wurden Dropout-Mechanismen verwendet, um die Unsicherheit der einzelnen Komponenten zu bestimmen. In zukünftigen Arbeiten muss untersucht werden, wie Verwerfungsschichten während des Trainings besser kontrolliert werden

können, um den Verlauf des Trainings zu stabilisieren. Im Rahmen der Dissertation wurde eine Verarbeitungskette untersucht, um die internen Prozesse verschiedener datengetriebener Architekturen während des Trainings und während der Ausführungsphase zu visualisieren, zu interpretieren und besser zu verstehen [391, 183]. Das gewonnene Wissen kann dann zur Optimierung der Architektur genutzt werden.

8.5.2 Automatisiertes maschinelles Lernen

Die Forschung und Entwicklung von Verfahren der künstlichen Intelligenz schreitet derzeit sehr schnell voran. Es gibt eine Vielzahl neuer Methoden und Ideen, die möglicherweise auch auf die hier zugrunde liegenden Daten angewendet werden können. Die Literaturrecherche dieser Dissertation verdeutlicht, dass große Anstrengungen unternommen werden, um eine optimale Architektur für das zugrunde liegende Problem zu finden. Eine geringfügige Anpassung einzelner Parameter führt häufig dazu, dass das Training überhaupt nicht konvergiert. Es ist daher notwendig, Methoden zu erforschen, die den Gradienten permanent überwachen und die Parameter entsprechend anpassen.

Methoden des verstärkenden Lernens (engl. Reinforcement Learning, RL) könnten hier nützlich sein, um bspw. verschiedene Methoden mit unterschiedlichen Startkonfigurationen zu trainieren und das Training dauerhaft anzupassen, sodass sich aus optimalen Stapelverteilungen ein optimaler Trainingskurs und optimale Gradienten ergeben. Auf diese Weise könnte die Rechenzeit effizienter genutzt und das Problem behoben werden, dass derzeit viel Energie für die Erforschung von Modellen mit zu wenig Daten verschwendet wird, die morgen veraltet sein werden.

Im Rahmen künftiger Arbeiten sollte versucht werden, das Konzept der Resteinheiten in rekurrente Netze zu integrieren. Erste Ideen mit dichten RNNs haben bereits gezeigt, dass Sprungverbindungen in LSTMs für einige Datenzusammensetzungen bessere Ergebnisse erzielen als herkömmliche LSTM-Zellen. Beim Springen gehen jedoch Informationen verloren, die von Resteinheiten weiter unten im Netz weitergegeben werden könnten. Dies könnte auch das Problem des einzelnen begrenzenden Kontextvektors in LSTM-Zellen angehen. Da Resteinheiten als Kontextvektor betrachtet werden könnten. Diese Architektur

sollte dann mit Transformer Modellen nach dem Stand der Technik [99] verglichen werden, die bereits Resteinheiten verwenden.

8.5.3 Kontinuierliche Weiterbildung

Alle hier (Kapitel 4 bis 7) vorgeschlagenen datengetriebenen Verfahren wurden geschlossen und überwacht trainiert. Die Daten sind bereits vor dem Training verfügbar und repräsentieren das Wissen, das das Modell am besten lernen kann. Dieses Wissen ist jedoch oft zu begrenzt und zu spezifisch, sodass mehr Daten erforderlich sind. Wenn die hier vorgeschlagene Methode in der unkontrollierten realen Welt eingesetzt wird, ist sie mit komplexer Dynamik, unbekannten Objekten und sich ständig ändernden Umgebungen konfrontiert. Daher sollte sich die Methode kontinuierlich und schrittweise an neue Umgebungen anpassen können.

Datengetriebene Modelle könnten Selbstüberwachungsmechanismen verwenden, um Signale zu identifizieren, wie z. B. schwache Signale oder Geometrieeinschränkungen wie die maximal möglichen Signalausbreitungszeiten eines Funksystems, um die Modellparameter an die neue Umgebung anzupassen Darüber hinaus könnten Aufgaben wie Bewegungsplanung und Navigation verwendet werden, um bspw. neue Funkdaten in Bereichen einer unbekannten oder veränderten Funkumgebung, in denen ein Positionsschätzer hohe Unsicherheiten aufweist, kontinuierlich zu erfassen. Referenzpositionen eines zuverlässigen Sensors, bspw. eine stabile Geschwindigkeitsschätzung, eines LIDAR Sensors, könnten verwendet werden, um Unsicherheiten des Positionsschätzers in unbekannten Funkgebieten zu korrigieren.

8.5.4 Transfer Lernen

Eine weitere sehr wichtige Forschungsfrage, die sich im Rahmen der Dissertation stellte, ist die Frage der Generalisierbarkeit der Methode von einer Domäne zu einer anderen, unbekannten oder neuen Domäne. Erste Experimente haben gezeigt, dass ein vorab trainiertes Modell in einer Domäne mit wenigen Daten aus einer unbekannten Domäne umgeschult werden kann und sehr genaue Ergebnisse liefert.

Eine erste Studie verwendet vorab trainierte Modelle, die auf Daten aus einer bekannten Funkumgebung basieren, und kalibriert diese

Modelle durch Umschulung mit einer kleinen Datenmenge aus einer neuen, unbekannten Umgebung, sodass die Modelle auch in der unbekannten Umgebung optimale Ergebnisse liefern können. Natürlich kann es auch zu einer Übertragung zwischen anderen Bereichen kommen, z. B. unterschiedlichen Sensoren, Bewegungsanomalien, kleinen und großen Personen und unterschiedlichen Merkmalen. Hier gibt es jedoch auch viele methodische Fragen, da unklar ist, wie die Modelle optimal abgestimmt werden können. Welche Schichten umgeschult werden können und welche Gewichte fixiert werden müssen.

8.5.5 Lokalisierung

Ein zukünftiger Schwerpunkt ist die optimale Lokalisierung mittels Beamforming in der 5G Funklokalisierung. Ziel ist es, die Datenrate und die Energiekosten, die für die Bereitstellung anwendungsspezifisch optimierter Funkkanäle für Mobilfunkkunden erforderlich sind, optimal einzustellen. Da die neuen Systeme bidirektional zwischen Sender und Empfänger kommunizieren, hilft die Lokalisierungskomponente, die strahlbildenden Antennen über die Position des Endbenutzers zu informieren, um den Strahl gezielt auf die Position der Endbenutzer zurichten. Hier gibt es jedoch viele Forschungsfragen, da die Kommunikations- und Lokalisierungsverbindung auch durch Mehrwegeausbreitung gestört werden kann. Daher müssen Lösungen gefunden werden, wie die Mehrwegkomponenten identifiziert und verfolgt werden können.

Natürlich kann es immer noch reale Situationen geben, die der Hybridschwindigkeitsschätzer und der Orientierungsschätzer noch nicht abdecken. Mehr Schulungen zu synthetischen Daten und Umschulungen zu realen Daten können dazu beitragen, die Grenzen der Generalisierbarkeit zu erweitern. In Experimenten reichte bereits ein einfacher Simulator aus, um synthetische Daten zu generieren, die die Menge der benötigten realen Daten drastisch reduzieren. Zukünftige Arbeiten müssen untersuchen, wie der Simulator Bewegungseigenschaften erzeugen kann, die in realen Daten selten zu finden sind, z. B. Daten mit Bewegungsanomalien.

Da Signalausfälle in realen Umgebungen auch in sehr unregelmäßigen Abständen auftreten können, muss die Sequenzlänge drastisch erhöht werden. In zukünftigen Arbeiten muss untersucht werden,

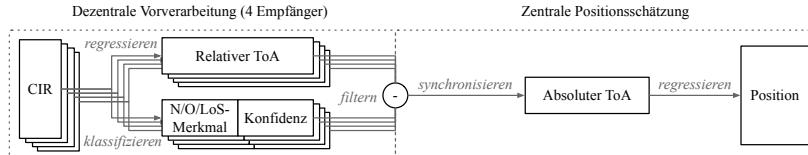


Abbildung 8.4: Zukünftige Verarbeitungskette der Funkpositionsschätzung.

wie LSTM und andere Modelle mit extrem langen Zeitabhängigkeiten umgehen und unregelmäßige und lang anhaltende Signalfehler identifizieren können. Andere aufmerksamkeitsbasierte Zeitreihen-Netzkonzepte wie Transformer (Bidirectional Encoder Representations from Transformers, BERT) sollten ebenfalls evaluiert werden [23, 443].

Erste Studien haben gezeigt, dass eine Anomalieerkennung zur Erkennung von NLoS-, OLoS- und LoS-Komponenten in Kanalimpulsantworten dazu beiträgt, falsche ToA-Werte zu identifizieren, um die Genauigkeit einer nachgeschalteten Positionsschätzung zu verbessern. Abbildung 8.4 zeigt die Verarbeitungskette, die relative ToA-Schätzung unterstützt und bewertet. Mehrwegkomponenten werden analysiert und schlechte ToA-Werte werden aussortiert und ein Optimierer nutzt saubere ToA Werte, um eine genauere Position zu schätzen. Die Positionsschätzung (Kapitel 4) kompensiert diese Probleme implizit, aber eine solche Vorfilterung der ToA-Werte könnte sich auch positiv auf diese Methode auswirken.

8.6 Letzte Worte

So wie wir Menschen durch Erfahrung geprägt sind und aus Erfahrung lernen, werden künstliche neuronale Netze durch die Art und Weise, wie sie lernen, geprägt. Ein globales Ziel in der KI-Forschung muss es daher sein, die Auswirkungen des Trainings auf die innere Struktur des Netzes im Detail zu untersuchen. Dies kann helfen, neuronale Netze besser zu verstehen und ein Äquivalent zur menschlichen Psychologie, d. h. Verhaltenstherapie, für neuronale Netze zu initiieren.

9 Eigene Veröffentlichungen, Patente und Qualifizierungsarbeiten

9.1 Konferenzbeiträge

Teile des Inhalts dieser Dissertation wurden bereits in Konferenzberichten veröffentlicht. Diese Beiträge finden sich in [130, 17, 323, 133, 285, 364, 365, 128, 127, 124, 363, 284, 126, 125, 122, 362].

9.2 Zeitschriftenartikel

Teile des Inhalts dieser Dissertation wurden bereits in Zeitschriftenartikeln veröffentlicht. Diese Beiträge finden sich in [134, 131, 129, 247, 323].

9.3 Patente

Teile dieser Dissertation wurden als Patente eingereicht oder angemeldet. Die Patentanmeldungen sind unter [308, 118, 119, 123, 121, 325, 120, 309, 10, 132] registriert.

9.4 Qualifizierungsarbeiten

Teile dieser Dissertation wurden von einigen meiner Studenten im Rahmen von Qualifizierungsarbeiten unterstützt und miterarbeitet. Ich möchte Thomas Altstidl [16], Jan Niklas Bauer [31], Peter Bauer [32], Frank Brendel [53], Christian Daxer [96], Felix Gruber [157], Lisa Gruner [159], Oskar Herrmann [183], Christian Jakob [207], Stephanie Mehlretter [296], Felix Ott [322], Lea Otte [324], Andreas Porada [341, 342], und Lukas Schmidt [391] danken, deren Beiträge die Entwicklung dieser Dissertation unterstützten.

Literaturverzeichnis

- [1] 3GPP Radio Access Network Working Group, et al. Study on channel model for frequencies from 0.5 to 100 GHz (release 16). Technical report, 3GPP TR 38.901, 2020.
- [2] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, Vladimir Makarenkov und Saeid Nahavandi. A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges, In arXiv:2011.06225 [cs.LG], 2021.
- [3] Chiraz Abdelkader, Ross Cutler und Larry Davis. Stride and cadence as a biometric in automatic person identification and verification. In *Proc. Intl. Conf. Autom. Face and Gesture Recogn.*, Seiten 372–377. Washington, DC, 2002.
- [4] Sundar Aditya, Andreas F. Mloisch und Hatim Mohammed Be'hairy. A survey on the impact of multipath on wideband time-of-arrival based localization. *Proc. Intl. IEEE*, 106(7):1183 – 1203, 2018.
- [5] Ehad Akeila, Zoran Salcic und Akshya Swain. Reducing low-cost INS error accumulation in distance estimation using self-resetting. *IEEE Trans. on Instrumentation and Measurement*, 63(1):177–184, 2013.
- [6] Auday Al-Mayyahi, Weiji Wang und Phil Birch. Levenberg-marquardt optimised Neural Networks for trajectory tracking of autonomous ground vehicles. *Intl. J. Mechatronics and Automation*, 5(2):140–153, 2015.
- [7] Klaithem Al Nuaimi und Hesham Kamel. A survey of indoor positioning systems and algorithms. In *Proc. Intl. Conf. Innovations in Information Techno.*, Seiten 185–190. Abu Dhabi, UAE, 2011.
- [8] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei und Silvio Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *Proc. Intl. Conf. Computer Vision and Pattern Recognition*, Seiten 961–971. Las Vegas, NV, 2016.

Literaturverzeichnis

- [9] Bardia Alavi und Kaveh Pahlavan. Modeling of the distance error for indoor geolocation. In *IEEE Proc. Intl. Conf. Wireless Commun. and Networking*, Seiten 668–672. Nantes, France, 2003.
- [10] Mohammad Alawieh, Ernst Eberlein, Tobias Feigl und Thomas v.d.Grün. Reporting Correlation Information, European Patent No. EP2019207844. Munich, Germany. European Patent Office, 2019.
- [11] Sheila Alemany, Jonathan Beltran, Adrián Pérez und Sam Ganzfried. Predicting hurricane trajectories using a Recurrent neural network. In *Proc. Intl. Conf. Artificial Intelligence*, Seiten 468–475. Honolulu, HI, 2019.
- [12] Robert McNeill Alexander. Stride length and speed for adults, children, and fossil hominids. *American J. of Physical Anthropology*, 63(1):23–27, 1984.
- [13] Eduard Allseits, Vibhor Agrawal, Jennifer Lučarević, Robert Gailey, Ignacio Gaunaurd und Christopher Bennett. A practical step length algorithm using lower limb angular velocities. *Biomechanics J.*, 66:137–144, 2018.
- [14] Zahangir Alom, Tarek Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C. Van Esen, Abdul A. S. Awwal und Vijayan K. Asari. The history began from AlexNet: A comprehensive survey on deep learning approaches, In arXiv:1803.01164 [CS], 2018.
- [15] Florent Altché und Arnaud de La Fortelle. An LSTM Network for highway trajectory prediction. In *Proc. Intl. Conf. Intelligent Transportation Systems (ITSC)*, Seiten 353–359. Maui, HI, 2017.
- [16] Thomas Altstidl. Application of deep learning methods to process natural phenomena. Bachelor Thesis, Friedrich-Alexander-University Erlangen-Nuremberg, 2019.
- [17] Anes Redzepagic, Christoffer Löffler, Tobias Feigl und Christopher Mutschler. A sense of quality for augmented reality assisted process guidance. In *Proc. Intl. Symp. on Mixed Reality and Augmented Reality (ISMAR)*, Seiten 1–6, 2020.
- [18] Lenkaran Arasaratnam und Simon Haykin. Cubature Kalman Filters. *IEEE Trans. on Automatic Control*, 54(6):1254–1269, 2009.

- [19] Lenkaran Arasaratnam, Simon Haykin und Thomas R Hurd. Cubature Kalman filtering for continuous-discrete systems: theory and simulations. *IEEE Trans. on Signal Processing*, 58(10):4977–4993, 2010.
- [20] Sanjeev M. Arulampalam, Simon Maskell, Neil Gordon und Tim Clapp. A tutorial on Particle Filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. on Signal Processing*, 50(2):174–188, 2002.
- [21] Christian Ascher, Lukasz Zwirello, Thomas Zwick und Gert Trommer. Integrity monitoring for UWB/INS tightly coupled pedestrian indoor scenarios. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation*, Seiten 1–6. Guimaraes, Portugal, 2011.
- [22] Matthias Bachtler, José Carrera und Thorsten Braun. Kalman Filter supported WiFi and PDR based indoor positioning system. *University of Bern: Bern, Switzerland*, 12(2):23–30, 2018.
- [23] Dzmitry Bahdanau, Kyunghyun Cho und Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proc. Intl. Conf. Learning Representations (ICLR)*, Seiten 57–64. Banff, Canada, 2015.
- [24] Siqi Bai, Mingjiang Yan, Qun Wan, Long He, Xinrui Wang und Junlin Li. DL-RNN: An accurate indoor localization method via double RNNs. *IEEE Sensors J.*, 18(6):1–1, 2019.
- [25] Jared B. Bancroft und Gérard Lachapelle. Data fusion algorithms for multiple inertial measurement units. *Sensors J.*, 11(7):6771–6798, 2011.
- [26] Stephen Bancroft. An algebraic solution of the GPS equations. *IEEE Trans. Aerospace and Electronic Systems*, 56(1):56–59, 1985.
- [27] Ling Bao und Stephen Intille. Activity recognition from user-annotated acceleration data. In *Proc. Intl. Conf. Pervasive Computing*, Seiten 1–17. Vienna, Austria, 2004.
- [28] István Barakonyi und Dieter Schmalstieg. Ubiquitous animated agents for augmented reality. In *Proc. Intl. Symp. Mixed and Augmented Reality*, Seiten 145–154. Santa Barbara, CA, 2006.

Literaturverzeichnis

- [29] Timothy D. Barfoot. *State estimation for Robotics*. Cambridge University Press, 2017.
- [30] Federico Bartoli, Giuseppe Lisanti, Lamberto Ballan und Alberto Del Bimbo. Context-aware trajectory prediction. In *Proc. Intl. Conf. Pattern Recognition (ICPR)*, Seiten 1941–1946. Beijing, China, 2018.
- [31] Jan Niklas Bauer. Evaluation of distributed neural networks for indoor radio positioning utilizing efficient embedded hardware. Master Thesis, Friedrich-Alexander-University Erlangen-Nuremberg, 2019.
- [32] Peter Bauer. Einfluss verschiedener Inertial- und Funksensordaten auf die Posenschätzung von Menschen mittels Rekurrenter Neuronaler Netze. Master Thesis, Technische Hochschule Nürnberg, 2020.
- [33] Stéphane Beauregard. A helmet-mounted pedestrian dead reckoning system. In *Proc. Intl. Forum Applied Wearable Computing*, Seiten 1–11. Bremen, Germany, 2006.
- [34] Stefan Becker, Ronny Hug, Wolfgang Hübner und Michael Arens. An evaluation of trajectory prediction approaches and notes on the TrajNet benchmark, In arXiv:1805.07663 [cs], 2018.
- [35] Abdelghani Belakbir, Mustapha Amghar und Nawal Sbiti. Sensor data fusion for an indoor and outdoor localization. *Radioelectronics and Communications Systems*, 57(4):149–158, 2014.
- [36] Yoshua Bengio, Patrice Y. Simard und Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *Trans. on Neural Networks*, 5(2):157–166, 1994.
- [37] Yoshua Bengio, Nicolas Boulanger-Lewandowski und Razvan Pascanu. Advances in optimizing Recurrent networks. In *Proc. Intl. Conf. Acoustics, Speech and Signal Processing*, Seiten 8624–8628. Vancouver, Canada, 2013.
- [38] Alessandro Benini, Adriano Mancini, Alessio Marinelli und Sauro Longhi. A biased extended Kalman Filter for indoor localization of a mobile agent using low-cost IMU and UWB wireless sensor network. *Intl. Federation of Automatic Control (IFAC)*, 45(22):735–740, 2012.

- [39] Alessandro Benini, Adriano Mancini und Sauro Longhi. An IMU/UWB/vision-based extended Kalman Filter for mini-uav localization in indoor environment using 802.15. 4a wireless sensor network. *J. of Intelligent and Robotic Systems*, 70(1-4):461–476, 2013.
- [40] Hamza Benzerrouk und Alexander Nebylov. Robust imu/uwb integration for indoor pedestrian navigation. In *Proc. Intl. Conf. Integrated Navigation Systems (ICINS)*, Seiten 1–5. Saint Petersburg, RUS, 2018.
- [41] Berta Bescos, José M. Fácil, Javier Civera und José Neira. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018.
- [42] Jamshed J. Bharucha und Peter M. Todd. Modeling the perception of tonal structure with Neural nets. *Computer Music J.*, 13(4):44–53, 1989.
- [43] Simone Bianco, Remi Cadene, Luigi Celona und Paolo Napoletano. Benchmark analysis of representative deep Neural Network architectures. *Access J.*, 6(9):64270–64277, 2018.
- [44] Jeff Bird und Dale Arden. Indoor navigation with foot-mounted strapdown inertial navigation and magnetic sensors. *IEEE Wireless Comm.*, 18(2):28–35, 2011.
- [45] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer Verlag, Berlin, Germany, 2006.
- [46] Michael Bloesch, Sammy Omari, Marco Hutter und Roland Siegwart. Robust visual inertial odometry using a direct EKF-based approach. In *Proc. Intl. Conf. Intelligent Robots and Systems (IROS)*, Seiten 298–304. Hamburg, Germany, 2015.
- [47] Li Bo, Qing-zhi Liu, Zhen-dong Yin und Zhi-lu Wu. A novel snr estimator for ds-uwb wireless sensor network. *IEEE Trans. on Computer Science and Engineering*, 17(3), 2017.
- [48] Dmytro Bobkov, Ferdinand Grimm, Eckehard Steinbach, Sebastian Hilsenbeck und Georg Schroth. Activity recognition on handheld devices for pedestrian indoor navigation. In *Proc. Intl. Conf.*

Literaturverzeichnis

Indoor Positioning and Indoor Navigation (IPIN), Seiten 1–10. Alberta, Canada, 2015.

- [49] Julian Bock, Till Beemelmanns, Markus Klösges und Jens Kotte. Self-learning trajectory prediction with Recurrent neural networks at intelligent intersections. In *Proc. Intl. Conf. Vehicle Technology and Intelligent Transport Systems*, Seiten 346–351. Science and Technology Publications, 2017.
- [50] Wolfgang Böge und Wilfried Pläßmann, editors. *Grundlagen und Grundbegriffe der Meßtechnik*, chapter 2, Seiten 735–740. Vieweg+Teubner, Wiesbaden, Germany, 2007.
- [51] Y-Lan Boureau, Jean Ponce und Yann LeCun. A Theoretical Analysis of Feature Pooling in Visual Recognition. In *Proc. Intl. Conf. Machine Learning*, Seiten 111–118. Haifa, Israel, 2010.
- [52] Juan Bravo, Edith Pulido Herrera und Daniel Alfonso Sierra. Comparison of step length and heading estimation methods for indoor environments. In *Proc. Intl. Conf. Electronics, Electrical Eng. and Comp.*, Seiten 12–19. Cusco, Peru, 2017.
- [53] Frank Brendel. Robuste Posenschätzung durch Identifikation von Kalibriermomenten mittels Machine Learning. Master Thesis, Technische Hochschule Nürnberg, 2018.
- [54] Martin Brossard, Axel Barrau und Silvere Bonnabel. RINS-w: Robust inertial navigation system on wheels, In arXiv:1903.02210 [CS], 2019.
- [55] Jason Brownlee. *Long Short-term Memory Networks with Python: Develop Sequence Prediction Models with Deep Learning*. Jason Brownlee, Vermont Victoria, AUD, 2017.
- [56] Nikhil Buduma und Nicholas Locascio. *Fundamentals of deep learning: designing next-generation machine intelligence algorithms*. O'Reilly Media, 1st edition, 2017. ISBN 978-1-4919-2561-4.
- [57] Mike Buerli und Stefan Misslinger. Introducing ARKit - Augmented Reality for iOS. In *Apple Worldwide Developers Conference (WWDC)*, Seiten 1–187. San Jose, CA, 2017.
- [58] Winfried Böhm. *Handbuch der Navigation: Begriffe, Formeln, Verfahren, Schemata*. Busse, 1978. ISBN 978-3-87120-323-7.

- [59] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid und John J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. on Robotics*, 32(6):1309–1332, 2016.
- [60] Adrián Canedo-Rodríguez, Víctor Alvarez-Santos, Carlos V. Regueiro, Roberto Iglesias, Senén Barro und J Presedo. Particle Filter robot localisation through robust fusion of laser, WiFi, compass, and a network of external cameras. *Information Fusion*, 27(1):170–188, 2016.
- [61] José Luis Carrera, Zhongliang Zhao, Thorsten Braun und Zan Li. A real-time indoor tracking system by fusing inertial sensor, radio Signal and floor plan. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–9. Barcelona, Spain, 2016.
- [62] Graeme Chambers, Svetha Venkatesh, Geoff West und Hung Hai Bui. Hierarchical recognition of intentional human gestures for sports video annotation. In *Proc. Intl. Conf. Pattern Recognition*, Seiten 1082–1085. Quebec City, Canada, 2002.
- [63] Michael B. Chang, Tomer Ullman, Antonio Torralba und Joshua B. Tenenbaum. A compositional object-based approach to learning physical dynamics, In arXiv:1612.00341v2 [cs.AI], 2016.
- [64] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall und W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Artificial Intelligence Research J.*, 16(3):321–357, 2002.
- [65] Badong Chen und José C Príncipe. Maximum correntropy estimation is a smoothed MAP estimation. *IEEE Signal Processing Letters*, 19(8):491–494, 2012.
- [66] Badong Chen, Lei Xing, Junli Liang, Nanning Zheng und Jose C Príncipe. Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion. *IEEE Signal Processing letters*, 21(7):880–884, 2014.
- [67] Badong Chen, Xi Liu, Haiquan Zhao und Jose C Príncipe. Maximum correntropy Kalman Filter. *Automatica*, 76(1):70–77, 2017.

- [68] Changhao Chen, Xiaoxuan Lu, Andrew Markham und Niki Trigoni. IONet: Learning to cure the curse of drift in inertial odometry, In arXiv:1802.02209 [CS], 2018.
- [69] Changhao Chen, Peijun Zhao, Chris Xiaoxuan Lu, Wei Wang, Andrew Markham und Niki Trigoni. Oxiod: The dataset for deep inertial odometry. *Computing Research Repository (CoRR)*, 13(1):1281–1293, 2018.
- [70] Changhao Chen, Xiaoxuan Lu, Johan Wahlstrom, Andrew Markham und Niki Trigoni. Deep neural network based inertial odometry using low-cost inertial measurement units. *Trans. on Mobile Computing*, 23(6):18–24, 2019.
- [71] Chien-Sheng Chen. Artificial Neural Network for location estimation in wireless communication systems. *Sensors J.*, 12(3):2798–2817, 2012.
- [72] Pengzhan Chen, Ye Kuang und Xiaoyue Chen. A uwb/improved PDR integration algorithm applied to dynamic indoor positioning for pedestrians. *Sensors J.*, 17(9):2065, 2017.
- [73] Zhenghua Chen, Han Zou, Hao Jiang, Qingchang Zhu, Yeng Chai Soh und Lihua Xie. Fusion of WiFi, smartphone sensors and landmarks using the Kalman Filter for indoor localization. *Sensors J.*, 18(9):715–732, 2015.
- [74] Long Cheng, Cheng-Dong Wu und Yun-Zhou Zhang. Indoor robot localization based on wireless sensor networks. *IEEE Trans. on Consumer Electronics*, 57(3):1099–1104, 2011.
- [75] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk und Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proc. Intl. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Seiten 1724–1734. Doha, Qatar, 2014.
- [76] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk und Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, In arXiv:1406.1078 [CS], 2014.

- [77] Seong Yun Cho und Chan Gook Park. Mems based pedestrian navigation system. *Navigation J.*, 59(1):135–153, 2006.
- [78] Dooseop Choi, Taeg-Hyun An, Kyounghwan Ahn und Jeongdan Choi. Future trajectory prediction via RNN and maximum margin inverse reinforcement learning. In *Proc. Intl. Conf. Machine Learning and Applications (ICMLA)*, Seiten 125–130. Orlando, FL, 2018.
- [79] Hone-Jay Chu, Guang-Je Tsai, Kai-Wei Chiang und Thanh-Trung Duong. GPS/MEMS INS data fusion and map matching in urban areas. *Sensors J.*, 13(9):11280–11288, 2013.
- [80] Alexander Chugunov, Elena Zakharova, Ana Mitic, Valery Semenov, Andrey Boldyrev, Dmitry Tsaregorodtsev und Nikita Petukhov. Integration of local ultrawideband toa/aoa phase difference of arrival system and inertial navigation systems. In *Proc. Intl. Conf. Integrated Navigation Systems (ICINS)*, Seiten 1–8. Saint Petersburg, RUS, 2020.
- [81] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho und Yoshua Bengio. Empirical evaluation of Gated Recurrent neural networks on sequence modeling, In arXiv:1412.3555 [cs.NE], 2014.
- [82] Marc Ciurana, Francisco Barceló und Sebastiano Cugno. Indoor tracking in WLAN location with TOA measurements. In *Proc. Intl. Workshop on Mobility Management and Wireless Access*, Seiten 121–125. Terromolinos, Spain, 2006.
- [83] Jürgen Cleve und Uwe Lämmel. *Data mining*. De Gruyter Oldenbourg, 2014. ISBN 978-3-486-71391-6.
- [84] Christophe Combettes und Valerie Renaudin. Comparison of misalignment estimation techniques between handheld device and walking directions. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 293–304. Banff, Canada, 2015.
- [85] Juan Antonio Corrales, Francisco A. Candelas und Fernando Torres. Hybrid tracking of human operators using IMU/UWB data fusion by a Kalman filter. In *Proc. Intl. Conf. Human-Robot Interaction (HRI)*, Seiten 193–200. Amsterdam, NL, 2008.

Literaturverzeichnis

- [86] Juan Antonio Corrales, Francisco A. Candelas und F Torres. Sensor data integration for indoor human tracking. *Robotics and Autonomous Systems*, 58(8):931–939, 2010.
- [87] Corinna Cortes und Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [88] Huseyin Coskun, Felix Achilles, Robert DiPietro, Nassir Navab und Federico Tombari. Long short-term memory Kalman Filters: Recurrent Neural estimators for pose regularization, In arXiv:1708.01885 [CS], 2017.
- [89] Thomas Cover und Peter Hart. Nearest neighbor pattern classification. *IEEE Trans. Information Theory*, 13(1):21–27, 1967.
- [90] Igor Cvišić, Josip Ćesić, Ivan Marković und Ivan Petrović. Soft-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles. *J. of field Robotics*, 35(4):578–595, 2018.
- [91] Huan Dai, Hong-Bo Liu, Xiao-Shuang Xing und Yong Jin. Indoor positioning algorithm based on parallel multilayer neural network. In *Proc. Intl. Conf. Information System and Artificial Intelligence*, Seiten 356–360, Hong Kong, China, 2016.
- [92] Jing Dai, Pinjia Zhang, Joy Mazumdar, Ronald G. Harley und Ganesh Venayagamoorthy. A comparison of mlp, rnn and esn in determining harmonic contributions from nonlinear loads. In *Proc. Intl. Conf. Industrial Electronics Society*, Seiten 3025–3032. Orlando, FL, 2008.
- [93] Davide Dardari, Chia-Chin Chong und Moe Win. Threshold-based time-of-arrival estimators in UWB dense multipath channels. *IEEE Trans. on Communic.*, 56(8):1366–1378, 2008.
- [94] Fred Daum. Nonlinear Filters: beyond the Kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 20(8):57–69, 2005.
- [95] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton und Oliver Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

- [96] Christian Daxer. Robustes Inside-Out Tracking für großflächige Mehrnutzer VR Systeme. Master Thesis, Technische Hochschule Nürnberg, 2018.
- [97] Alessio De Angelis, John Nilsson, Isaac Skog, Peter Händel und Paolo Carbone. Indoor positioning by ultrawide band radio aided inertial navigation. *Metrology and Measurement Systems*, 17(3): 67–78, 2010.
- [98] Zhi-An Deng, Ying Hu, Jianguo Yu und Zhenyu Na. Extended Kalman Filter for real time indoor localization by fusing WiFi and smartphone inertial sensors. *Micromachines*, 6(4):523–543, 2015.
- [99] Jacob Devlin, Ming-Wei Chang, Kenton Lee und Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, In arXiv:1810.04805 [cs.CL], 2019.
- [100] Vitomir Djaja-Josko und Marcin Kolakowski. A new map based method for nlos mitigation in the UWB indoor localization system. In *Proc. Intl. Conf. Telecommunication Forum (TELFOR)*, Seiten 1–4. Belgrade, Serbia, 2017.
- [101] Thai Minh Do und Thong Chi Le. Performance analysis of FFT Filter to measure displacement Signal in road roughness profiler. *Intl. J. Electrical Eng.*, 5(4):356–361, 2013.
- [102] Tri-Nhut Do, Ran Liu, Chau Yuen und U-Xuan Tan. Personal dead reckoning using IMU device at upper torso for walking and running. In *Proc. Intl. Conf. Sensors*, Seiten 1–3. Orlando, FL, 2016.
- [103] Giuseppe Donato, Vitor Sequeira und Abdul Sadka. Stereoscopic helmet mounted system for real time 3D environment reconstruction and indoor ego-motion estimation. In *Proc. Intl. Symp. on Defense and Security (SPIE)*, Seiten 1–12. Orlando, FL, 2008.
- [104] Liang Dong und Frank L. Severance. Position estimation with moving beacons in wireless sensor networks. In *Proc. Intl. Conf. Wireless Communications and Networking*, Seiten 2317–2321. Hong Kong, China, 2007.
- [105] Luis Enrique Díez, Alfonso Bahillo, Jon Otegui und Timothy Ottim. Step length estimation methods based on inertial sensors: A review. *Sensors J.*, 18(7):908–926, 2018.

Literaturverzeichnis

- [106] Marcus Edel und Enrico Köppe. An advanced method for pedestrian dead reckoning using BLSTM-RNNs. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–6. Alberta, Canada, 2015.
- [107] Naser El-Sheimy, Haiying Hou und Xiaoji Niu. Analysis and modeling of inertial sensors using Allan variance. *Trans. on Instrumentation and Measurement*, 57(1):140–149, 2008.
- [108] Mohammed Elbes, Ahmad Alkhatib, Ala Al-Fuqaha und Junaid Qadir. Using phase shift fingerprints and inertial measurements in support of precise localization in urban areas. *Personal and Ubiquitous Computing*, 23(5–6):861–872, 2019.
- [109] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.
- [110] Thomas Emter, Arda Saltoglu und Janko Petereit. Multi-sensor fusion for localization of a mobile robot in outdoor environments. In *Proc. Intl. Symposium on Robotics (ISR)*, Seiten 1–6. Munich, Germany, 2010.
- [111] Jakob Engel, Jurgen Sturm und Daniel Cremers. Semi-dense visual odometry for a monocular camera. In *Proc. Intl. Conf. Computer Vision*, Seiten 1449–1456, 2013.
- [112] Chiara Falsi, Davide Dardari, Lorenzo Mucchi und Moe Z. Win. Time of arrival estimation for UWB localizers in realistic environments. *J. Advances in Signal Processing (EURASIP)*, 2006(1):32–82, 2006.
- [113] Qigao Fan, Yaheng Wu, Jing Hui, Lei Wu, Zhenzhong Yu und Li-juan Zhou. Integrated navigation fusion strategy of INS/UWB for indoor carrier attitude angle and position synchronous tracking. *The Scientific World J.*, 14(20):851–858, 2014.
- [114] Qigao Fan, Biwen Sun, Yan Sun, Yaheng Wu und Xiangpeng Zhuang. Data fusion for indoor mobile robot positioning based on tightly coupled INS/UWB. *J. of Navigation*, 70(5):1079, 2017.
- [115] Qigao Fan, Biwen Sun, Yan Sun und Xiangpeng Zhuang. Performance enhancement of mems-based INS/UWB integration for indoor navigation applications. *IEEE Sensors J.*, 17(10):3116–3130, 2017.

- [116] Qigao Fan, Jie Jia, Peng Pan, Hai Zhang und Yan Sun. An improved INS/PDR/UWB integrated positioning method for indoor foot-mounted pedestrians. *Sensor J.*, 57(1):109–124, 2019.
- [117] Anton Fedosov, Sefan Misslinger und Peter Meier. Method for representing points of interest in a view of a real environment on a mobile device and mobile device therefor. Google Patents, July 2017. URL <https://www.google.com/patents/US20170213393>. US Patent App. 15/480,142.
- [118] Tobias Feigl und Christopher Mutschler. Method for predicting a motion of an object, method for calibrating a motion model, method for deriving a predefined quantity and method for generating a Virtual Reality view, European Patent No. EP201708794. Munich, Germany. European Patent Office, 2017.
- [119] Tobias Feigl und Christopher Mutschler. Apparatuses and Methods for Correcting Orientation Information from one or more Inertial Sensors, European Patent No. DE20171006222. Munich, Germany. European Patent Office, 2017.
- [120] Tobias Feigl und Christopher Mutschler. Low-Latency and Long-term Stable Pose (Position and Orientation) Estimation on Time-Series Sensor Data, European Patent No. WO2019129355. Munich, Germany. European Patent Office, 2017.
- [121] Tobias Feigl und Christopher Mutschler. Langzeitstabile Kopfforientierungsmessung in einzelpositionsbasierten Lokalisierungssystemen, European Patent No. US20190346280A1. Munich, Germany. European Patent Office, 2018.
- [122] Tobias Feigl, Christopher Mutschler, Michael Philippse und Eliise Kõre. Acoustical manipulation for redirected walking. In *Proc. Intl. Symp. on Virtual Reality Software and Technology (VRST)*, Seiten 45:1–45:2. Gothenburg, Sweden, 2017.
- [123] Tobias Feigl, Christopher Mutschler, Cosmin Bercea, Christian Daxer und Stephan Otto. Unsichtbare Langzeitstabile Orientierungsangleichung, European Patent No. WO2017198441A1. Munich, Germany. European Patent Office, 2018.
- [124] Tobias Feigl, Christopher Mutschler und Michael Philippse. Supervised Learning for Yaw Orientation Estimation. In *Proc. Intl.*

Conf. Indoor Positioning and Indoor Navigation (IPIN), Seiten 1–8. Nantes, France, 2018.

- [125] Tobias Feigl, Christopher Mutschler und Michael Philippsen. Human Compensation Strategies for Orientation Drifts. In *Proc. Intl. Conf. Virtual Reality and 3D User Interfaces (IEEE VR)*, Seiten 409–414. Reutlingen, Germany, 2018.
- [126] Tobias Feigl, Christopher Mutschler und Michael Philippsen. Head-to-Body-Pose Classification in No-Pose VR Tracking Systems. In *Proc. Intl. Conf. Virtual Reality and 3D User Interfaces (IEEE VR)*, Seiten 545–546. Reutlingen, Germany, 2018.
- [127] Tobias Feigl, Thorsten Nowak, Michael Philippsen, Thorsten Edelhäuser und Christopher Mutschler. Recurrent Neural Networks on Drifting Time-of-Flight Measurements. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–8. Nantes, France, 2018.
- [128] Tobias Feigl, Sebastian Kram, Philipp Woller, Ramiz H. Siddiqui, Michael Philippsen und Christopher Mutschler. A Bidirectional LSTM for Estimating Dynamic Human Velocities from a Single IMU. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–8. Pisa, Italy, 2019.
- [129] Tobias Feigl, Daniel Roth, Stefan Gradl, Markus Wirth, Marc Erich Latoschik, Björn Eskofier, Michael Philippsen und Christopher Mutschler. Sick Moves! Motion Parameters as Indicators of Simulator Sickness. *Trans. on Visualization and Computer Graphics (TVCG)*, 25(4):3146–3157, 2019.
- [130] Tobias Feigl, Lisa Gruner, Christopher Mutschler und Daniel Roth. Real-Time Gait Reconstruction for Virtual Reality using a Single Sensor. In *Proc. Intl. Symp. on Mixed Reality and Augmented Reality (ISMAR)*, Seiten 1–6, 2020.
- [131] Tobias Feigl, Sebastian Kram, Philipp Woller, Ramiz H. Siddiqui, Michael Philippsen und Christopher Mutschler. RNN-aided Human Velocity Estimation from a Single IMU. *Sensors J.*, 13(4):1–31, 2020.

- [132] Tobias Feigl, Ernst Eberlein Christopher Mutschler und Sebastian Kram. Robust ToA-Estimation using Convolutional Neural Networks (or other function approximations) on Randomized Channel Models, European Patent No. EP20202158525. Munich, Germany. European Patent Office, 2020.
- [133] Tobias Feigl, Andreas Porada, Steve Steiner, Christoffer Löffler, Christopher Mutschler und Michael Philippsen. Localization Limitations of ARCore, ARKit, and Hololens in Dynamic Large-Scale Industry Environments. In *Proc. Intl. Conf. Computer Graphics Theory and Applications (GRAPP)*, Seiten 307–318. Valletta, Malta, 2020.
- [134] Tobias Feigl, Sebastian Kram, Ernst Eberlein und Christopher Mutschler. Robust ToA-Estimation using Convolutional Neural Networks on Randomized Channel Models. *IEEE Trans. on Signal Processing (IEEE TSP)*, 68(8234):1–13, 2020 (submitted).
- [135] André Filipe Gonçalves Ferreira, Duarte Manuel Azevedo Fernandes, Andre Paulo Catarino und Joao L. Monteiro. Localization and positioning systems for emergency responders: A survey. *IEEE Communications Surveys and Tutorials*, 19(4):2836–2870, 2017.
- [136] André G. Ferreira, Duarte Fernandes, André P. Catarino und João L. Monteiro. Performance analysis of toa-based positioning algorithms for static and dynamic targets with low ranging measurements. *Sensors J.*, 17(8):1915, 2017.
- [137] André G. Ferreira, Duarte Fernandes, André P. Catarino, Ana M. Rocha und João L. Monteiro. A loose-coupled fusion of inertial and UWB assisted by a decision-making algorithm for localization of emergency responders. *Electronics*, 8(12):1463, 2019.
- [138] Louis B. Fertig und James H. McClellan. Instantaneous frequency estimation using linear prediction with comparisons to the desas. *IEEE Signal Processing Letters*, 3(2):54–56, 1996.
- [139] Carl Fischer, Kavitha Muthukrishnan und Mike Hazas. SLAM for pedestrians and ultrasonic landmarks in emergency response scenarios. In *Proc. Intl. Conf. Advances in Computers*, Seiten 103–160. Chennai, India, 2011.

- [140] Christian Forster, Luca Carlone, Frank Dellaert und Davide Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Trans. on Robotics*, 33(1):1–21, 2016.
- [141] Eric Foxlin. Head tracking relative to a moving vehicle or simulator platform using differential inertial sensors. In *Proc. Intl. Symp. AeroSense (SPIE)*, Seiten 133–144. Orlando, FL, 2000.
- [142] Eric Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Compu. Graphics and Appl.*, 25(6):38–46, 2005.
- [143] Katerina Fragkiadaki, Sergey Levine, Panna Felsen und Jitendra Malik. Recurrent network models for human dynamics. In *Proc. Intl. Conf. Computer Vision (ICCV)*, Seiten 4346–4354. Santiago, Chile, 2015.
- [144] Tak-chung Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.
- [145] Tobias Gadeke, Johannes Schmid, Wilhelm Stork und Klaus Müller-Glaser. Pedestrian dead reckoning for person localization in a wireless sensor network. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation*, Seiten 1–4. Guimaraes, Portugal, 2011.
- [146] Enrique García, Pablo Poudereux, Álvaro Hernández, Jesús Ureña und David Gualda. A robust UWB indoor positioning system for highly complex environments. In *Proc. Intl. Conf. Industrial Technology (ICIT)*, Seiten 3386–3391. Seville, Spain, 2015.
- [147] Christian Gentner und Markus Ulmschneider. Simultaneous localization and mapping for pedestrians using low-cost ultra-wideband system and gyroscope. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–8. Sapporo, Japan, 2017.
- [148] Felix A. Gers, Jürgen Schmidhuber und Fred A. Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471, 2000.
- [149] Laura Giarré, Federica Pascucci, Simone Morosi und Alessio Martinelli. Improved PDR localization via uwb-anchor based on-line calibration. In *Proc. Intl. Forum on Research and Technology for Society and Industry (RTSI)*, Seiten 1–5. Palermo, Italy, 2018.

- [150] Javier González, Jose-Luis Blanco, Cipriano Galindo, A. Ortiz-de Galisteo, Juan-Antonio Fernández-Madrigal, Francisco Angel Moreno und Jorge L. Martinez. Mobile robot localization based on ultra-wide-band ranging: A Particle Filter approach. *Robotics and autonomous systems*, 57(5):496–507, 2009.
- [151] Ali Grami. Chapter 12 - wireless communications. In Ali Grami, editor, *Introduction to Digital Communications*, Seiten 493–527. Academic Press, 2016.
- [152] Daniel Grupe. *Principles of Artificial Neural Networks*, chapter 4. World Scientific, 3rd edition, 2013. doi: 10.1142/8868.
- [153] Alex Graves. Generating sequences with Recurrent neural networks, In arXiv:1308.0850 [cs.NE], 2014.
- [154] Alex Graves, Abdel-rahman Mohamed und Geoffrey Hinton. Speech recognition with deep Recurrent Neural networks. In *Proc. Intl. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Seiten 6645–6649. Vancouver, Canada, 2013.
- [155] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink und Jürgen Schmidhuber. LSTM: A search space odyssey. *Trans. on Neural Network Learning Syst.*, 28(10):2222–2232, 2017.
- [156] Paul D. Groves. Navigation using inertial sensors [tutorial]. *IEEE Aerospace and Electronic Systems Magazine*, 30(2):42–69, 2015.
- [157] Felix Gruber. Einfluss dynamischer Objekttransformationen auf die Bewegungswahrnehmung in VR. Bachelor Thesis, Technische Hochschule Nürnberg, 2018.
- [158] Thomas von der Grün, Norbert Franke, Daniel Wolf, Nicolas Witt und Andreas Eidloth. A real-time tracking system for football match and training analysis. In Albert Heuberger, Günter Elst und Randolph Hanke, editors, *Microelectronic Systems: Circuits, Systems and Applications*, Seiten 199–212. Springer, 2011.
- [159] Lisa Gruner. Avatar Synchronisation zur Analyse von Bewegungswahrnehmung der unteren Extremitäten in VR. Bachelor Thesis, Technische Hochschule Nürnberg, 2019.

- [160] Fuqiang Gu, Kourosh Khoshelham, Chunyang Yu und Jianga Shang. Accurate step length estimation for Pedestrian Dead Reckoning localization using stacked autoencoders. *Trans. on Instrum. and Meas.*, 2(1):1–9, 2018.
- [161] Hang Guo, Junge Guo, Min Yu, Haibin Hong, Jian Xiong und Baolian Tian. A weighted combination Filter with nonholonomic constrains for integrated navigation systems. *Advances in Space Research*, 55(5):1470–1476, 2015.
- [162] Kexin Guo, Xiuxian Li und Lihua Xie. Ultra-wideband and odometry-based cooperative relative localization with application to multi-uav formation control. *IEEE Trans. on Cybernetics*, 50(6):2590–2603, 2019.
- [163] Dominik Gusenbauer, Carsten Isert und Jens Krösche. Self-contained indoor positioning on off-the-shelf mobile devices. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation*, Seiten 1–9. Zurich, Switzerland, 2010.
- [164] Ismail Guvenc und Chia-Chin Chong. A survey on TOA based wireless localization and nlos mitigation techniques. *IEEE Commun. Surveys & Tutorials*, 11(3):107–124, 2009.
- [165] Ismail Guvenc, Chia-Chin Chong und Fujio Watanabe. Joint TOA estimation and localization technique for UWB sensor network applications. In *Proc. Intl. Conf. Vehicular Technology*, Seiten 1574–1578. Dublin, Ireland, 2007.
- [166] Ismail Guvenc, Sinan Gezici und Zafer Sahinoglu. Ultra-wideband range estimation: Theoretical limits and practical algorithms. In *Proc. Intl. Conf. Ultra-Wideband*, volume 3, Seiten 93–96. Hannover, Germany, 2008.
- [167] Ulrich Hammes, Eric Wolsztynski und Abdelhak M Zoubir. Robust tracking and geolocation for wireless Networks in nlos environments. *J. of Selected Topics in Signal Processing*, 3(5):889–901, 2009.
- [168] Julius Hannink, Thomas Kautz, Cristian F. Pasluosta, Jens Barth, Samuel Schülein, Karl-Günter Gaßmann, Jochen Klucken und Bjoern M Eskofier. Mobile stride length estimation with deep Convolutional Neural networks. *Biomedical and Health Informatics J.*, 22(2):354–362, 2017.

- [169] Brecht Hanssens, Emmeric Tanghe, Davy P. Gaillot, Martine Liénard, Claude Oestges, David Plets, Luc Martens und Wout Joseph. An extension of the rimax multipath estimation algorithm for ultra-wideband channel modeling. *Wireless Communic. and Networking*, 2018(1):164–176, 2018.
- [170] Robert Harle. A survey of indoor inertial positioning systems for pedestrians. *IEEE Comm. Surveys & Tutorials*, 15(3):1281–1293, 2013.
- [171] Frank Hartmann, Dhafar Rifat und Wilhelm Stork. Hybrid indoor pedestrian navigation combining an INS and a spatial non-uniform uwb-network. In *Proc. Intl. Conf. Information Fusion (FUSION)*, Seiten 549–556. Heidelberg, Germany, 2016.
- [172] Janne Haverinen und Anssi Kemppainen. Global indoor self-localization based on the ambient magnetic field. *Robotics and Autonomous Systems*, 57(10):1028–1035, 2009.
- [173] Simon S. Haykin und Simon S. Haykin. *Neural Networks and learning machines*. Prentice Hall, 3rd edition, 2009. ISBN 978-0-13-147139-9.
- [174] Ding He, Fuhu Liu, Dave Pape, Greg Dawe und Dan Sandin. Video-based measurement of system latency. In *Proc. Intl. Conf. Immersive Projection Technology Workshop*, Seiten 78–90. Ames, IA, 2000.
- [175] Kai He, Yueyue Zhang, Yaping Zhu, Weiwei Xia, Ziyan Jia und Li-anfeng Shen. A hybrid indoor positioning system based on UWB and inertial navigation. In *Proc. Intl. Conf. Wireless Communications and Signal Processing (WCSP)*, Seiten 1–5, Nanjing, China, 2015.
- [176] Kaiming He, Xiangyu Zhang, Shaoqing Ren und Jian Sun. Deep residual learning for image recognition. In *Proc. Intl. Conf. Computer Vision and Pattern Recognition*, Seiten 770–778, 2016.
- [177] Ziming He, Yi Ma und Rahim Tafazolli. Improved high resolution TOA estimation for ofdm-wlan based indoor ranging. *IEEE Wireless Communic. Letters*, 2(2):163–166, 2013.

Literaturverzeichnis

- [178] Hendrik Hellmers, Andreas Eichhorn, Abdelmoumen Norrdine und Jörg Blankenbach. Imu/magnetometer based 3d indoor positioning for wheeled platforms in nlos scenarios. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–7. Barcelona, Spain, 2016.
- [179] Hendrik Hellmers, Zakaria Kasmi, Abdelmoumen Norrdine und Andreas Eichhorn. Accurate 3d positioning for a mobile platform in non-line-of-sight scenarios based on IMU/magnetometer sensor fusion. *Sensors J.*, 18(1):126, 2018.
- [180] Michiel Hermans und Benjamin Schrauwen. Training and analysing deep Recurrent Neural networks. In *Advances in Neural information Processing systems*, Seiten 190–198. MIT Press, 2013.
- [181] Federico Hernández und Juan Ignacio Giribet. Comparison between linearized Kalman Filters and Particle Filters applied to integrated navigation systems. In *Proc. Intl. Conf. Guidance, Navigation and Control Systems*. Loutraki, Greece, 2006.
- [182] René Herren, Andrea Sparti, Kiamar Aminian und Yves Schutz. The prediction of speed and incline in outdoor running in humans using accelerometry. *Medicine and Science in Sports and Exercise*, 31(7):1053–1059, 1999.
- [183] Oskar Herrmann. Feature extraction of a radio frequency based localization system using beta-vae. Bachelor Thesis, Friedrich-Alexander-University Erlangen-Nuremberg, 2019.
- [184] Joel A. Hesch, Dimitrios G. Kottas, Sean L. Bowman und Stergios I. Roumeliotis. Consistency analysis and improvement of Vision-aided inertial navigation. *IEEE Trans. on Robotics*, 30(1):158–176, 2013.
- [185] Joel A. Hesch, Dimitrios G. Kottas, Sean L. Bowman und Stergios I. Roumeliotis. Camera-imu-based localization: Observability analysis and consistency improvement. *Intl. J. of Robotics Research*, 33(1):182–201, 2014.
- [186] Wolfgang Hess, Damon Kohler, Holger Rapp und Daniel Andor. Real-time loop closure in 2d lidar SLAM. In *Proc. Intl. Conf. Robotics and Automation (ICRA)*, Seiten 1271–1278. Stockholm, Sweden, 2016.

- [187] Jacob Y. Hesterman, Matthew A. Kupinski, Eric Clarkson und Harrison H. Barrett. Hardware assessment using the multi-module, multi-resolution system : A signal-detection study. *Medical Physics*, 34(7):3034–3044, 2007.
- [188] Sebastian Hilsenbeck, Dmytro Bobkov, Georg Schroth, Robert Huitl und Eckehard Steinbach. Graph-based data fusion of pedometer and WiFi measurements for mobile indoor positioning. *Proc. Intl. Conf. Pervasive and Ubiquitous Computing (UbiComp)*, 14(1):147–158, 2014.
- [189] Peter Hippe. *Windup in control: Its Effects and Their Prevention*. Springer Science+Business Media, London, UK, 2006.
- [190] Ngoc-Huynh Ho, Phuc Huu Truong und Gu-Min Jeong. Step-detection and adaptive step-length estimation for pedestrian dead-reckoning at various walking speeds using a smartphone. *Sensors J.*, 16(9):1423–1432, 2016.
- [191] Sepp Hochreiter und Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [192] Sepp Hochreiter und Jürgen Schmidhuber. LSTM can solve hard long time lag problems. In M. C. Mozer, M. I. Jordan und T. Petsche, editors, *Advances in Neural Information Processing Systems*, Seiten 473–479. MIT Press, 1997.
- [193] Guenter Hofmann und Marco Breiling. Device and method for determining a time of arrival of a receive sequence, 2009. US Patent 7,627,063.
- [194] Jeroen D. Hol. *Sensor fusion and calibration of inertial sensors, Vision, ultra-wideband and GPS*. PhD thesis, Linköping University Electronic Press, 2011.
- [195] Jeroen D. Hol, Fred Dijkstra, Henk Luinge und Thomas B. Schon. Tightly coupled UWB/IMU pose estimation. In *Proc. Intl. Conf. Ultra-Wideband*, Seiten 688–692, Vancouver, Canada, 2009.
- [196] Kurt Hornik, Maxwell B. Stinchcombe und Halbert White. Multi-layer feedforward Networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

Literaturverzeichnis

- [197] Siavash Hosseinyalamdary. Deep Kalman Filter: Simultaneous multi-sensor integration and modelling; a GNSS/IMU case study. *Sensors J.*, 18(5):64–72, 2018.
- [198] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *J. Educational Psychology*, 24(1):417–441, 1933.
- [199] Samuel House, Sean Connell, Ian Milligan, Daniel Austin, Tama-ra L. Hayes und Patrick Chiang. Indoor localization using pedestrian dead reckoning updated with rfid-based fiducials. In *Proc. Intl. Conf. Engineering in Medicine and Biology Society*, Seiten 7598–7601, Boston, MA, 2011.
- [200] Guoquan P. Huang, Anastasios I. Mourikis und Stergios I. Rou-meliotis. An observability-constrained sliding window Filter for SLAM. In *Proc. Intl. Conf. Intelligent Robots and Systems (RSJ)*, Seiten 65–72. San Francisco, CA, 2011.
- [201] Sergey Ioffe und Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. Intl. Conf. Machine Learning*, Seiten 448–456. Lille, France, 2015.
- [202] Dan Iter, Jonathan Kuck, Philip Zhuang und Learning. Target tracking with Kalman Filtering, knn and lstms. Technical report, Computer Science Dept., Stanford, CA, 2016.
- [203] Pavel Ivanov, Matti Raitoharju und Robert Piché. Kalman-type Filters and smoothers for pedestrian dead reckoning. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 206–212. Nantes, France, 2018.
- [204] Hong Lee Jae, Ju Hojin und Gook Park Chan. Error Analysis of PDR System Using Dual Foot-mounted IMU. *Sensors J.*, 94(1):41–56, 2019.
- [205] Stephan Jaeckel, Leszek Raschkowski, Kai Börner und Lars Thiele. QuaDRiGa: A 3-d multi-cell channel model with time evolution for enabling virtual field trials. *IEEE Trans. on Antennas and Propagation*, 62(6):3242–3256, 2014.

- [206] Jasper Jahn, Ulrich Batzer, Jochen Seitz, Lucila Patino-Studencka und Javier Gutiérrez Boronat. Comparison and evaluation of acceleration based step length estimators for handheld devices. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–6. Nantes, France, 2010.
- [207] Christian Jakob. Virtual-Reality-optimierte Sensor Fusion für eine langzeitstabile Gestenerkennung der oberen Extremität. Master Thesis, Technische Hochschule Nürnberg, 2018.
- [208] Andreas Jakobsson, A. Lee Swindlehurst und Petre Stoica. Subspace-based estimation of time delays and Doppler shifts. *IEEE Trans. on Signal Processing*, 46(9):2472–2483, 1998.
- [209] Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato und Yann LeCun. What is the best multi-stage architecture for object recognition? In *Proc. Intl. Conf. Computer Vision (ICCV)*, Seiten 2146–2153. Kyoto, Japan, 2009.
- [210] Changhui Jiang, Shuai Chen, Yuwei Chen, Boya Zhang, Ziyi Feng, Hui Zhou und Yuming Bo. A MEMS IMU de-noising method using long short term memory Recurrent neural networks (LSTM-RNN). *Sensors J.*, 18(10):34–56, 2018.
- [211] Changhui Jiang, Yuwei Chen, Shuai Chen, Yuming Bo, Wei Li, Wenxin Tian und Jun Guo. A mixed deep Recurrent neural network for MEMS gyroscope noise suppressing. *Electronics J.*, 8(2):181, 2019.
- [212] Chao Jiang, Muhammad Fahad, Yi Guo und Yingying Chen. Robot-assisted smartphone localization for human indoor tracking. *Robotics and Autonomous Systems*, 106(1):82–94, 2018.
- [213] Yonghang Jiang, Zhenjiang Li und Jianping Wang. Ptrack: Enhancing the applicability of pedestrian tracking with wearables. *Trans. on Mobile Computing*, 18(2):431–443, 2018.
- [214] Antonio Ramón Jiménez und Fernando Seco. Finding objects using UWB or BLE localization technology: A museum-like use case. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–8. Sapporo, Japan, 2017.

Literaturverzeichnis

- [215] Antonio Ramón Jiménez, Fernando Seco, José Carlos Prieto und Jorge Guevara. Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU. In *Proc. Intl. Workshop on Positioning Navigation and Comm.*, Seiten 135–143. Dresden, Germany, 2010.
- [216] Yunye Jin, Wee-Seng Soh, Mehul Motani und Wai-Choong Wong. A robust indoor pedestrian tracking system with sparse infrastructure support. *IEEE Trans. on Mobile Computing*, 12(7):1392–1403, 2012.
- [217] Rafal Józefowicz, Wojciech Zaremba und Ilya Sutskever. An empirical exploration of Recurrent network architectures. In *Proc. Intl. Conf. Machine Learning (ICML)*, Seiten 2342–2350. Lille, France, 2015.
- [218] Simon Julier, Jeffrey Uhlmann und Hugh F Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in Filters and estimators. *IEEE Trans. on Automatic Control*, 45(3):477–482, 2000.
- [219] Michael Kaess, Ananth Ranganathan und Frank Dellaert. isam: Incremental smoothing and mapping. *IEEE Trans. on Robotics*, 24(6):1365–1378, 2008.
- [220] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ilă, John J. Leonard und Frank Dellaert. isam2: Incremental smoothing and mapping using the Bayestree. *Intl. J. of Robotics Researc*, 31(2):216–235, 2012.
- [221] Nal Kalchbrenner, Edward Grefenstette und Phil Blunsom. A convolutional neural network for modelling sentences. In *Proc. Intl. Conf. Assoc. for Comp. Linguistics*, Seiten 655–664. Baltimore, Maryland, 2014.
- [222] Rudolf Kálmán Kalman. A New Approach to Linear Filtering and Prediction Problems. *J. of Basic Engineering*, 82(1):35–45, 1960.
- [223] Jiheon Kang, Joonbeom Lee und Doo-Seop Eom. Smartphone-based traveled distance estimation using individual walking patterns for indoor localization. *Sensors J.*, 18(9):3149, 2018.

- [224] Wonho Kang, Seongho Nam, Youngnam Han und Sookjin Lee. Improved heading estimation for smartphone-based indoor positioning systems. In *Proc. Intl. Symp. Personal, Indoor and Mobile Radio Commu.*, Seiten 2449–2453. Sydney, Australia, 2012.
- [225] Jani Kappi, Jari Syrjäinne und Jukka Saarinen. MEMS-IMU based pedestrian navigator for handheld devices. In *Proc. Intl. Tech. Meeting Satellite Division of Institute of Navigation (ION)*, Seiten 1369–1373. Salt Lake City, UT, 2001.
- [226] Dean Karantonis, Michael Narayanan, Merryn Mathie, Nigel Lovell und Branko Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Trans. Information Techno. in Biomedicine*, 10(1):156–167, 2006.
- [227] Maximilian Karl, Maximilian Soelch, Justin Bayer und Patrick Van der Smagt. Deep variational Bayesfilters: Unsupervised learning of state space models from raw data. In *Proc. Intl. Conf. Intuitive Physics Workshop (NIPS)*, Seiten 2–5. Reutlingen, Germany, 2016.
- [228] Andrej Karpathy, Justin Johnson und Li Fei-Fei. Visualizing and understanding Recurrent networks, In arXiv:1506.02078 [cs.LG], 2015.
- [229] Paul Kemppi, Thomas Rautiainen, Victor Ranki, Florian Belloni und Juuso Pajunen. Hybrid positioning system combining angle-based localization, pedestrian dead reckoning and map filtering. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation*, Seiten 1–7. Zurich, Switzerland, 2010.
- [230] Thomas Kersten, Heinz Przybilla, Maren. Lindstaedt, Felix. Tschirschwitz und Martin Misgaiski-Hass. Comparative geometrical investigations of hand-held scanning systems. *Intl. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 7(10):507–514, 2016.
- [231] Samina Khalid, Tehmina Khalil und Shamila Nasreen. A survey of feature selection and feature extraction techniques in machine learning. In *Proc. Intl. Conf. Science and Information Conference (SAI)*, Seiten 372–378. London, UK, 2014.

Literaturverzeichnis

- [232] Nguyen-Huu Khanh, Lee KyungHo und Lee Seon-Woo. An indoor positioning system using pedestrian dead reckoning with WiFi and map-matching aided. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 623–631. Sapporo, Japan, 2017.
- [233] Do Kim, Hojin Ju und Chan Gook Park. Comparison of step length estimation models using inertial sensor on pelvis. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation*, Seiten 222–225. Nantes, France, 2018.
- [234] J. Won Kim, Han Jin Jang, Dong-Hwan Hwang und Chansik Park. A step, stride and heading determination for the pedestrian navigation system. *Global Positioning Systems J.*, 3(2):273–279, 2004.
- [235] Diederik P. Kingma und Jimmy Ba. Adam: A method for stochastic optimization, In arXiv:1412.6980 [CS], 2017.
- [236] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj und Daniel J. Inman. 1d Convolutional Neural Networks and applications: A survey, In arXiv:1905.03554v1 [eess.SP], 2019.
- [237] Matthew Klingensmith, Ivan Dryanovski, Siddhartha Srinivasa und Jizhong Xiao. Chisel: Real Time Large Scale 3D Reconstruction Onboard a Mobile Device. In *Proc. Intl. Conf. Robotics Science and Systems*, Seiten 1–9. Pittsburgh, PA, 2015.
- [238] Jens M. Knudsen und Poul G. Hjorth. *Elements of Newtonian Mechanics*. Springer Berlin Heidelberg, 1995.
- [239] Manon Kok und Thomas B Schön. A fast and robust algorithm for orientation estimation using inertial sensors. *IEEE Signal Processing Letters*, 26(11):1673–1677, 2019.
- [240] Manon Kok, Niklas Wahlström, Thomas B. Schön und Fredrik Gustafsson. Mems-based inertial navigation based on a magnetic field map. In *Proc. Intl. Conf. Acoustics, Speech and Signal Processing*, Seiten 6466–6470, Vancouver, Canada, 2013.
- [241] Manon Kok, Jeroen Hol und Thomas Schön. An optimization-based approach to human body motion capture using inertial sensors. In *Proc. Intl. Federation of Automatic Control (IFAC)*, Seiten 79–85. Cape Town, South Africa, 2014.

- [242] Manon Kok, Jeroen D. Hol und Thomas B. Schön. Indoor positioning using ultrawideband and inertial measurements. *IEEE Trans. on Vehicular Technology*, 64(4):1293–1303, 2015.
- [243] Manon Kok, Jeroen D. Hol und Thomas B. Schön. Using inertial sensors for position and orientation estimation. *Foundations and Trends in Signal Processing*, 11(1):1–153, 2017.
- [244] Kishore Reddy Konda und Roland Memisevic. Learning visual odometry with a Convolutional network. In *Proc. Intl. Conf. Computer Vision Theory and Applications (VISAPP)*, Seiten 486–490. Berlin, Germany, 2015.
- [245] Hema Swetha Koppula und Ashutosh Saxena. Anticipating human activities for reactive robotic response. In *Proc. Intl. Conf. Intelligent Robots and Systems*, Seiten 2071–2071. Tokyo, Japan, 2013.
- [246] Masakatsu Kourogi und Takeshi Kuratta. A wearable augmented reality system with personal positioning based on walking locomotion analysis. In *Proc. Intl. Conf. Mixed and Augmented Reality*, Seiten 342–349. Tokyo, JP, 2003.
- [247] Sebastian Kram, Maximilian Stahlke, Tobias Feigl, Jochen Seitz und Jörn Thielecke. UWB Channel Impulse Responses for Positioning in Complex Environments: A Detailed Feature Analysis. *Sensors J.*, 24(2):1–23, 2019.
- [248] Rahul G. Krishnan, Uri Shalit und David Sontag. Deep Kalman Filters, In arXiv:1511.05121 [CS], 2015.
- [249] Sampo Kuutti, Saber Fallah, Konstantinos Katsaros, Mehrdad Dianati, Francis McCullough und Alexandros Mouzakitis. A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications. *IEEE Internet of Things J.*, 5(2):829–846, 2018.
- [250] Elise Lachat, Hélène Macher, Tania Landes und Pierre Grussenmeyer. Assessment and calibration of a RGB-D camera towards a potential use for close-range 3D modeling. *Remote Sens.*, 1(7):13070–13097, 2015.

Literaturverzeichnis

- [251] Quentin Ladetto. On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering. In *Proc. Intl. Conf. Institute of Navigation GPS (ION)*, Seiten 1735–1740. Salt Lake City, UT, 2000.
- [252] Kun-Chan Lan und Wen-Yuh Shih. On calibrating the sensor errors of a PDR-based indoor localization system. *Sensors J.*, 13(4):4781–4810, 2013.
- [253] Steven M. LaValle, Anna Yershova, Max Katsev und Michael Antonov. Head tracking for the oculus rift. In *Proc. Intl. Conf. Robotics and Automation (ICRA)*, Seiten 187–194. Hong Kong, China, 2014.
- [254] Quoc V. Le, Navdeep Jaitly und Geoffrey E. Hinton. A simple way to initialize Recurrent networks of rectified linear units, In arXiv:1504.00941 [cs.NE], 2015.
- [255] Yann LeCun, Yoshua Bengio und Geoffrey E. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [256] Boon-Giin Lee und Wan-Young Chung. Multitarget three-dimensional indoor navigation on a PDA in a wireless sensor network. *IEEE Sensors J.*, 11(3):799–807, 2010.
- [257] Joon-Yong Lee und Robert A. Scholtz. Ranging in a dense multipath environment using an UWB radio link. *Selected Areas in Communications*, 20(9):1677–1683, 2002.
- [258] Seungwoo Lee, Byounggeun Kim, Hoon Kim, Rhan Ha und Ho-jung Cha. Inertial sensor-based indoor pedestrian localization with minimum 802.15. 4a configuration. *IEEE Trans. on Industrial Informatics*, 7(3):455–466, 2011.
- [259] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart und Paul Furgale. Keyframe-based visual-inertial odometry using nonlinear optimization. *Intl. J. of Robotics Research*, 34(3):314–334, 2015.
- [260] Kenneth Levenberg. A method for the solution of certain nonlinear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.

- [261] Haiwen Li, Nae Zheng, Xiyu Song und Yinghua Tian. Fast estimation method of space-time two-dimensional positioning parameters based on hadamard product. *J. Antennas and Propagation*, 2018(1):1–9, 2018.
- [262] Jin-feng Li, Qing-hui Wang, Xiao-mei Liu und Mu-yuan Zhang. An autonomous waist-mounted Pedestrian Dead Reckoning system by coupling low-cost MEMS inertial sensors and GPS receiver for 3d urban navigation. *J. of Engineering Science and Technology Review*, 7(1):369, 2014.
- [263] Kang Li, Can Wang, Sheng Huang, Guoyuan Liang, Xinyu Wu und Yubin Liao. Self-positioning for uav indoor navigation based on 3d laser scanner, UWB and ins. In *Proc. Intl. Conf. Information and Automation (ICIA)*, Seiten 498–503, Ningbo, China, 2016.
- [264] Linchuan Li, Zhiyong Wu, Mingxing Xu, Helen Meng und Lianhong Cai. Combining CNN and BLSTM to extract textual and acoustic features for recognizing stances in mandarin ideological debate competition. In *Proc. Intl. Conf. Speech Communication Assoc. (INTERSPEECH)*, Seiten 1392–1396, San Francisco, CA, 2016.
- [265] Mingyang Li und Anastasios I Mourikis. High-precision, consistent EKF-based visual-inertial odometry. *Intl. J. of Robotics Research*, 32(6):690–711, 2013.
- [266] Xin Li und Yang Wang. Research on the UWB/IMU fusion positioning of mobile vehicle based on motion constraints. *Acta Geodaetica et Geophysica*, 7(1):1–19, 2020.
- [267] Xin Li, Jian Wang und Chunyan Liu. A Bluetooth/PDR integration algorithm for an indoor positioning system. *Sensors J.*, 15(10):24862–24885, 2015.
- [268] Xin Li, Yan Wang und Kourosh Khoshelham. A robust and adaptive complementary Kalman Filter based on mahalanobis distance for ultra wideband/inertial measurement unit fusion positioning. *Sensors J.*, 18(10):3435, 2018.
- [269] Xin Li, Yan Wang und Kourosh Khoshelham. UWB/PDR tightly coupled navigation with robust extended Kalman Filter for NLOS environments. *Mobile Information Systems*, 6(5):1–14, 2018.

Literaturverzeichnis

- [270] Xin Li, Yan Wang und Kourosh Khoshelham. Comparative analysis of robust extended Kalman Filter and incremental smoothing for uwb/pdr fusion positioning in nlos environments. *Acta Geodaetica et Geophysica*, 54(2):157–179, 2019.
- [271] Xinrong Li und Kaveh Pahlavan. Super-resolution TOA estimation with diversity for indoor geolocation. *IEEE Trans. on Wireless Communic.*, 3(1):224–234, 2004.
- [272] You Li, Peng Zhang, Xiaoji Niu, Yuan Zhuang, Haiyu Lan und Naser El-Sheimy. Real-time indoor navigation using smartphone sensors. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–10. Alberta, Canada, 2015.
- [273] Zengke Li, Ren Wang, Jingxiang Gao und Jian Wang. An approach to improve the positioning performance of gps/INS/UWB integrated system with two-step filter. *Remote Sensing*, 10(1):19, 2018.
- [274] Tao Lin, Lingran Li und Gérard Lachapelle. Multiple sensors integration for pedestrian indoor navigation. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–9. Alberta, Canada, 2015.
- [275] Peter Lincoln, Alex Blate, Montek Singh, Turner Whitted, Andrei State, Anselmo Lastra und Henry Fuchs. From motion to photons in 80 microseconds: Towards minimal latency for virtual and augmented reality. *IEEE Trans. on Visualization and Computer Graphics (TVCG)*, 22(4):1367–1376, 2016.
- [276] Fei Liu, Jian Wang, Jixian Zhang und Houzeng Han. An indoor localization method for pedestrians base on combined uwb/pdr/floor map. *Sensors J.*, 19(11):2578, 2019.
- [277] Haitao Liu, Yew-Soon Ong, Xiaobo Shen und Jianfei Cai. When gaussian process meets big data: A review of scalable gps. *Trans. on Neural Networks and Learning Systems*, 53(7):1346–1356, 2020.
- [278] Jianfeng Liu, Jiexin Pu, Lifan Sun und Zishu He. An approach to robust INS/UWB integrated positioning for autonomous indoor mobile robots. *Sensors J.*, 19(4):950, 2019.
- [279] Ran Liu, Chau Yuen, Tri-Nhut Do, Dewei Jiao, Xiang Liu und U-Xuan Tan. Cooperative relative positioning of mobile users by

- fusing IMU inertial and UWB ranging information. In *Proc. Intl. Conf. Robotics and Automation (ICRA)*, Seiten 5623–5629. Marina Bay Sands, Singapore, 2017.
- [280] Tao Liu, Xiaoji Niu, Jian Kuang, Shuai Cao, Lei Zhang und Xiang Chen. Doppler shift mitigation in acoustic positioning based on pedestrian dead reckoning for smartphone. *IEEE Trans. on Instrumentation and Measurement*, 1(1), 2020.
- [281] Yuan Liu, Sheng Li, Qiang Sun, Chenfei Chang, Guangjian He und Xiao Kang. An UWB/PDR Fusion Algorithm Based on Improved Square Root Unscented Kalman Filter. In *Proc. Intl. Conf. Chinese Control (CCC)*, Seiten 4124–4129. Guangzhou, China, 2019.
- [282] Jonathan Long, Evan Shelhamer und Trevor Darrell. Fully Convolutional Networks for semantic segmentation. In *Proc. Intl. Conf. Computer Vision and Pattern Recognition, (CVPR)*, Seiten 3431–3440. Boston, MA, 2015.
- [283] Jiazen Lu, Kechuan Chen, Baoguo Li und Miaomiao Dai. Hybrid navigation method of INS/PDR based on action recognition. *IEEE Sensors J.*, 18(20):8541–8548, 2018.
- [284] Jean-Luc Lugrin, Florian Kern, Ruben Schmidt, Constantin Kleinbeck, Daniel Roth, Christian Daxer, Tobias Feigl, Christopher Mutschler und Marc-Erich Latoschik. A Location-Based VR Museum. In *Proc. Intl. Conf. Virtual Worlds and Games for Serious Applications (VS-Games)*, Seiten 1–8. Würzburg, Germany, 2018.
- [285] Jean-Luc Lugrin, Florian Kern, Constantin Kleinbeck, Daniel Roth, Christian Daxer, Tobias Feigl, Christopher Mutschler und Marc Erich Latoschik. A Framework for Location-Based VR Applications. In *Virtuelle und Erweiterte Realität: 16. Workshop der GI-Fachgruppe VR/AR (Berichte aus der Informatik)*, Seiten 148–159. Fulda, Germany, 2019.
- [286] Dimitrios Lymberopoulos, Jie Liu, Xue Yang, Romit Roy Choudhury, Vlado Handziski und Souvik Sen. A realistic evaluation and comparison of indoor location technologies: experiences and lessons learned. In *Proc. Intl. Conf. Information Processing in Sensor Networks - IPSN*, Seiten 178–189. ACM Press, 2015.

Literaturverzeichnis

- [287] Sebastian Madgwick. An efficient orientation Filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK)*, 25(3):113–118, 2010.
- [288] Ishrat Maherin und Qilian Liang. Multistep information fusion for target detection using UWB radar sensor network. *IEEE Sensors J.*, 15(10):5927–5937, 2015.
- [289] Utkarsh Mall, G. Roshan Lal, Siddhartha Chaudhuri und Parag Chaudhuri. A deep Recurrent framework for cleaning motion capture data, In arXiv:1712.03380 [cs], 2017.
- [290] Andrea Mannini und Angelo Maria Sabatini. Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers. *Sensors J.*, 10(1):1154–1175, 2010.
- [291] Andrea Mannini und Angelo Maria Sabatini. Walking speed estimation using foot-mounted inertial sensors: Comparing machine learning and strap-down integration methods. *Medical Eng. & Physics J.*, 36(10):1312–1321, 2014.
- [292] Jani Mantyjarvi, Johan Himberg und Tapio Seppanen. Recognizing human motion with multiple acceleration sensors. In *IEEE Proc. Intl. Conf. Systems, Man, and Cybernetics*, Seiten 747–752. Tucson, AZ, 2001.
- [293] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *J. Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.
- [294] Alessio Martinelli, Han Gao, Paul D. Groves und Simone Morosi. Probabilistic context-aware step length estimation for pedestrian dead reckoning. *Sensors J.*, 18(4):1600–1611, 2017.
- [295] Andrea Masiero, Alberto Guarnieri, Francesco Pirotti und Antonio Vettore. A Particle Filter for smartphone-based indoor pedestrian navigation. *Micromachines*, 5(4):1012–1033, 2014.
- [296] Stephanie Mehltretter. Radio Localization to Enable Robust People Tracking in High-Resolution Images. Bachelor Thesis, Friedrich-Alexander-University Erlangen-Nuremberg, 2020.

- [297] Peter Meier, Selim Ben Himane, Stefan Misslinger und Ben Blachnitzky. Methods and systems for determining the pose of a camera with respect to at least one object of a real environment, 2015. URL <https://www.google.com/patents/US20150310666>. US Patent App. 14/633,386.
- [298] German Mendoza Silva, Joaquín Torres-Sospedra und Joaquín Huerta. A meta-review of indoor positioning systems. *Sensors J.*, 19(20):4507–4531, 2019.
- [299] Chang Michael B., Ullman Tomer, Torralba Antonio und Tenenbaum Joshua B. A compositional object-based approach to learning physical dynamics. In *Proc. Intl. Conf. Learning Representations (ICLR)*, Seite 155–173. Toulon, France, 2017.
- [300] Anton Milan, Seyed Hamid Rezatofighi, Anthony R. Dick, Ian D. Reid und Konrad Schindler. Online multi-target tracking using Recurrent neural networks. *Assoc. Advancement of Artificial Intelligence (AAAI)*, 6(3):4225–4232, 2017.
- [301] Sanjit Kumar Mitra und Yonghong Kuo. *Digital Signal Processing: a computer-based approach*, volume 2. McGraw-Hill New York, 2006.
- [302] Anastasios I. Mourikis und Stergios I. Roumeliotis. A multi-state constraint Kalman Filter for Vision-aided inertial navigation. In *Proc. Intl. Conf. Robotics and Automation*, Seiten 3565–3572. Roma, Italy, 2007.
- [303] Raul Mur-Artal und Juan D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. on Robotics*, 33(5):1255–1262, 2017.
- [304] Raul Mur-Artal, Jose Maria Martinez Montiel und Juan D Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans. on Robotics*, 31(5):1147–1163, 2015.
- [305] Masayuki Murata, Dragan Ahmetovic, Daisuke Sato, Hironobu Takagi, Kris M. Kitani und Chieko Asakawa. Smartphone-based localization for blind navigation in building-scale indoor environments. *Pervasive and Mobile Computing*, 57:14–32, 2019.

Literaturverzeichnis

- [306] Erik Murphy-Chutorian, Anup Doshi und Mohan Manubhai Tri-vedi. Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. In *Proc. Intl. Conf. Intelligent Transportation Systems*, Seiten 709–714. Bellevue, WA, 2007.
- [307] Christopher Mutschler, Holger Ziekow und Zbigniew Jerzak. The DEBS 2013 grand challenge. In *Proc. Intl. Conf. Distributed event-based systems (DEBS)*, Seite 289. Arlington, TX, 2013.
- [308] Christopher Mutschler, Sebastian Kram, Christian Nickel, Tobias Feigl und Jochen Seitz. Vorrichtung und Verfahren zu effizienten Zustandsbestimmung und Lokalisierung zwischen mobilen Plattformen, European Patent No. EP2018025108. Munich, Germany. European Patent Office, 2018.
- [309] Christopher Mutschler, Sebastian Kram, Christian Nickel, Tobias Feigl, Jochen Seitz und Niels Hadaschik. Effiziente Zustandsbestimmung und Lokalisierung durch Datenaustausch und zwischen mobilen Plattformen und logischen Paarbildungen für gemeinsame Bewegungsmodell zur Lokalisierung und Assoziation, European Patent No. WO2019197006A1. Munich, Germany. European Patent Office, 2019.
- [310] Vinod Nair und Geoffrey E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proc. Intl. Conf. Machine Learning*, Seiten 807–814. Haifa, Israel, 2010.
- [311] A. Niitsoo, T. Edelhäußer, E. Eberlein, N. Hadaschik und C. Mutschler. A Deep Learning Approach to Position Estimation from Channel Impulse Responses. *Sensors J.*, 19(5):1064–1087, 2019.
- [312] Arne Niitsoo, Thorsten Edelhäußer und Christopher Mutschler. Convolutional neural networks for position estimation in TDoA-based locating systems. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–8. Nantes, France, 2018.
- [313] Arne Niitsoo, Thorsten Edelhäußer, Ernst Eberlein, Niels Hadaschik und Christopher Mutschler. Deep learning approach to position estimation from channel impulse responses. *Sensors J.*, 19(1064):1–24, 2019.

- [314] John-Olof Nilsson, Isaac Skog, Peter Händel und K. V. S. Hari. Foot-mounted INS for everybody—an open-source embedded implementation. In *Proc. Intl. Conf. Position, Location and Navigation Symposium (ION)*, Seiten 140–145. Myrtle Beach, SC, 2012.
- [315] Abdelmoumen Norrdine, Zakaria Kasmi und Jörg Blankenbach. Step detection for ZUPT-aided inertial pedestrian navigation system using foot-mounted permanent magnet. *Sensors J.*, 16(7):766–773, 2016.
- [316] Thorsten Nowak und Andreas Eidloth. Dynamic Multipath Mitigation Applying Unscented Kalman Filters in Local Positioning Systems. *Intl. J. Microwave and Wireless Technologies*, 3(3):365–372, 2011.
- [317] Katja Nummiaro, Esther Koller-Meier und Luc Van Gool. An adaptive color-based Particle filter. *Image and Vision Computing*, 21(1):99–110, 2003.
- [318] Henri Nurminen, Tohid Ardesthiri, Robert Piché und Fredrik Gustafsson. A nlos-robust TOA positioning Filter based on a skew-t measurement noise model. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–7. Banff, Canada, 2015.
- [319] Brian J. Odelson, Murali R. Rajamani und James B. Rawlings. A new autocovariance least-squares method for estimating noise covariances. *Automatica*, 42(2):303–308, 2006.
- [320] Radiocommunication Sector of ITU (ITU-R) Spectrum Management. Comparison of time-difference-of-arrival and angle-of-arrival methods of Signal geolocation. Technical report, SM.2211-2, 2018.
- [321] Fredrik Olsson, Jouni Rantakokko und Jonas Nygårds. Cooperative localization using a foot-mounted inertial navigation system and ultrawideband ranging. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 122–131. Busan, South Korea, 2014.
- [322] Felix Ott. Komplementieren Relativer und Absoluter Eigenlokalisierungsverfahren. Master Thesis, Friedrich-Alexander-University Erlangen-Nuremberg, 2019.

Literaturverzeichnis

- [323] Felix Ott, Tobias Feigl, Christoffer Löffler und Christopher Mutschler. ViPR: Visual-Odometry-aided Pose Regression for 6DoF Camera Localization. In *Proc. Intl. Conf. Computer Vision and Pattern Recognition (CVPR) - Workshop on Long-Term Visual Localization, Visual Odometry and Geometric and Learning-based SLAM*, Seiten 42–43. Seattle, Washington, 2020.
- [324] Lea Otte. Eine explorative Untersuchung zu Textverständnis und Lerneffekt im Medium Virtuelle Realität - Lohnt sich der Implementierungsaufwand für die Anwendung im Fraunhofer IIS? Bachelor Thesis, Technische Hochschule Nürnberg, 2019.
- [325] Stephan Otto, Tobias Feigl, Christian Daxer, Alexander Bruckmann, Christoffer Löffler, Christopher Mutschler und Marc Faßbinder. Vr/ar-spezifische Angleichung der Pose, European Patent No. WO2019114925A1. Munich, Germany. European Patent Office, 2017.
- [326] Deepak Pai, Inguva Sasi, Phani Shekhar Mantripragada, Mudit Malpani und Nitin Aggarwal. Padati: A robust pedestrian dead reckoning system on smartphones. In *Proc. Intl. Conf. Trust, Security and Privacy in Computing and Commu.*, Seiten 2000–2007. Liverpool, UK, 2012.
- [327] George Papagiannakis, Gurminder Singh und Nadia Magnenat-Thalmann. A survey of mobile and wireless technologies for augmented reality systems. *Computer Animation and Virtual Worlds*, 19(1):3–22, 2008.
- [328] Abhinav Parate, Meng-Chieh Chiu, Chanel Chadowitz, Deepak Ganeshan und Evangelos Kalogerakis. Risq: Recognizing smoking gestures with inertial sensors on a wristband. In *Proc. Intl. Conf. Mobile Systems, Applications, and Services (MobiSys)*, Seiten 149–161. Bretton Woods, NH, 2014.
- [329] Razvan Pascanu, Tomas Mikolov und Yoshua Bengio. On the Difficulty of Training Recurrent Neural Networks. In *Proc. Intl. Conf. Machine Learning*, Seiten 1310–1318. Atlanta, GA, 2013.
- [330] Razvan Pascanu, Çaglar Gülcöhre, Kyunghyun Cho und Yoshua Bengio. How to construct deep Recurrent neural networks. In *Proc. Intl. Conf. Learning Representations (ICLR)*, Seiten 78–85. Banff, Canada, 2014.

- [331] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho und Yoshua Bengio. How to construct deep Recurrent neural networks, In arXiv:1312.6026 [cs, stat], 2014.
- [332] Valter Pasku, Alessio De Angelis, Marco Dionigi, Antonio Moschitta, Guido De Angelis und Paolo Carbone. Analysis of nonideal effects and performance in magnetic positioning systems. *IEEE Trans. on Instrumentation and Measurement*, 65(12):2816–2827, 2016.
- [333] Ling Pei, Jingbin Liu, Robert Guiness, Yuwei Chen, Heidi Kuusniemi und Ruizhi Chen. Using LS-SVM Based Motion Recognition for Smartphone Indoor Wireless Positioning. *Sensors J.*, 12(1):6155–6175, 2012.
- [334] Stefano Pellegrini, Andreas Ess, Konrad Schindler und Luc Van Gool. You'll never walk alone: modeling social behavior for multi-target tracking. In *Proc. Intl. Conf. Computer Vision (ICCV)*, Seiten 36–48. Kyoto, Japan, 2009.
- [335] Dandan Peng, Zhiliang Liu, Huan Wang, Yong Qin und Limin Jia. A novel deeper one-dimensional cnn with residual learning for fault diagnosis of wheelset bearings in high-speed trains. *Access J.*, 7:1022–1029, 2018.
- [336] Lucia Pepa, Giacomo Marangoni, Matteo Di Nicola, Lucio Ciabattoni, Federica Verdini, Luca Spalazzi und Sauro Longhi. Real time step length estimation on smartphone. In *Proc. Intl. Conf. Consumer Electronics (ICCE)*, Seiten 315–316. Bombay, India, 2016.
- [337] Juan Antonio Pérez-Ortiz, Felix A. Gers, Douglas Eck und Jürgen Schmidhuber. Kalman Filters improve LSTM Network performance in problems unsolvable by traditional Recurrent nets. *Neural Networks*, 16(2):241–250, 2003.
- [338] Arto Perttula, Helena Leppäkoski, Martti Kirkko-Jaakkola, Pavel Davidson, Jussi Collin und Jarmo Takala. Distributed indoor positioning system with inertial measurements and map matching. *IEEE Trans. on Instrumentation and Measurement*, 63(11):2682–2695, 2014.
- [339] James Pinchin, Chris Hide, Khari Abdulrahim, Terry Moore und Chris Hill. Integration of heading-aided MEMS IMU with GPS for

- pedestrian navigation. In *Proc. Intl. Conf. Satellite Division of the Institute of Navigation (ION GNSS)*, Seiten 1346–1356, Portland, OR, 2011.
- [340] Sylvain Pittet, Valerie Renaudin, Bertrand Merminod und Michel Kasser. UWB and MEMS based indoor navigation. *J. of Navigation*, 61(3):369, 2008.
- [341] Andreas Porada. Analyse und Evaluierung aktueller MR Tracking Systeme am Beispiel ARKit. Bachelor Thesis, Georg-Simon-Ohm Hochschule Nürnberg, 2018.
- [342] Andreas Porada. Feasibility Study of RNN-Supported Kalman Filters - On the Example of Radio Frequency Positions. Master Thesis, Technische Hochschule Nürnberg, 2021.
- [343] Mohsen Pourkhaatoun und Seyed A. Zekavat. *A review on TOA estimation techniques and comparison*. Wiley, Hoboken, New Jersey, 2011.
- [344] John Proakis. *Digital communications*. McGraw-Hill, Boston, 2008.
- [345] Honghui Qi und John B. Moore. Direct Kalman filtering approach for gps/ins integration. *IEEE Trans. on Aerospace and Electronic Systems*, 38(2):687–693, 2002.
- [346] Jiuchao Qian, Jiabin Ma, Rendong Ying, Peilin Liu und Ling Pei. An improved indoor localization method using smartphone inertial sensors. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation*, Seiten 1–7. Montbéliard, France, 2013.
- [347] Tong Qin, Peiliang Li und Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. on Robotics*, 34(4):1004–1020, 2018.
- [348] Anshul Rai, Krishna Kant Chintalapudi, Venkata N. Padmanabhan und Rijurekha Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *Proc. Intl. Conf. Mobile Computing and Networking*, Seiten 293–304. Istanbul, Turkey, 2012.
- [349] Milad Ramezani und Kourosh Khoshelham. Vehicle positioning in gnss-deprived urban areas by stereo visual-inertial odometry. *IEEE Trans. on Intelligent Vehicles*, 3(2):208–217, 2018.

- [350] Milad Ramezani, Kourosh Khoshelham und Laurent Kneip. Omnidirectional visual-inertial odometry using multi-state constraint Kalman filter. In *Proc. Intl. Conf. Intelligent Robots and Systems (IROS)*, Seiten 1317–1323. Vancouver, Canada, 2017.
- [351] Milad Ramezani, Kourosh Khoshelham und Clive Fraser. Pose estimation by omnidirectional visual-inertial odometry. *Robotics and Autonomous Systems*, 105(1):26–37, 2018.
- [352] Jouni Rantakokko, Joakim Rydell, Peter Strömbäck, Peter Händel, Jonas Callmer, David Törnqvist, Fredrik Gustafsson, Magnus Jobs und Mathias Grudén. Accurate and reliable soldier and first responder indoor positioning: multisensor systems and cooperative localization. *Wireless Communications*, 18(2):10–18, 2011.
- [353] Jesperi Rantanen, Maija Mäkelä, Laura Ruotsalainen und Martti Kirkko-Jaakkola. Motion context adaptive fusion of inertial and visual Pedestrian navigation. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 206–212. Nantes, France, 2018.
- [354] Carl Edward Rasmussen und Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. MIT Press, Cambridge, MA, 2005.
- [355] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore und Michael Littman. Activity recognition from accelerometer data. In *Proc. Intl. Conf. Innovative Appl. of Artif. Intelligence*, Seiten 1541–1546. Pittsburgh, PA, 2005.
- [356] Waseem Rawat und Zenghui Wang. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, 29(9):2352–2449, 2017.
- [357] Nils Reimers und Iryna Gurevych. Optimal hyperparameters for deep LSTM-networks for sequence labeling tasks, In arXiv:1707.06799 [CS], 2017.
- [358] Valerie Renaudin, Christophe Combettes und François Peyret. Quaternion based heading estimation with handheld MEMS in indoor environments. In *Proc. Intl. Conf. Position, Location and Navigation Symp. (PLANS)*, Seiten 645–656. Monterey, CA, 2014.

- [359] Valerie Renaudin, Miguel Ortiz und Julien Le Scornec. Foot-mounted pedestrian navigation reference with tightly coupled GNSS carrier phases, inertial and magnetic data. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 14–19. Sapporo, Japan, 2017.
- [360] Andreas Richter. *Estimation of radio channel parameters: Models and algorithms*. ISLE, Ilmenau, Germany, 2005.
- [361] Liu Rong, Zhou Jianzhong, Liu Ming und Hou Xiangfeng. A wearable acceleration sensor system for gait recognition. In *Proc. Intl. Conf. Industr. Electro. and Appl.*, Seiten 2654–2659. Harbin, China, 2007.
- [362] Daniel Roth, Constantin Kleinbeck, Tobias Feigl, Christopher Mutschler und Marc Erich Latoschik. Social Augmentations in Multi-User Virtual Reality: A Virtual Museum Experience. In *Proc. Intl. Symp. on Mixed and Augmented Reality (ISMAR)*, Seiten 42–43. Nantes, France, 2017.
- [363] Daniel Roth, Constantin Kleinbeck, Tobias Feigl, Christopher Mutschler und Marc-Erich Latoschik. Beyond Replication: Augmenting Social Behaviors in Multi-User Social Virtual Realities. In *Proc. Intl. Conf. Virtual Reality and 3D User Interfaces (IEEE VR)*, Seiten 215–222. Reutlingen, Germany, 2018.
- [364] Daniel Roth, Larissa Brübach, Franziska Westermeier, Christian Schell, Tobias Feigl und Marc Erich Latoschik. A Social Interaction Interface Supporting Affective Augmentation Based on Neuronal Data. In *Proc. Intl. Symp. on Spatial User Interaction (SUI)*, Seiten 1–4. New York, NY, 2019.
- [365] Daniel Roth, Franziska Westermeier, Larissa Brübach, Tobias Feigl, Christian Schell und Marc Erich Latoschik. Brain 2 Communicate: EEG-based Affect Recognition to Augment Virtual Social Interactions. In *Mensch und Computer 2019 - Workshopband*, Seiten 564–565. Hamburg, Germany, 2019.
- [366] Jana Rovňáková und Dušan Kocur. Toa estimation and data association for through-wall tracking of moving targets. *J. Wireless Communic. and Networking (EURASIP)*, 2010(1):1–11, 2010.

- [367] Nirupam Roy. *WalkCompass: Finding Walking Direction Leveraging Smartphone Inertial Sensors*. PhD thesis, Univ. of Shipbur, Columbia, SC, 2013.
- [368] Nirupam Roy, He Wang und Romit Roy Choudhury. I am a smartphone and I can tell my user's walking direction. In *Proc. Intl. Conf. Mobile Systems, Appl., and Services*, Seiten 329–342. Bretton Woods, NH, 2014.
- [369] Sebastian Ruder. An overview of gradient descent optimization algorithms, In arXiv:1609.04747 [cs.LG], 2017.
- [370] Antonio Ramón Jiménez Ruiz und Fernando Seco Granja. Comparing ubisense, bespoon, and decawave UWB location systems: Indoor performance analysis. *IEEE Trans. on Instrumentation and Measurement*, 66(8):2106–2117, 2017.
- [371] Antonio Ramón Jiménez Ruiz, Fernando Seco Granja, José Carlos Prieto Honorato und Jorge I. Guevara Rosas. Accurate pedestrian indoor navigation by tightly coupling foot-mounted IMU and RFID measurements. *IEEE Trans. on Instrumentation and Measurement*, 61(1):178–189, 2011.
- [372] David E. Rumelhart, Geoffrey E. Hinton und Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323 (6088):533–536, 1986.
- [373] Thomas A. Runkler. Datenvorverarbeitung. In *Data Mining*, Seiten 23–36. Springer Fachmedien Wiesbaden, 2015. doi: 10.1007/978-3-8348-2171-3_3.
- [374] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Zhiheng Huang Sean Ma, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg und Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *Intl. J. of Computer Vision*, 115(3):211–252, 2015.
- [375] Stuart J. Russell, Peter Norvig und Ernest Davis. *Artificial intelligence: a modern approach*. Prentice Hall series in artificial intelligence. Prentice Hall, 3rd edition, 2010.

Literaturverzeichnis

- [376] UkJae Ryu, Kyungho Ahn, Entae Kim, Munhae Kim, Boyeon Kim, Sunghun Woo und Yunseok Chang. Adaptive step detection algorithm for wireless smart step counter. In *Proc. Intl. Conf. Information Science and Applications (ICISA)*, Seiten 1–4. Pattaya, Thailand, 2013.
- [377] Angelo Sabatini. Quaternion-based extended Kalman Filter for determining orientation by inertial and magnetic sensing. *Trans. on Biomed. Eng.*, 53(7):1346–1356, 2006.
- [378] Rasoul Safavian und David Landgrebe. A survey of decision tree classifier methodology. *IEEE Trans. Systems, Man, and Cybernetics*, 21(3):83–124, 1991.
- [379] Andrew P. Sage und Gary W. Husa. Adaptive filtering with unknown prior statistics. In *Proc. Intl. Conf. Joint Automatic Control*, Seiten 760–769, Boulder, CO, 1969.
- [380] Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak und Shahrokh Valaee. Recent advances in Recurrent neural networks, In arXiv:1801.01078 [cs.NE], 2018.
- [381] Jussi Salmi, Andreas Richter und Visa Koivunen. Enhanced tracking of radio propagation path parameters using state-space modeling. In *Proc. Intl. Conf. European Signal Processing*, Seiten 1–5. Florence, Italy, 2006.
- [382] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas und Aleksander Madry. How Does Batch Normalization Help Optimization? In *Advances in Neural Information Processing Systems*, Seiten 2488–2498. Montréal, CA, 2018.
- [383] Simo Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.
- [384] Simo Särkkä, Aki Vehtari und Jouko Lampinen. Rao-blackwellized Particle Filter for multiple target tracking. *Information Fusion*, 8(1):2–15, 2007.
- [385] Jerzy Z. Sasiadek und Pande Hartana. Gps/ins sensor fusion for accurate positioning and navigation based on Kalman Filtering. *Intl Federation of Automatic Control (IFAC)*, 37(5):115–120, 2004.

- [386] Paul G. Savage. Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms. *Guidance, Control and Dynamics (GCD) J.*, 21(1):19–28, 1998.
- [387] Paul G. Savage. Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms. *J. of Guidance, Control, and Dynamics*, 21(2):208–221, 1998.
- [388] Abraham Savitzky und Marcel Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964.
- [389] Victor R. Schinazi, Tyler Thrash und Daniel-Robert Chebat. Spatial navigation by congenitally blind individuals. *WIREs Cognitive Science*, 7(1):37–58, 2016.
- [390] Thomas Schmid, Zainul Charbiwala, Jonathan Friedman, Young H. Cho und Mani B. Srivastava. Exploiting manufacturing variations for compensating environment-induced clock drift in time synchronization. *ACM SIGMETRICS Performance Evaluation Review*, 36(1):97, 2008.
- [391] Lukas Schmidt. How far is far? Evaluation, Visualization, and Interpretation of RNNs on Physically Correct Movements. Master Thesis, Friedrich-Alexander-University Erlangen-Nuremberg, 2019.
- [392] Mike Schuster und Kuldip K. Paliwal. Bidirectional Recurrent Neural networks. *Trans. on Signal Processing*, 45(11):2673–2681, 1997.
- [393] Sebastian Sczyslo, Jens Schroeder, Stefan Galler und Thomas Kaiser. Hybrid localization using UWB and inertial sensors. In *Proc. Intl. Conf. Ultra-Wideband*, Seiten 89–92, 2008.
- [394] Jochen Seitz, Lucila Patino-Studencka, Bernd Schindler, Stephan Haimerl, Boronat Javier Gutierrez, Steffen Meyer und Jörn Thielcke. Sensor Data Fusion for Pedestrian Navigation Using WLAN and INS. In *Proc. Intl. Symp. on Gyro Technology*, Seiten 11–16. Bonn, Germany, 2007.
- [395] Wenhua Shao, Haiyong Luo, Fang Zhao, Cong Wang, Antonio Crivello und Muhammad Zahid Tunio. Depedo: Anti periodic negative-step movement pedometer with deep convolutional

- neural networks. In *Proc. Intl. Conf. Communications*, Seite 395–401. Kansas City, MO, 2018.
- [396] Bai Shaojie, Kolter J. Zico und Koltun Vladlen. An empirical evaluation of generic convolutional and Recurrent networks for sequence modeling, In arXiv:1803.01271 [cs.LG], 2018.
- [397] Shirli Di-Castro Shashua und Shie Mannor. Deep robust Kalman Filter, In arXiv:1703.02310 [CS], 2017.
- [398] Hailong Shi, Hao Zhang und Xiqin Wang. A tdoa technique with super-resolution based on the volume cross-correlation function. *IEEE Trans. on Signal Processing*, 64(21):5682–5695, 2016.
- [399] Qin Shi, Sihao Zhao, Xiaowei Oui, Mingquan Lu und Mengdi Jia. Anchor self-localization algorithm based on UWB ranging and inertial measurements. *Tsinghua Science and Technology*, 24(6):728–737, 2019.
- [400] Seung Hyuck Shin und Chan Gook Park. Adaptive step length estimation algorithm using optimal parameters and movement status awareness. *Medical Engineering & Physics J.*, 33(9):1064–1071, 2011.
- [401] Andrei M. Shkel. Inertial MEMS sensors are becoming 3d and atomically precise. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–13. Nantes, France, 2018.
- [402] Ken Shoemake. Animating Rotation with Quaternion Curves. In *Proc. Intl. Conf. Computer Graphics and Interactive Techniques*, Seiten 245–254. San Francisco, CA, 1985.
- [403] Yuanchao. Shu, Kang G. Shin, Tian He und Jiming Chen. Last-mile navigation using smartphones. In *Proc. Intl. Conf. Mobile Computing (MobiCom)*, Seite 512–524. Berlin, Germany, 2015.
- [404] Robert H. Shumway. *Time series analysis and its applications: with R examples*. Springer Science+Business Media, 2017.
- [405] Prince Singh und Sunil Agrawal. Tdoa based node localization in wsn using Neural networks. In *Proc. Intl. Conf. Communications and Network Technologies*, Seiten 400–404. Gwalior, India, 2013.

- [406] Isaac Skog, Peter Handel, John-Olof Nilsson und Jouni Rantakokko. Zero-velocity detection—an algorithm evaluation. *Trans. on Biomedical Engineering*, 57(11):2657–2666, 2010.
- [407] Steven W. Smith. *The scientist and engineer’s guide to digital Signal Processing*. California Technical Pub., 1999. ISBN 978-0-9660176-3-2.
- [408] Philipp Sommer und Roger Wattenhofer. Gradient clock synchronization in wireless sensor networks. In *Proc. Intl. Conf. Information Processing in Sensor Networks (IPSN)*, Seiten 87–95. IEEE Computer Society, 2009.
- [409] Qingping Song und Rongke Liu. Weighted adaptive filtering algorithm for carrier tracking of deep space signal. *J. of Aeronautics*, 28(4):1236–1244, 2015.
- [410] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever und Ruslan Salakhutdinov. Dropout: a simple way to prevent Neural Networks from overfitting. *Machine Learning Research J.*, 15(1):1929–1958, 2014.
- [411] Maximilian Stahlke, Sebastian Kram, Thorbjørn Mumme und Jochen Seitz. Discrete positioning using UWB channel impulse responses and machine learning. In *Proc. Intl. Conf. Localization and GNSS (ICL-GNSS)*, Seiten 1–6. Nuremberg, Germany, 2019.
- [412] Mikhail Startsev, Ioannis Agtzidis und Michael Dorr. 1D CNN with BLSTM for automated classification of fixations, saccades, and smooth pursuits. *Behavior Research Methods J.*, 51(2):556–572, 2019.
- [413] Anthony Steed und Simon Julier. Behaviour-aware sensor fusion: Continuously inferring the alignment of coordinate systems from user behaviour. In *Proc. Intl. Symp. Mixed and Augmented Reality (ISMAR)*, Seiten 163–172. Adelaide, Australia, 2013.
- [414] Frank Steinicke, Gerd Bruder, Jason Jerald, Harald Frenz und Markus Lappe. Analyses of human sensitivity to redirected walking. In *Proc. Intl. Symp. on Virtual Reality Softw. and Techn.*, Seiten 149–156. Bordeaux, France, 2008.

- [415] Frank Steinicke, Gerd Bruder, Jason Jerald, Harald Frenz und Markus Lappe. Estimation of detection thresholds for redirected walking techniques. *IEEE Trans. Visualization and Computer Graphics*, 16(1):17–27, 2010.
- [416] Frank Steinicke, Yon Visell, Jennifer Campos und Anatole Lécuyer (Eds.). *Human walking in virtual environments: perception, technology, and applications*. Springer Berlin, New York, 2013.
- [417] Odongo Steven Eyobu und Dong Seog Han. Feature representation and data augmentation for human activity classification based on wearable IMU sensor data using a deep LSTM neural network. *Sensors J.*, 18(9):75–83, 2018.
- [418] Russell Stewart und Stefano Ermon. Label-free supervision of Neural Networks with physics and domain knowledge. In *Proc. Intl. Conf. Artificial Intelligence (AAAI)*, Seiten 187–196. San Francisco, CA, 2017.
- [419] Lucian Stoica, Alberto Rabbachin und Ian Oppermann. A low-complexity noncoherent IR-UWB transceiver architecture with TOA estimation. *Trans. on Microwave Theory and Techniques*, 54(4):1637–1646, 2006.
- [420] Hauke Strasdat, José M. M. Montiel und Andrew J. Davison. Visual SLAM: why filter? *Image and Vision Computing*, 30(2):65–77, 2012.
- [421] Michael Strohmeier, Thomas Walter, Julian Rothe und Sergio Montenegro. Ultra-wideband based pose estimation for small unmanned aerial vehicles. *IEEE Access*, 6(1):57526–57535, 2018.
- [422] Ilya Sutskever, James Martens, George E. Dahl und Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *Proc. Intl. Conf. Machine Learning (ICML)*, Seiten 1139–1147. Atlanta, GA, 2013.
- [423] Ilya Sutskever, Oriol Vinyals und Quoc V. Le. Sequence to sequence learning with neural networks. In *Proc. Intl. Conf. Advances in Neural Information Processing Systems*, Seiten 3104–3112. Montreal, Canada, 2014.
- [424] Li Tan. *Digital Signal Processing: fundamentals and applications*. Academic Press, 2008. ISBN 978-0-12-374090-8.

- [425] Makoto Tanigawa, Jeroen D. Hol, Fred Dijkstra, Henk Luinge und Per Slycke. Augmentation of low-cost GPS/MEMS INS with UWB positioning system for seamless outdoor/indoor positioning. In *Proc. Intl. Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS)*, Seiten 16–19. Savannah, GA, 2008.
- [426] Dante I. Tapia, Ricardo S. Alonso, Sara Rodriguez, Fernando de la Prieta, Juan M. Corchado und Javier Bajo. Implementing a real-time locating system based on wireless sensor Networks and artificial Neural Networks to mitigate the multipath effect. In *Proc. Intl. Conf. Information Fusion*, Seiten 1–8. Chicago, IL, 2011.
- [427] Sebastian Thrun, Wolfram Burgard und Dieter Fox. *Probabilistic Robotics*. Intelligent Robotics and autonomous agents. MIT Press, 2005.
- [428] Qinglin Tian, Zoran Salcic, Kevin Wang und Yun Pan. A multi-mode dead reckoning system for pedestrian tracking using smartphones. *Sensors J.*, 16(7):2079–2093, 2015.
- [429] Qinglin Tian, Kevin Wang und Zoran Salcic. Human body shadowing effect on UWB-based ranging system for pedestrian tracking. *IEEE Trans. on Instrumentation and Measurement*, 68(10):4028–4037, 2018.
- [430] Qinglin Tian, Kevin Wang und Zoran Salcic. A resetting approach for INS and UWB sensor fusion using Particle Filter for pedestrian tracking. *IEEE Trans. on Instrumentation and Measurement*, 7(3): 45–56, 2019.
- [431] Qinglin Tian, Kevin Wang und Zoran Salcic. A low-cost INS and UWB fusion pedestrian tracking system. *IEEE Sensors J.*, 19(10): 3733–3740, 2019.
- [432] Qinglin Tian, Kevin Wang und Zoran Salcic. An INS and UWB fusion-based gyroscope drift correction approach for indoor Pedestrian tracking. *Sensors J.*, 20(16):4476, 2020.
- [433] Qinglin Tian, Kevin Wang und Zoran Salcic. An INS and UWB fusion approach with adaptive ranging error mitigation for Pedestrian tracking. *IEEE Sensors J.*, 20(8):4372–4381, 2020.

Literaturverzeichnis

- [434] Janis Tiemann, Fabian Eckermann und Christian Wietfeld. AT-LAS - An open-source TDOA-based Ultra-wideband Localization System. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–6. Alcala de Henares, Spain, 2016.
- [435] David H. Titterton und John L. Weston. *Strapdown inertial navigation technology*. Number 17 in IEEE Radar, Sonar, Navigation, and Avionics Series. Institution of Electrical Engineers, 2nd edition, 2004. ISBN 978-0-86341-358-2.
- [436] Peerapong Torteeka, Xiu Chundi und Yang Dongkai. Hybrid technique for indoor positioning system based on Wi-Fi received Signal strength indication. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation*, Seiten 48–57. Busan, Korea, 2014.
- [437] Alexander Toshev und Christian Szegedy. DeepPose: Human pose estimation via deep neural networks. In *Proc. Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, Seiten 1653–1660. Columbus, OH, 2014.
- [438] Begumhan Turgut und Richard P Martin. Restarting Particle Filters: an approach to improve the performance of dynamic indoor localization. In *IEEE Proc. Intl. Conf. Global Telecommunications (GLOBECOM)*, Seiten 1–7. Honolulu, HI, 2009.
- [439] Markus Ulmschneider, Calvo D. Luz und Christian Gentner. Exchanging transmitter maps in multipath assisted positioning. In *Proc. Intl. Conf. Position, Location and Navigation Symposium (PLANS)*, Seiten 1020–1025. Monterey, CA, 2018.
- [440] Aris Valtazanos, Damal K. Arvind und Subramanian Ramamoorthy. Using wearable inertial sensors for posture and position tracking in unconstrained environments through learned translation manifolds. In *Proc. Intl. Conf. Information Processing in Sensor Networks (IPSN)*, Seiten 241–252. Philadelphia, PA, 2013.
- [441] Alle-Jan Van der Veen, Michaela C. Vanderveen und Arogyaswami Paulraj. Joint angle and delay estimation using shift-invariance techniques. *IEEE Trans. on Signal Processing*, 46(2): 405–418, 1998.
- [442] Kristof Van Laerhoven und Ozan Cakmakci. What shall we teach our pants? In *Proc. Intl. Symp. on Wearable Compu.*, Seiten 77–83. Atlanta, GA, 2000.

- [443] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser und Illia Polosukhin. Attention is all you need. In *Proc. Intl. Conf. Advances in Neural Information Processing Systems*, Seiten 5998–6008. Long Beach, CA, 2017.
- [444] Harshvardhan Vathsangam, Adar Emken, Donna Spruijt-Metz und Gaurav S. Sukhatme. Toward free-living walking speed estimation using gaussian process-based regression with on-body accelerometers and gyroscopes. In *Proc. Intl. Conf. Perv. Computing Technologies for Healthcare*, Seiten 22–31. Munich, Germany, 2010.
- [445] Josep Vidal und Rene Jativa. First arriving path detection for subscriber location in mobile communication systems. In *Proc. Intl. Conf. Acoustics, Speech, and Signal Processing*, Seiten 2721–2733. Orlando, FL, 2002.
- [446] Josep Vidal, Montse Nájar und René Játiva. High resolution time-of-arrival detection for wireless positioning systems. In *Proc. Intl. Conf. Vehicular Technology*, Seiten 2283–2287. Miami Beach, Florida, 2002.
- [447] Arun Vydhyanathan, Henk Luinge, Makoto Tanigawa, Fred Dijkstra, Michael S. Braasch und Maarten Uijt de Haag. Augmenting low-cost GPS/INS with ultra-wideband transceivers for multi-platform relative navigation. In *Proc. Intl. Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS)*, Seiten 547–554. Savannah, GA, 2009.
- [448] Brandon Wagstaff und Jonathan Kelly. LSTM-based zero-velocity detection for robust inertial navigation. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–8. Nantes, France, 2018.
- [449] Johan Wahlström, Isaac Skog, Fredrik Gustafsson, Andrew Markham und Niki Trigoni. Zero-velocity detection – a bayesian approach to adaptive thresholding. *Sensors J.*, 27(1):206–212, 2019.
- [450] Alexander H. Waibel, Toshiyuki Hanazawa, Geoffrey E. Hinton, Kiyohiro Shikano und Kevin J. Lang. Phoneme recognition using time-delay Neural networks. *Trans. on Acoustics, Speech, and Signal Processing*, 37(3):328–339, 1989.

Literaturverzeichnis

- [451] Simoes C. S. S. Walter, Yuri Motta L. R. Silva und de Lucena F. Jr. Vicente. A location technique based on hybrid data fusion used to increase the indoor location accuracy. *J. Computer Science*, 113(1):368–375, 2017.
- [452] Sheng Wan und Eric Foxlin. Improved pedestrian navigation based on drift-reduced MEMS IMU chip. In *Proc. Intl. Technical Meeting Institute of Navigation (ION)*, Seiten 220–229. San Diego, CA, 2010.
- [453] Haidong Wang, Li Cong und Honglei Qin. A real-time Pedestrian Dead Reckoning system with fm-aided motion mode recognition. *IEEE Sensors J.*, 19(8):3020–3032, 2018.
- [454] Jeen-Shing Wang, Che-Wei Lin, Ya-Ting C. Yang und Yu-Jen Ho. Walking pattern classification and walking distance estimation algorithms using gait phase information. *Trans. on Biomedical Engineering*, 59(10):2884–2892, 2012.
- [455] Jian Wang, Andong Hu, Chunyan Liu und Xin Li. A floor-map-aided WiFi/pseudo-odometry integration algorithm for an indoor positioning system. *Sensors J.*, 15(4):7096–7124, 2015.
- [456] Jian Wang, Yang Gao, Zengke Li, Xiaolin Meng und Craig M Hancock. A tightly-coupled GPS/INS/UWB cooperative positioning sensors system supported by v2i communication. *Sensors J.*, 16(7):944, 2016.
- [457] Lei Wang, Ruizhi Chen, Liang Chen, Lili Shen, Peng Zhang, Yuan-jin Pan und Ming Li. A robust Filter for TOA based indoor localization in mixed los/nlos environment. In *Proc. Intl. Conf. Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, Seiten 1–9. Wuhan, China, 2018.
- [458] Qu Wang, Langlang Ye, Haiyong Luo, Aidong Men, Fang Zhao und Yan Huang. Pedestrian stride-length estimation based on LSTM and denoising autoencoders. *Sensors J.*, 19(4):840, 2019.
- [459] Rui Wang, Celalettin Umit Bas, Zihang Cheng, Thomas Choi, Hao Feng, Zheda Li, Xiaokang Ye, Pan Tang, Seun Sangodoyin, Jorge Gómez-Ponce et al. Enabling super-resolution parameter estimation for mm-wave channel sounding. *IEEE Trans. on Wireless Communic.*, 19(5):3077–3090, 2020.

- [460] Sen Wang, Hongkai Wen, Ronald Clark und Niki Trigoni. Key-frame based large-scale indoor localisation using geomagnetic field and motion pattern. In *Proc. Intl. Conf. Intelligent Robots and Systems (IROS)*, Seiten 1910–1917. Daejeon, Korea, 2016.
- [461] Xiaoying Wang, Reza Hoseinnezhad, Amirali K. Gostar, Tharindu Rathnayake, Benlian Xu und Alireza Bab-Hadiashar. Multi-sensor control for multi-object BayesFilters. *Signal Processing*, 142(1):260–270, 2018.
- [462] Yan Wang und Xin Li. The IMU/UWB fusion positioning algorithm based on a Particle Filter. *Intl. J. of Geo-Information (ISPRS)*, 6(8):235, 2017.
- [463] Yan Wang und Xin Li. Graph-optimization-based ZUPT/UWB fusion algorithm. *Intl. Society for Photogrammetry and Remote Sensing (ISPRS)*, 7(1):18, 2018.
- [464] Yusheng Wang, Andrei Chernyshoff und Andrei M. Shkel. Error Analysis of ZUPT-Aided Pedestrian Inertial Navigation. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 206–212. Nantes, France, 2018.
- [465] Wasiq Waqar, Yuanzhu Chen und Andrew Vardy. Incorporating user motion information for indoor smartphone positioning in sparse wi-fi environments. In *Proc. Intl. Conf. Modeling, analysis and simulation of wireless and mobile systems*, Seiten 267–274. Montreal, Canada, 2014.
- [466] Mao Wei-Lung. GPS interference mitigation using derivative-free Kalman filter-based RNN. *Radioengineering*, 25(3):519, 2016.
- [467] Harvey Weinberg. Using the adxl202 in pedometer and personal navigation applications. *Analog Devices AN-602 application note*, 2(2):11–20, 2002.
- [468] Gary Weiss und Foster Provost. The effect of class distribution on classifier learning: an empirical study. *CiteSeer*, 11(1):48–63, 2001.
- [469] Greg Welch und Gary Bishop. An introduction to the Kalman Filter. *University of North Carolina at Chapel Hill*, 8(3):16, 2006.

Literaturverzeichnis

- [470] Kai Wen, Kegen Yu, Yingbing Li, Shubi Zhang und Wanwei Zhang. A new quaternion Kalman Filter based foot-mounted IMU and UWB tightly-coupled method for indoor Pedestrian navigation. *IEEE Trans. on Vehicular Technology*, 69(4):4340–4352, 2020.
- [471] Ronald J. Williams und Jing Peng. An efficient gradient-based algorithm for on-line training of Recurrent network trajectories. *Neural Computation*, 2(4):490–501, 1990.
- [472] Jens Windau und Laurent Itti. Walking compass with head-mounted IMU sensor. In *Proc. Intl. Conf. Robotics and Automation*, Seiten 5542–5547. Stockholm, Sweden, 2016.
- [473] Jürgen Winter und Christian Wengerter. High resolution estimation of the time of arrival for gsm location. In *Proc. Intl. Conf. Vehicular Technology*, Seiten 1343–1347. Boston, MA, 2000.
- [474] Wera Winterhalter, Freya Fleckenstein, Bastian Steder, Luciano Spinello und Wolfram Burgard. Accurate indoor localization for RGB-D smartphones and tablets given 2D floor plans. In *Proc. Intl. Conf. Intelligent Robots and Systems (IROS)*, Seiten 3138–3143. Hamburg, Germany, 2015.
- [475] Oliver J. Woodman. *An introduction to inertial navigation*. University of Cambridge. Computer Laboratory, 2007.
- [476] Ming Xia, Chundi Xiu, Dongkai Yang und Li Wang. A novel PDR aided UWB indoor positioning method. In *Proc. Intl. Conf. Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, Seiten 1–7. Wuhan, China, 2018.
- [477] Jiang Xiao, Zimu Zhou, Youwen Yi und Lionel M. Ni. A survey on wireless indoor localization from the device perspective. *ACM Computing Surveys*, 49(2):1–31, 2016.
- [478] Xuesu Xiao und Shuayb Zarar. Machine learning for placement-insensitive inertial motion capture. In *Proc. Intl. Conf. Robotics and Automation (ICRA)*, Seiten 716–721. Brisbane, AUD, 2018.
- [479] Zhuoling Xiao, Hongkai Wen, Andrew Markham und Niki Trigoni. Lightweight map matching for indoor localisation using conditional random fields. In *Proc. Intl. Symp. on Information Processing in Sensor Networks (IPSN)*, Seiten 131–142. Berlin, Germany, 2014.

- [480] Haifeng Xing, Jinglong Li, Bo Hou, Yongjian Zhang und Meifeng Guo. Pedestrian stride length estimation from IMU measurements and ann based algorithm. *Sensors J.*, 15(1):13–21, 2017.
- [481] Cheng Xu, Jie He, Xiaotong Zhang, Cui Yao und Po-Hsuan Tseng. Geometrical kinematic modeling on human motion using method of multi-sensor fusion. *Information Fusion*, 41(1):243–254, 2018.
- [482] Cheng Xu, Jie He, Yuanyuan Li, Xiaotong Zhang, Xinghang Zhou und Shihong Duan. Optimal estimation and fundamental limits for target localization using imu/toa fusion method. *IEEE Access*, 7(1):28124–28136, 2019.
- [483] Xiaosu Xu, Xinghua Liu, Beichen Zhao und Bo Yang. An extensible positioning system for locating mobile robots in unfamiliar environments. *Sensors J.*, 19(18):25–40, 2019.
- [484] Yuan Xu und Xiyuan Chen. Range-only UWB/INS tightly-coupled integrated navigation method for indoor pedestrian. *J. of Scientific Instrument*, 37(9):2115–2121, 2016.
- [485] Yuan Xu, Xiyuan Chen, Jin Cheng, Qinjun Zhao und Yimin Wang. Improving tightly-coupled model for indoor pedestrian navigation using foot-mounted IMU and UWB measurements. In *Proc. Intl. Conf. Instrumentation and Measurement Technology*, Seiten 1–5. Taipei, Taiwan, 2016.
- [486] Yuan Xu, Choon Ki Ahn, Yuriy S. Shmaliy, Xiyuan Chen und Yueyang Li. Adaptive robust INS/UWB-integrated human tracking using ufir Filter bank. *Measurement J.*, 123(1):1–7, 2018.
- [487] Yuan Xu, Tao Shen, Xi-Yuan Chen, Li-Li Bu und Ning Feng. Predictive adaptive Kalman Filter and its application to INS/UWB-integrated human localization with missing uwbt-based measurements. *Intl. J. of Automation and Computing*, 16(5):604–613, 2019.
- [488] Yuan Xu, Yuriy S. Shmaliy, Luchi Hua, Liyao Ma und Yuan Zhuang. Decision tree-extended FIR filtering for pedestrian tracking over tightly-integrated INS/UWB data. *Measurement Science and Technology*, 1(1), 2020.

- [489] Hang Yan, Qi Shan und Yasutaka Furukawa. Ridi: Robust IMU double integration. In *Proc. Intl. Conf. Computer Vision*, Seite 111–119. Munich, Germany, 2018.
- [490] Hang Yan, Sachini Herath und Yasutaka Furukawa. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, and new methods, In arXiv:1905.12853 [cs.CV], 2019.
- [491] Guanci Yang, Zhanjie Chen, Yang Li und Zhidong Su. Rapid relocation method for mobile robot based on improved ORB-SLAM₂ algorithm. *Remote Sensing*, 11(2):149, 2019.
- [492] Hai Yang, Rui Zhang, Joan Bordoy, Fabian Höflinger, Wei Li, Christian Schindelhauer und Leonhard Reindl. Smartphone-based indoor localization system using inertial sensor and acoustic transmitter/receiver. *IEEE Sensors J.*, 16(22):8051–8061, 2016.
- [493] Liuqing Yang und Georgios B. Giannakis. Ultra-wideband communications: an idea whose time has come. *IEEE Signal Processing Magazine*, 21(6):26–54, 2004.
- [494] Leehter Yao, Yeong-Wei Andy Wu, Lei Yao und Zhe Zheng Liao. An integrated IMU and UWB sensor based indoor positioning system. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–8. Sapporo, Japan, 2017.
- [495] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang und Tarek Abdelzaher. DeepSense: A unified deep learning framework for time-series mobile sensing data Processing. In *Proc. Intl. Conf. World Wide Web (WWW)*, Seiten 351–360. ACM Press, 2017.
- [496] Hong Ying, Carmen Silex, Andreas Schnitzer, Steffen Leonhardt und Michael Schiek. Automatic step detection in the accelerometer signal. In *Proc. Intl. Conf. Wearable and Implantable Body Sensor Networks (BSN)*, Seiten 80–85. Aachen, Germany, 2007.
- [497] Ma Yongtao, Wang Bobo, Pei Shuyang, Zhang Yunlei, Zhang Shuai und Yu Jiexiao. An Indoor Localization Method Based on AOA and PDOA Using Virtual Stations in Multipath and NLOS Environments for Passive UHF RFID. *IEEE Access*, 6:31772 – 31782, 2018.

- [498] Siamak Yousefi, Hirokazu Narui, Sankalp Dayal, Stefano Ermon und Shahrokh Valaei. A survey on behavior recognition using WiFi channel state information. *IEEE Communications Magazine*, 55(10):98–104, 2017.
- [499] Joe Youssef, Benoit Denis, Christelle Godin und Suzanne Les-ecq. Loosely-coupled ir-uwb handset and ankle-mounted inertial unit for indoor navigation. In *Proc. Intl. Conf. Ultra-Wideband (ICUWB)*, Seiten 160–164. Bologna, Italy, 2011.
- [500] Ning Yu, Yunfei Li, Xiaofeng Ma, Yinfeng Wu und Renjian Feng. Comparison of pedestrian tracking methods based on foot-and waist-mounted inertial sensors and handheld smartphones. *IEEE Sensors J.*, 19(18):8160–8173, 2019.
- [501] Risang Gatot Yudanto und Frederik Petre. Sensor fusion for indoor navigation and tracking of automated guided vehicles. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–8. Banff, Canada, 2015.
- [502] Mitchell Yuwono, Steven W. Su, Ying Guo, Bruce D. Moulton und Hung T. Nguyen. Unsupervised nonparametric method for gait analysis using a waist-worn inertial sensor. *Applied Soft Computing J.*, 14(12):72–80, 2014.
- [503] Faheem Zafari, Athanasios Gkelias und Kin K. Leung. A survey of indoor localization systems and technologies. *IEEE Communic. Surveys & Tutorials*, 21(3):2568–2599, 2019.
- [504] Sergey Zagoruyko und Nikos Komodakis. Wide residual networks, In arXiv:1605.07146 [cs.CV], 2017.
- [505] Francisco Zampella, Alessio De Angelis, Isaac Skog, Dave Zachariah und Antonio Jimenez. A constraint approach for UWB and PDR fusion. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Seiten 1–9. Sydney, Australia, 2012.
- [506] Francisco Zampella, Fernando Seco et al. Robust indoor positioning fusing PDR and RF technologies: The RFID and UWB case. In *Proc. Intl. Conf. Indoor Positioning and Indoor Navigation*, Seiten 1–10. Montbeliard, France, 2013.

Literaturverzeichnis

- [507] Tahmina Zebin, Patricia J. Scully, Niels Peek, Alexander J. Casson und Krikor B. Ozanyan. Design and implementation of a convolutional neural network on an edge computing smartphone for human activity recognition. *IEEE Access*, 7(5):133509–133520, 2019.
- [508] Zhuoqi Zeng, Steven Liu, Wei Wang und Lei Wang. Infrastructure-free indoor pedestrian tracking based on foot mounted UWB/IMU sensor fusion. In *Proc. Intl. Conf. Signal Processing and Communication Systems (ICSPCS)*, Seiten 1–7. Surfers Paradise, Australia, 2017.
- [509] Zhuoqi Zeng, Steven Liu und Lei Wang. A novel NLOS mitigation approach for TDOA based on IMU measurements. In *Proc. Intl. Conf. Wireless Communications and Networking (WCNC)*, Seiten 1–6. Barcelona, Spain, 2018.
- [510] Hui Zhang, Zonghua Zhang, Nan Gao, Yanjun Xiao, Zhaozong Meng und Zhen Li. Cost-effective wearable indoor localization and motion analysis via the integration of UWB and IMU. *Sensors J.*, 20(2):344, 2020.
- [511] Peng Zhang, Qile Zhao, You Li, Xiaoji Niu, Yuan Zhuang und Jingnan Liu. Collaborative WiFi fingerprinting using sensor-based navigation on smartphones. *Sensors J.*, 15(7):17534–17557, 2015.
- [512] Wei Zhang, Kan Liu, Weidong Zhang, Youmei Zhang und Jason Gu. Deep Neural Networks for wireless localization in indoor and outdoor environments. *Neurocomputing*, 194(2):279–287, 2016.
- [513] He Zhao und Zheyao Wang. Motion measurement using inertial sensors, ultrasonic sensors, and magnetometers with extended kalman filter for data fusion. *IEEE Sensors J.*, 12(5):943 – 953, 2012.
- [514] Shunyi Zhao, Biao Huang und Fei Liu. Localization of indoor mobile robot using minimum variance unbiased FIR filter. *IEEE Trans. on Automation Science and Engineering*, 15(2):410–419, 2016.
- [515] Yilin Zhao. Standardization of mobile phone positioning for 3g systems. *IEEE Communications Magazine*, 40(7):108–116, 2002.

- [516] Simeng Zhong, Kaiming Zhang, Guodong Zhu und Shuang Liu. Uwb-inertial fusion location algorithm based on Kalman Filtering. In *Proc. Intl. Conf. Control, Automation, Robotics and Vision (ICARCV)*, Seiten 574–578. Singapore, 2018.
- [517] Ying Zhong, Tianxia Liu, Bofeng Li, Ling Yang und Lizhi Lou. Integration of UWB and IMU for precise and continuous indoor positioning. In *Proc. Intl. Conf. Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, Seiten 1–5. Wuhan, China, 2018.
- [518] Hui Zhu, Bo Huang, Yuji Tanabe und Takaaki Baba. Local positioning with artificial neural network and time of arrival technique. In *Proc. Intl. Conf. Innovative Computing Information and Control*, Seiten 509–509. Dalian, China, 2008.
- [519] Yuan Zhuang und Naser El-Sheimy. Tightly-coupled integration of WiFi and MEMS sensors on handheld devices for indoor Pedestrian navigation. *IEEE Sensors J.*, 16(1):224–234, 2016.
- [520] Yuan Zhuang, You Li, Longning Qi, Haiyu Lan, Jun Yang und Naser El-Sheimy. A two-filter integration of MEMS sensors and WiFi fingerprinting for indoor positioning. *IEEE Sensors J.*, 16(13):5125–5126, 2016.
- [521] Shaghayegh Zihajehzadeh, Paul K. Yoon, Bong-Soo Kang und Edward J. Park. UWB-aided inertial motion capture for lower body 3-D dynamic activity and trajectory tracking. *IEEE Trans. on Instrumentation and Measurement*, 64(12):3577–3587, 2015.
- [522] Shaghayegh Zihajehzadeh, Paul K. Yoon und Edward J. Park. A magnetometer-free indoor human localization based on loosely coupled IMU/UWB fusion. In *Proc. Intl. Conf. Engineering in Medicine and Biology Society (EMBC)*, Seiten 3141–3144. Milano, Italy, 2015.
- [523] Tobias Zimmermann, Bertram Taetz und Gabriele Bleser. IMU-to-segment assignment and orientation alignment for the lower body using deep learning. *Sensors J.*, 18(1):302, 2018.
- [524] Hui Zou, Trevor Hastie und Robert Tibshirani. Sparse principal component analysis. *Computational and Graphical Statistics J.*, 15(2):265–286, 2006.

Literaturverzeichnis

- [525] Lukasz Zwirello, Christian Ascher, Gert F. Trommer und Thomas Zwick. Study on UWB/INS integration techniques. In *Proc. Intl. Workshop on Positioning, Navigation and Communication*, Seiten 13–17. Dresden Germany, 2011.
- [526] Lukasz Zwirello, Xuyang Li, Thomas Zwick, Christian Ascher, Sebastian Werling und Gert Trommer. Sensor data fusion in UWB-supported inertial navigation systems for indoor navigation. In *Proc. Intl. Conf. Robotics and Automation*, Seiten 3154–3159. Karlsruhe, Germany, 2013.
- [527] Alex Zyner, Stewart Worrall, James Ward und Eduardo Nebot. Long short term memory for driver intent prediction. In *Proc. Intl. Conf. Intelligent Vehicles Symp. (IV)*, Seiten 1484–1489. Redondo Beach, CA, 2017.
- [528] Bernt M. Åkesson, John Bagterp Jørgensen, Niels Kjølstad Poulsen und Sten Bay Jørgensen. A generalized autocovariance least-squares method for Kalman Filter tuning. *J. Process Control*, 18 (7):769 – 779, 2008.