

# Kooperative autonome Exploration in der Außenbereichsrobotik

DISSERTATION

zur Erlangung des Grades eines Doktors  
der Ingenieurwissenschaften

vorgelegt von

Dipl.-Inform. Lars Kuhnert

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät  
der Universität Siegen

Siegen 2015

Erster Gutachter: Univ.-Prof. Dr. rer. nat. Rainer Brück  
Zweiter Gutachter: Univ.-Prof. Dr.-Ing. Hubert Roth

Datum der mündlichen Prüfung: 28. April 2016

Für Bea und meine Eltern.



## Danksagungen

Zu Beginn meiner Promotion am Institut für Echtzeitlearnsysteme der Universität Siegen steckte die Forschung im Bereich der mobilen Außenbereichsrobotik noch in den Kinderschuhen. Heute bin ich rückblickend sehr froh und stolz, ein Teil der rasanten Entwicklung dieses Forschungsbereichs am EZLS gewesen sein zu dürfen und freue mich, dass ich meinen Beitrag dazu leisten konnte.

Mein Dank gilt allen Menschen, die diese Zeit zu einer extrem spannenden Achterbahnfahrt gemacht haben, die an ihren Hochpunkten mit gewonnen Wettbewerben, Forschungserfolgen und einem großartigen Institutsteam begeisterte, an ihren Tiefpunkten aber auch eine hohe Frustrationstoleranz bei allerlei Alltagskatastrophen im Umgang mit oft doch ziemlich eigenwilligen autonomen Robotern forderte.

Zunächst möchte ich besonders meinem Vater und Doktorvater Klaus-Dieter dafür danken, dass er er sich mit mir gemeinsam auf den Weg gemacht hat, meine Vision von autonom agierenden, kooperierenden Außenbereichsrobotern tatsächlich Realität werden zu lassen. Ich habe mich stets gleichermaßen unterstützt und gefordert gefühlt, was mich angetrieben hat, auch sehr komplexe Problemstellungen erfolgreich anzugehen, die auf den ersten Blick den Rahmen meiner Promotion klar gesprengt hätten. Unzählige, sehr interessante, aber nicht selten auch kontroverse Diskussionen mit ihm haben meine Arbeit erst zu der Qualität verholfen, die sie zum heutigen Zeitpunkt erreicht hat.

Weiterhin möchte ich Prof. Rainer Brück für die Übernahme des Korreferats und Prof. Madjid Fathi für den Vorsitz der Prüfungskommission herzlich danken.

Ich danke außerdem meinen ehemaligen Kollegen Duong, Ievgen, Jan, Jens, Khaled, Klaus, Marc-Steven, Martin, Matthias, Markus, Stefan, Tao und Wolfgang am EZLS, die mich in den letzten Jahren begleitet haben. Äußerst kommunikative und aufschlussreiche Mensa-Mittagsrunden, lange Abende der Fehlersuche und Reparaturmaßnahmen oder gemeinsame Minuten der Anspannung vor jedem Wettbewerbsstart haben mir sehr deutlich gezeigt, dass es im Berufsleben nicht nur darauf ankommt, welche Aufgabe zu bewältigen ist. Mindestens genauso wichtig ist, mit wem man diese Aufgabe bewältigen darf. Mein herzlicher Dank gilt allen Studenten, die meine Arbeit durch Diplom-, Master-, Studien- und Bachelorarbeiten oder als studentische Hilfskräfte un-

terstützt haben. Besonders möchte ich mich bei Benjamin, Johannes, Michael, Moritz und Simon bedanken, da ihr unermüdlicher Einsatz bei der Weiterentwicklung und Instandhaltung der Roboter-Familie des Instituts eine großartige Hilfe für meine Arbeit war und es erst möglich gemacht hat, Forschungsergebnisse im vorliegenden Umfang zu produzieren.

Ein großes Dankeschön gilt auch allen Freunden (besonders Vero), die mit unermüdlicher Ausdauer geholfen haben, diese Arbeit von allerlei formalen und sprachlichen Fehlern zu befreien.

Abschließend möchte ich den wichtigsten Menschen in meinem Leben für deren bedingungslose Unterstützung danken. Meine Eltern Doris und Klaus-Dieter sind und waren stets ein unumstößlicher Rückhalt in meinem Leben und haben mich unabhängig von zwischenzeitlichen Zweifeln immer darin bestätigt, dass ich auf dem richtigen Weg bin. Meiner Schwester Meike danke ich für ihre erfrischend andere Sicht auf das Leben, die viele Dinge in einem anderen Licht erscheinen lässt und mir immer wieder vor Augen geführt hat, wie wichtig es ist, dass man eine Sache tun sollte, weil man sich für sie begeistern kann, und nicht, weil sie irgendwie vernünftig erscheint. Mein größter Dank gilt meinem Sonnenschein und Fels in der Brandung Bea. Durch ihre lebensfrohe, begeisterte Art kann sie auch der anstrengendsten Situation immer etwas positiv Verrücktes abgewinnen und macht damit jeden Tag mit ihr zum Abenteuer. Ohne ihre Energie, ihr Verständnis und ihre Unterstützung hätte diese Arbeit nicht entstehen können, wofür ich zutiefst dankbar bin.

# Zusammenfassung

Autonome Exploration in der Außenbereichsrobotik ist eine komplexe Aufgabenstellung, die sich mit Navigation und Kartierung in vorab unbekanntem Außenbereichsarealen befasst. Motiviert durch die Möglichkeit, Gegenden automatisch zu erkunden, zu denen Menschen keinen Zugang haben oder deren äußere Umweltbedingungen ein zu großes Begehungsrisiko für Menschen darstellen, ist der Nutzen von Robotern, die Explorationsaufgaben bewältigen können, äußerst hoch einzuschätzen. Erkundung von Weltraum, Tiefsee oder Polarregionen sind nur einige ausgewählte prominente Anwendungsszenarien solcher Systeme. In semi- oder unstrukturiertem Gelände stellt eine große Vielfalt an Hindernisformen, Terraintypen und äußeren Störeinflüssen dabei jedoch eine erhebliche Herausforderung für robotische Explorationssysteme dar. Weiter erschwert wird die Aufgabe, wenn die Roboter während der Exploration vollständig autonom agieren sollen. Dabei sind eine Vielzahl an Themenkomplexen zu bearbeiten und schließlich in einem integrierten, robotischen System zusammenzuführen, die für sich genommen bereits eigene Forschungsfelder darstellen.

Die vorliegende Arbeit nimmt sich dieser umfangreichen Aufgabe an und beschreibt ein integriertes System zur Lösung des Problems der autonomen Exploration im unstrukturierten Außenbereichsszenario. Dabei wird ein Team aus Boden- und Flugroboter eingesetzt: AMOR und PSYCHE. Der Flugroboter folgt stets dem Bodenroboter und dient als externe Sensorplattform. Die so entstehende zusätzliche Betrachtung der aktuellen lokalen Szene aus einem fundamental anderen Blickwinkel ermöglicht vollkommen neue Lösungsansätze. Die wissenschaftlichen Innovationen in dieser Arbeit konzentrieren sich neben der Konzeption und Implementierung eines funktionierenden, realen Gesamtsystems auf drei essenzielle Teilkomponenten: Lokalisierung, Umgebungsmodellierung und Pfadplanung.

Für die Lokalisierung des Roboters im erdfesten Koordinatensystem werden zwei neue Verfahren eingeführt, die ohne die zwar etablierte, aber teilweise unzuverlässige GPS-Technologie auskommen. Zum einen wird ein probabilistischer Algorithmus vorgestellt, der eine Schätzung der aktuellen Pose des Roboters ausschließlich auf der Basis von Daten aus Koppelnavigation und

metrisch-topologischen Kartendaten ermittelt. Zum anderen wird eine Registrierung des Live-Luftbilds des kooperierenden Flugroboters mit einem georegistrierten Kartenausschnitt aus einer Kartendatenbank durchgeführt, um eine präzise Lokalisierung des im Live-Luftbild sichtbaren Bodenroboters zu erreichen.

Die Umgebungsmodellierung des Roboterteams wird durch Fusion von Daten aus multiplen, sowohl redundanten als auch komplementären, Sensorquellen erreicht, darunter mehrere aktiv bewegte Laserscanner und verschiedene Kameras. Das Ergebnis der Modellierung ist nicht wie häufig üblich eine Punktwolke, sondern ein Dreiecksnetz, das die geometrische Beschaffenheit der Umwelt nicht als ungeordnete Menge von Punkten, sondern als Oberfläche beschreibt. Zusätzlich angereichert wird das geometrische Modell mit visuellen Informationen. Dabei kommen Kameras mit entozentrischem und katadioptrischem Objektiv auf dem Bodenroboter sowie eine Luftbildkamera auf dem Flugroboter zum Einsatz, wobei die Nutzung von Luftbilddaten eines Flugroboters in Echtzeit eine innovative Erweiterung des Stands der Technik darstellt. Das resultierende Modell kann zur Lösung von Problemen herangezogen werden, die mit einem rein geometrischen Modell nicht lösbar gewesen wären, wie etwa die Unterscheidung ebener Teilflächen, die verschiedenen Untergrundformen entsprechen.

Bezogen auf die Pfadplanung eines autonomen Roboters bedeutet Exploration, dass kollisionsfreie Pfade geplant werden, die den Roboter aus der bekannten Umgebung ausbrechen lassen und ins Unbekannte führen. Aufgrund der meist vorherrschenden Start-Ziel-Planungsdirektive sind explorationsgetriebene Algorithmen jedoch unterrepräsentiert. Aus diesem Grund wird in dieser Arbeit eine neue Methode zur explorativen lokalen Pfadplanung auf der Basis von Rapidly Exploring Random Trees eingeführt. Dabei liegt das Hauptaugenmerk auf einer effizienten Ermittlung aller sogenannten Explorationspfade. Die explorative Pfadplanung wird dabei sowohl von klassischen geometrischen Merkmalen als auch von visuellen Merkmalen im zuvor registrierten Luftbild geleitet.

Die entwickelten Verfahren werden experimentell in einer ganzen Reihe von Situationen unter realen Bedingungen im Außenbereich mit dem vollständig autonom operierenden Roboterteam verifiziert und quantifiziert.

# Abstract

Autonomous exploration is a complex task for outdoor robots which is concerned with navigation in and mapping of previously unknown outdoor areas. Motivated by the possibility of automatically exploring parts of the world that are inaccessible to or too dangerous for humans, it has to be stated that the value of robots that can handle exploration tasks is very high. Space, deep sea or polar region exploration are just a few prominent of the many examples of such systems. Complex obstacle configurations and terrain types encountered in semi- or unstructured environments present significant challenges for robotic exploration systems. Demanding complete autonomy of the operating robots increases the difficulty of the task even further. A large variety of topics, each being a separate field of research, has to be dealt with, when building a fully integrated robotic system in this context.

This thesis proposes a solution to this complex undertaking and describes an integrated system, which is able to execute the task of autonomous exploration in an unstructured outdoor environment. A cooperating team consisting of a ground and an aerial robot, AMOR and PSYCHE, is designed to tackle the problem at hand. The aerial robot acts as an external sensor platform for the ground robot, while always following the ground robot. The additional and fundamentally different angle of view on the current scene provided by the aerial robot allows for completely new approaches to solving the posed problems. The scientific innovations presented in this work are focused on the three essential research topics: localization, environment modelling, and path planning. Additionally the conception and implementation of a fully functional, integrated robotic system is addressed.

Two new approaches for solving the task of localizing robotic systems in an earth fixed coordinate system are introduced, which do not build on the well-established yet sometimes unreliable GPS technology. The first approach a probabilistic algorithm for estimating the current pose of the ground robot is developed, which makes exclusive use of data from a dead-reckoning sensor and metric-topological map data. The second approach realizes a registration of live aerial imagery of the cooperating flying robot with a geo-registered orthophoto from a geo-database to achieve a precise localization of the ground robot by identifying its position in the live aerial image.

Environment modelling is achieved by fusing data of multiple redundant as

well as complementary sensors. Actuated laserscanners are used to create a description of the geometric structure of the environment in form of a triangle mesh, which is an advancement to the common approach of using unstructured point clouds for this task. The geometric model is additionally enhanced with visual information originating from different visual sensor sources. Beside cameras with entocentric and catadipotric lenses mounted on the ground robot, an airborne camera mounted on the aerial robot is used, which constitutes an innovative evolution of the current state-of-the-art in this context. The resulting model of the environment can be used to solve problems that could not be solved by using a strictly geometric model previously. Distinguishing between traversable and non-traversable planar surfaces is only one of many examples that underline the benefit of a visually enriched geometric model for outdoor robotics.

Path planning in the context of an exploring autonomous robot stands for the planning of collision-free paths that enable the robot to escape from the known environment and lead it into the unknown. Unfortunately algorithms focusing on exploration are underrepresented among the currently available path planning algorithms, which often focus on the classical concept of start-goal-planning. Due to this shortcoming a novel method for local explorative path planning based on the concept of Rapidly Exploring Random Trees is proposed in this thesis. It focuses on the efficient determination of all so-called exploration paths. The new method of local explorative path planning uses classical geometrical features as well as visual features to guide the path planning process.

The developed algorithms have been experimentally evaluated in several real-world situations with the fully autonomous robot team described in this thesis.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Ziel der Arbeit . . . . .	5
1.2	Wissenschaftlicher Beitrag . . . . .	6
1.2.1	Algorithmen . . . . .	7
1.2.2	Systemintegration . . . . .	11
1.3	Aufbau der Arbeit . . . . .	12
<b>2</b>	<b>Mobile Robotik im Außenbereich</b>	<b>13</b>
2.1	AMOR: Der Bodenroboter . . . . .	18
2.1.1	Aktorik . . . . .	19
2.1.2	Sensorik . . . . .	21
2.1.3	Sensor/Aktor Abstraktion . . . . .	25
2.2	Das Team: Unterstützung aus der Luft . . . . .	27
2.3	PSYCHE: Der Flugroboter . . . . .	29
<b>3</b>	<b>Systemarchitektur des Roboterteams</b>	<b>33</b>
3.1	Globale Komponenten . . . . .	35
3.1.1	Lokalisierung . . . . .	35
3.1.2	Pfadplanung und Navigation . . . . .	37
3.1.3	Schritthaltende Verfolgung des Bodenroboters durch den Flugroboter . . . . .	41
3.2	Lokale Komponenten . . . . .	44
3.2.1	Lokalisierung . . . . .	44
3.2.2	Umgebungsmodellierung . . . . .	47
3.2.3	Pfadplanung . . . . .	49
3.2.4	Pfadverfolgung . . . . .	51

3.2.5	Reaktive Hindernisvermeidung . . . . .	54
3.3	Missionsplanung . . . . .	57
<b>4</b>	<b>Lokalisierung im Außenbereich</b>	<b>61</b>
4.1	Odometrie und Koppelnavigation . . . . .	70
4.2	Positionsbestimmung bezüglich erdfester Koordinatensysteme .	73
4.3	Straßenkarten-basierte Lokalisierung . . . . .	74
4.3.1	Repräsentation von digitalen Wegenetzen . . . . .	77
4.3.2	Rekursive Bayes-Schätzung . . . . .	79
4.3.3	Grundprinzip der Partikelfilter-basierten Lokalisierung mit digitalen Kartendaten . . . . .	84
4.3.4	Entwurf des Partikelfilters . . . . .	85
4.3.5	Ergebnisse . . . . .	94
4.4	Lokalisierung durch Luft-Boden-Kooperation . . . . .	99
4.4.1	Ortho-Registrierung des Luftbildes . . . . .	103
4.4.2	Detektion des Bodenfahrzeugs im Live-Luftbild . . . . .	106
4.4.3	Ergebnisse . . . . .	108
<b>5</b>	<b>Umgebungsmodellierung durch Sensorfusion</b>	<b>111</b>
5.1	Lokaler Bezugspunkt: Die Szene . . . . .	113
5.2	Geometrische Beschreibung der Umgebung . . . . .	115
5.2.1	Sensorik . . . . .	115
5.2.2	Geometrische Messungen als Punktwolken . . . . .	118
5.2.3	Hierarchische Definition von Koordinatensystemen . . . .	120
5.2.4	Statische Transformationen im fahrzeugfesten Koordina- tensystem . . . . .	124
5.2.5	Dynamische Transformationen im szenenfesten Koordi- natensystem . . . . .	129
5.2.6	Variable Szenenkonstruktion aus dem kontinuierlichen Sensordatenstrom . . . . .	134
5.2.7	Oberflächenrekonstruktion aus Punktwolken . . . . .	135
5.3	Visuelle Beschreibung der Umgebung . . . . .	144
5.3.1	Sensorik . . . . .	145
5.3.2	Intrinsische Kalibrierung . . . . .	146
5.3.3	Extrinsische Kalibrierung . . . . .	155

5.3.4	Texturierung des geometrischen Oberflächenmodells . . .	165
5.4	Oberflächentexturierung mit Live-Luftbildern . . . . .	168
5.4.1	Posenbestimmung des Bodenroboters im Live-Luftbild .	170
5.4.2	Zeitliche Synchronisation bei bewegtem Bodenroboter . .	178
5.5	Ergebnisse . . . . .	180
<b>6</b>	<b>Lokale explorative Pfadplanung</b>	<b>193</b>
6.1	Rapidly Exploring Random Trees . . . . .	197
6.2	Lokale Hinderniskarte . . . . .	204
6.2.1	Geometrische Hindernisdetektion . . . . .	204
6.2.2	Fahrkorridorerkennung durch Luftbildauswertung . . . .	209
6.3	Spezialisiertes Baumwachstum . . . . .	214
6.3.1	Zustandsraum . . . . .	214
6.3.2	Zustandsübergangsfunktion . . . . .	214
6.3.3	Kollisionsprüfung . . . . .	216
6.3.4	Auswertung der Inputs . . . . .	220
6.3.5	Abstandsmetrik . . . . .	221
6.4	Pfadselektion . . . . .	221
6.4.1	Merkmale . . . . .	222
6.5	Ergebnisse . . . . .	227
<b>7</b>	<b>Fazit</b>	<b>235</b>



# Abbildungsverzeichnis

1.1	Beispiele für Exploration in der Außenbereichsrobotik. . . . .	2
1.2	Zusammenhang zwischen wichtigen Teilproblemen der mobilen Robotik. . . . .	4
2.1	Unterschiedliche Lokomotionsformen mobiler Roboter für den Einsatz im Außenbereichsszenario. . . . .	15
2.2	Robotische Radfahrzeuge für den Einsatz im Außenbereichsszenario. . . . .	17
2.3	Bodenroboter AMOR. . . . .	18
2.4	Modifizierte Aktorik-Komponenten des Bodenroboters AMOR. . . . .	20
2.5	Datenfluss bei der Ansteuerung der Aktorik-Komponenten. . . . .	21
2.6	Sensor-Ausstattung des Bodenroboters AMOR. . . . .	22
2.7	Abstraktion der Aktorik- und Sensorikkomponenten. . . . .	26
2.8	Das kooperierende Team, bestehend aus dem Bodenroboter AMOR und dem Flugroboter PSYCHE. . . . .	28
2.9	Flugroboter PSYCHE. . . . .	29
2.10	Systemkomponenten des Flugroboters PSYCHE. . . . .	31
3.1	Gesamtübersicht über die Abhängigkeiten zwischen den wichtigsten funktionalen Systemkomponenten des Multirobotersystems. . . . .	34
3.2	Datengrundlage der kartenbasierten globalen Planung. . . . .	38
3.3	Visualisierung der globalen Planungskomponente während einer Testfahrt. . . . .	39
3.4	Schritthaltende Verfolgung des Bodenroboters AMOR durch den Flugroboter PSYCHE in einer Außenbereichsszene. . . . .	42
3.5	Vorauselender Wegpunkt für die Navigation des Flugroboters. . . . .	43

3.6	Radschlupf bei einer autonomen Testfahrt des Roboters AMOR.	45
3.7	Vergleich der Lokalisierungsergebnisse bei klassischer Koppelnavigation bzw. visueller Odometrie. . . . .	46
3.8	Mit Luftbilddaten texturiertes Umgebungsmodell des Bodenroboters. . . . .	48
3.9	Schritte im Wachstum eines Rapidly Exploring Random Trees. .	50
3.10	Ergebnis der lokalen explorativen Pfadplanung in einer Außenbereichsszene. . . . .	51
3.11	Pfadverfolgung nach der <i>Pure Pursuit</i> -Methode. . . . .	53
3.12	Auswirkungen verschiedener Vorausschau-Distanzen auf das Fahrzeugverhalten. . . . .	53
3.13	Hinderniskarte der reaktiven Hindernisvermeidung. . . . .	55
3.14	Trajektorienanalyse in der reaktiven Hindernisvermeidung. . . .	56
3.15	Hierarchische Zustandsautomaten in Statechart-Notation mit beispielhaftem Zustandsübergang. . . . .	58
3.16	Hierarchische Zustandsautomaten zur Umsetzung der Mission: Autonome Navigation. Hauptmission. . . . .	59
3.17	Hierarchische Zustandsautomaten zur Umsetzung der Mission: Autonome Navigation. Untermission: Sackgassen-Behandlung. .	60
4.1	Koordinatensystem nach der ENU-Konvention. . . . .	64
4.2	Grundlegender Ablauf der rekursiven Bayes-Schätzung. . . . .	80
4.3	Iterationsschritte des kartenbasierten Partikelfilter-Algorithmus zur Lokalisierung eines mobilen Roboters. . . . .	84
4.4	Varianten bei der Initialisierung der Partikelmenge zu Beginn des Filterprozesses. . . . .	86
4.5	Lokaler Bewegungspuffer. . . . .	89
4.6	Linie-zu-Linie-Abgleich zwischen lokalem Bewegungspuffer und Wegenetz. . . . .	90
4.7	Schematische Darstellung der zwei möglichen Fälle beim Punkt-zu-Linie-Abgleich. . . . .	91
4.8	Partikelgewichtung unter Berücksichtigung des Bewegungspuffers.	92
4.9	Evaluationsergebnis der Partikelfilter-basierten Lokalisierung mit digitalen Kartendaten. . . . .	95
4.10	Beispielszene mit einer Sequenz zweier scharfer Abbiegevorgänge.	96

4.11	Beispielszene mit stark gestörten Koppelnavigationsdaten. . . . .	97
4.12	Beispielszene einer Y-Abzweigung. . . . .	98
4.13	Ablauf der kooperativen Luft-Boden-Lokalisierung. . . . .	102
4.14	Vorverarbeitungsschritte des Live-Luftbildes vor der Nutzung im Registrierungsprozess. . . . .	104
4.15	Ergebnis des Schablonenvergleichs. . . . .	105
4.16	Verarbeitungsschritte des Verfahrens zur Detektion des Boden- roboters im Live-Luftbild. . . . .	107
4.17	Ergebnisse der kooperativen Lokalisierung des Bodenroboters AMOR. . . . .	109
5.1	Beispiel einer Szene. . . . .	113
5.2	Punktwolke einer lokalen Szene. . . . .	115
5.3	Rotationsvarianten planarer Laserscanner zur Erzeugung von Punktvolken. . . . .	117
5.4	Zusammenhang zwischen räumlichen Polarkoordinaten und kar- tesischen Koordinaten. . . . .	117
5.5	Beispiel einer Punktwolke einer Außenbereichsszene. . . . .	119
5.6	Hierarchie der Koordinatensysteme für die Umgebungsrekon- struktion auf dem Roboter AMOR. . . . .	123
5.7	Hierarchie der Koordinatensysteme in Baumdarstellung. . . . .	124
5.8	Manuelle Bestimmung des Nick- und Rollwinkels des rückwärtigen Laserscanners. . . . .	127
5.9	Ergebnisse der experimentellen Bestimmung der Position des Rotationszentrums des Roboters. . . . .	130
5.10	Punktvolken zweier Laserscanner vor und nach der Registrierung. . . . .	131
5.11	Punktwolke als Ergebnis der Umgebungsmodellierung durch Fu- sion multipler Laserscanner. . . . .	133
5.12	Ablauf der variablen Szenenkonstruktion aus dem kontinuierlichen Sensordatenstrom. . . . .	134
5.13	Triangulationsschema für geordnete Punktwolke. . . . .	136
5.14	Beispiel der Triangulation einer geordneten Punktwolke. . . . .	137
5.15	Greedy-Triangulationsmethode . . . . .	143
5.16	Ergebnis der Triangulation einer ungeordneten Punktwolke. . . . .	144

5.17	Beispiele unterschiedlicher visueller Beschreibungen der gleichen Szene. . . . .	145
5.18	Koordinatentransformationen in einem katadioptrischen System mit hyperbolischem Spiegel. . . . .	152
5.19	Projektionsvarianten von Bilddaten einer omnidirektionalen Kamera. . . . .	154
5.20	Beispielszene mit manuell gewählten 2D/3D-Punktkorrespondenzen. . . . .	157
5.21	Probleme bei der manuellen Auswahl von Punktkorrespondenzen.	158
5.22	Künstliche Kalibrierobjekte für die präzise Ermittlung von 2D/3D-Punktkorrespondenzen. . . . .	159
5.23	Ebenenanpassung zur präzisen Bestimmung von Kalibrierpunkten auf einem künstlichen Kalibrierobjekt. . . . .	160
5.24	Beschreibung von Bildinformationen durch <i>Bearing-Vektoren</i> . . .	163
5.25	Vergleich unterschiedlicher Texturinterpolationsmethoden. . . .	167
5.26	Skizze der vereinfachenden Annahme des Zusammenhangs zwischen Luftbildkamera und Bodenebene bei nivellierender Kameraaufhängung. . . . .	169
5.27	Schablone für die Bestimmung der Pose des Bodenroboters im Luftbild. . . . .	170
5.28	Region-of-Interest für den Schablonenvergleich. . . . .	171
5.29	Visuelle Positionsermittlung der Region-of-Interest um den Bodenroboter. . . . .	174
5.30	Ergebnis des Schablonenvergleichs zur Bestimmung von Position und Orientierung des Bodenroboters. . . . .	176
5.31	Vergleich von Luftbild- und Szenenkoordinatensystem in einer Beispielszene. . . . .	179
5.32	Überblick einer Testfahrt des Roboterteams bestehend aus AMOR und PSYCHE zur Evaluation der entwickelten Methode zur kooperativen Umweltmodellierung. . . . .	182
5.33	Szene A: Ergebnisse der Kooperation Umweltmodellierung durch Sensorfusion. . . . .	185
5.34	Szene B: Ergebnisse der Kooperation Umweltmodellierung durch Sensorfusion. . . . .	186

5.35	Szene C: Ergebnisse der Kooperation Umweltmodellierung durch Sensorfusion. . . . .	187
5.36	Szene D: Ergebnisse der Kooperation Umweltmodellierung durch Sensorfusion. . . . .	188
5.37	Szene E: Ergebnisse der Kooperation Umweltmodellierung durch Sensorfusion. . . . .	189
5.38	Szene F: Ergebnisse der Kooperation Umweltmodellierung durch Sensorfusion. . . . .	190
5.39	Szene G: Ergebnisse der Kooperation Umweltmodellierung durch Sensorfusion. . . . .	191
5.40	Szene H: Ergebnisse der Kooperation Umweltmodellierung durch Sensorfusion. . . . .	192
6.1	Erweiterungsschritt beim Aufbau eines RRTs. . . . .	202
6.2	Vergleich verschiedener Methoden des zufälligen Baumwachstums.	203
6.3	Voronoi-Diagramme der Knoten des RRTs während unterschiedlicher Erweiterungsstadien des Planungsbaums. . . . .	203
6.4	Erzeugung einer Hindernisgitterkarte aus einer lokalen Punktwolke. . . . .	208
6.5	Unterschiedliche Untergrundtypen in einer Außenbereichsszene.	210
6.6	Segmentierungsergebnisse der visuellen Wegdetektion im Live-Luftbild. . . . .	212
6.7	Fusion der Ergebnisse aus geometrischer und visueller Hindernisdetektion. . . . .	213
6.8	Zellabdeckung des Bodenroboters bei verschiedenen Orientierungen. . . . .	216
6.9	Prüfungsreihenfolge abgedeckter Gitterzellen während der Kollisionsprüfung eines betrachteten Zustands. . . . .	217
6.10	Autonome Testfahrt des Roboters AMOR zur Evaluation der entwickelten Methode zur explorativen lokalen Pfadplanung. . .	228
6.11	Testfahrt-Szenen A bis E. . . . .	230
6.12	Testfahrt-Szenen F bis J. . . . .	231
6.13	Ergebnisse der Pfadplanung mit visueller Fahrkorridorerkennung im Live-Luftbild. . . . .	233



# Tabellenverzeichnis

2.1	Szenario-Eigenschafts-Matrix der mobilen Bodenrobotik im Innen- bzw. Außenbereich. . . . .	13
2.2	Vergleich von Boden- und Flugroboter im Außenbereichs-Szenario.	28
3.1	Vergleich verschiedener Methoden zur Pfadverfolgung anhand deren Einsatzszenarios. . . . .	52
5.1	Maximale Winkelfehler bei Annotation pro Scanzeile. . . . .	131



# Algorithmenverzeichnis

1	Globale Wegpunktnavigation . . . . .	40
2	Aufbau eines RRT . . . . .	201
3	Konstruktion der lokalen geometrischen Höhenkarte . . . . .	206
4	Ablauf der Prüfung auf Kollisionsfreiheit . . . . .	219



# Kapitel 1

## Einleitung

Exploration, also die Erforschung des örtlich Unbekannten, hat stets einen großen Reiz auf den Menschen ausgeübt und ihn dazu ermutigt, ambitionierte und mitunter auch sehr gefährliche Missionen zu unternehmen, um die Grenzen des Bekannten zu überschreiten. Roboter zu befähigen, dabei zu helfen, diese Aufgabe zu bewältigen, ist ein komplexes Unterfangen, das jedoch einen hohen Nutzen verspricht. Gerade in Szenarien, die gefährlich für Menschen sind oder eine präzise und gleichzeitig ausdauernde Leistung über lange Zeiträume hinweg fordern, ist der Einsatz von intelligenten Maschinen sehr naheliegend und wünschenswert. Durch große Fortschritte im Rahmen der mobilen Robotik in den letzten Jahren ist der Einsatz von Robotern für Explorationsaufgaben in den Bereich des Möglichen gerückt (siehe Abb. 1.1 für einige Beispiele aus der Robotikforschung). Dennoch sind aufgrund der Komplexität der Aufgabe und der damit zusammenhängenden Vielfältigkeit an zu lösenden Problemen verschiedenste Lösungsansätze vorstellbar.

Im Kontext der mobilen Robotik wird der Begriff der Exploration oftmals mit der Entsendung von Robotern auf andere Planeten für ausgedehnte Erkundungsmissionen verbunden. Dies entspricht jedoch keinem repräsentativen Teil der Forschungsbemühungen auf diesem Gebiet, denn auch andere Szenarien der mobilen Robotik legen eine intensive Beschäftigung mit dem Explorationsbegriff nahe. Gerade die robuste, autonome Ausführung von Explorationsaufgaben ist durch die Vielfalt der Unteraufgaben sehr komplex und bisher nicht uneingeschränkt gelöst.

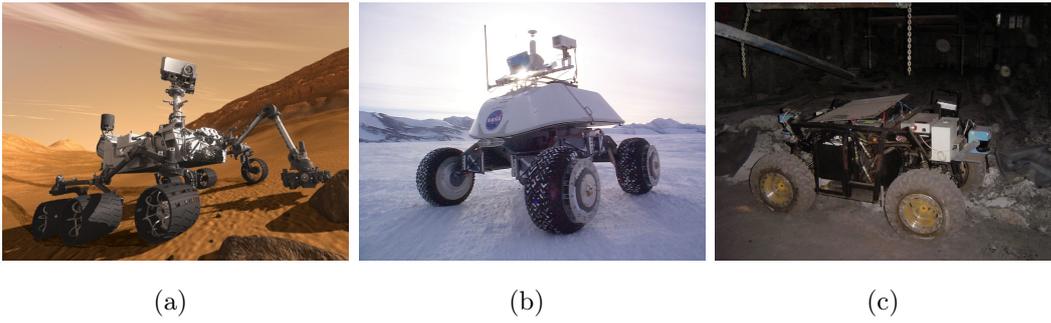


Abbildung 1.1: **Beispiele für Exploration in der Außenbereichsrobotik.** Mars Science Laboratory (Curiosity) (a) [69]. Nomad (b) [206]. Groundhog (c) [57][191].

Der Begriff der **Exploration** ist nicht einheitlich definiert und abhängig davon, in welchem fachlichen Kontext er verwendet wird. Daher wird an dieser Stelle eine Begriffsdefinition gewählt, welche die zentralen Aspekte der vorliegenden Arbeit einbindet und den Begriff auf den Forschungsbereich der Außenbereichsrobotik bezieht. Die Inspirationsquelle für die folgende Definition ist die weit verbreitete Basisdefinition des Explorationsbegriffs in der Robotik nach Yamauchi [212]:

We define exploration to be the act of moving through an unknown environment while building a map that can be used for subsequent navigation.

Exploration im Sinne der vorliegenden Arbeit wird davon ausgehend als Akt der autonomen Navigation in unbekannter Umgebung definiert, mit dem Ziel, die Umgebung mit einem möglichst realitätsgetreuen und detaillierten Modell zu beschreiben und somit die Unbekanntheit der Umwelt nachhaltig aufzulösen. Das Ergebnis, eine Karte bzw. ein Umgebungsmodell, stellt eine sehr wertvolle Datenbasis dar und kann sowohl von anderen Robotern als auch von menschlichen Benutzern eingesetzt werden, um die Umgebung zu analysieren oder zukünftige Missionen effizienter zu gestalten.

Beide Aspekte, d.h. sowohl die Navigation in unbekanntem, unstrukturiertem Gelände als auch die Erzeugung einer möglichst detailgetreuen Abbildung der Umwelt in Form eines Modells, sind dabei gleichwertig zu sehen, da sie nur in Kombination sinnvoll sind. Explorative Navigation als Selbstzweck ist ge-

nauso wenig sinnvoll wie Umgebungsmodelle eines größeren Areals erzeugen zu wollen, ohne sich autonom kollisionsfrei fortbewegen zu können. Daher sind beide Aspekte und deren komplexes Zusammenspiel in dieser Arbeit intensiv betrachtet worden.

Die Notwendigkeit der integrierten Betrachtung verschiedener Teilprobleme der mobilen Robotik wurde bereits im Jahr 2002 von Makarenko et al. [133] festgestellt. Abbildung 1.2 fasst diese Erkenntnisse in grafischer Form zusammen. Die zentralen Teilprobleme der mobilen Robotik sind gegliedert in die Problembereiche Lokalisierung, Kartierung und Bewegungskontrolle. Die Schnittmengen je zwei dieser Probleme stellen jeweils eigene spezialisierte Forschungsfelder dar. Integriert man Lokalisierung und Kartierung, entsteht das SLAM<sup>1</sup>-Problem [192], das sich mit der gleichzeitigen Ermittlung von Umgebungskarte und Pose eines Roboters ausschließlich aufgrund von Sensormessungen und Aktorikkommandos befasst. Werden Lokalisierung und Bewegungskontrolle in einem integrierten Problem bearbeitet, wird von aktiver Lokalisierung gesprochen. Ziel ist es hier, die Bewegung des Roboters bewusst so zu steuern, dass das Ausmaß der Unsicherheit des Lokalisierungsergebnisses minimiert wird (siehe bspw. [36]). Die Integration von Kartierung und Bewegungskontrolle in einem spezialisierten Teilproblem führt zur klassischen Explorationsproblematik. Wird zusätzlich der Teilbereich der Lokalisierung integriert, wird von der anspruchsvollsten Problemstellung, der integrierten Exploration (siehe bspw. [56]), gesprochen.

Ein wichtiges Unterscheidungskriterium ist weiterhin der Grad der Autonomie des entwickelten Systems. Dabei können grob vollautonome, teilautonome und teleoperierte Explorationsroboter unterschieden werden. Teleoperation benötigt die ständige Aufmerksamkeit eines menschlichen Benutzers, der über eine Funkverbindung die Daten der Sensoren des Roboters erhält und auf deren Basis Handlungsentscheidungen trifft und direkte Steuerkommandos an den Roboter sendet. Teilautonome Systeme können Unteraufgaben autonom ausführen. Sie unterstützen und entlasten den menschlichen Bediener somit bei der Steuerung. Ohne einen menschlichen Bediener kommt ein teilautonomes System jedoch ebenfalls nicht aus, obwohl die Kommandos bereits ein höheres Abstraktionsniveau aufweisen. Bei vollautonomen Systemen ist dies anders.

---

<sup>1</sup>Simultaneous Localization And Mapping

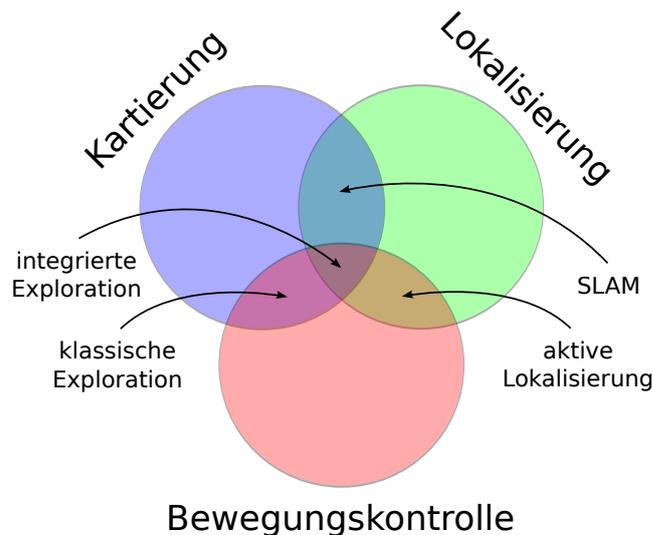


Abbildung 1.2: **Zusammenhang zwischen wichtigen Teilproblemen der mobilen Robotik.** Abbildung analog zu [133].

Der menschliche Bediener übermittelt lediglich einen Missionsplan auf hohem Abstraktionsniveau an den Roboter. Nach der Übermittlung ist der Roboter auf sich alleine gestellt und führt die Mission ohne Interaktion mit dem menschlichen Benutzer durch. Im Rahmen dieser Arbeit wurde ein vollautonomes System zur Exploration entwickelt.

Auch das Einsatzszenario beeinflusst die Anforderungen an einen Explorationsroboter stark. Methoden, die im Innenbereich problemlos funktionieren, müssen im Außenbereich oft vollständig neu überdacht werden. Und auch im Außenbereich selbst gibt es große Unterschiede zwischen dem Einsatz im urbanen, im ruralen oder im Offroad-Bereich. Unstrukturierte Umgebung, stark wechselnde äußere Störeinflüsse und fehlende Verfügbarkeit von etablierten Methoden zur Lokalisierung sind nur einige drängende Gründe für die Entwicklung neuartiger Methoden der robotischen Exploration im Außenbereich. Das Robotersystem, das in der vorliegenden Arbeit entwickelt wurde, ist für den Einsatz im Übergang vom ruralen zum Offroad-Bereich konzipiert. Sowohl mechanische Plattformen als auch eingesetzte Algorithmen sind so ausgelegt, dass abgesehen vom grundsätzlichen Vorhandensein eines befahrbaren Wegenetzes keinerlei Anforderungen an Beschaffenheit oder Strukturierung der Umgebung gestellt werden.

Ein weiterer entscheidender Punkt ist der Typ des oder der eingesetzten Ro-

boter. Neben Bodenrobotern, die mithilfe verschiedener Lokomotionsformen angetrieben werden, existieren in der Außenbereichsrobotik auch fliegende und schwimmende Systeme. Hybride Formen, wie amphibische Systeme, sind ebenfalls realisierbar. Genauso ist es möglich, verschiedene Robotertypen zu kombinieren, um ein Gesamtsystem zu realisieren, das von den komplementären Eigenschaften aller beteiligten Einzelsysteme profitiert. Dieser Idee folgend, wurde in dieser Arbeit ein kooperierendes Team aus Boden- und Luftroboter entwickelt.

## 1.1 Ziel der Arbeit

Das Ziel dieser Arbeit besteht darin, ein kooperierendes, autonom agierendes Team aus einem Bodenroboter und einem Flugroboter zu befähigen, Exploration eines zuvor vollständig unbekanntes Geländes durchzuführen. Das angestrebte Anwendungsszenario ist der Außenbereich und hier im Speziellen der rurale und der Offroad-Außenbereich, was einen deutlichen Unterschied bezüglich der Anforderungen an das entwickelte Robotersystem im Vergleich zur Fokussierung auf den Innenbereich oder den urbanen Außenbereich bedeutet. Einschränkend wird die Existenz eines grundsätzlich befahrbaren Wegenetzes vorausgesetzt, das sowohl befestigte als auch unbefestigte Wege beinhalten kann.

Das Ziel ist weiterhin, das zu erkundende Areal nachhaltig zu beschreiben. Das kann durch die detaillierte Kartierung der Umgebung in Form eines mit den Informationen verschiedener Sensoren angereicherten Umgebungsmodells erreicht werden. Diese hochqualitative Modellierung der Umgebung bietet eine sehr wertvolle Datenbasis für aktuelle und zukünftige Operationen des Roboterteams und seiner menschlichen Benutzer.

Um auch unter schwierigen Umgebungsbedingungen stets eine zuverlässige Lokalisierung des Roboterteams zu ermöglichen, werden alternative Lokalisierungsverfahren entwickelt, die speziell im Außenbereichsszenario eine robuste Positionierung der Roboter gewährleisten.

Neben der Entwicklung von störungsresistenten Methoden zur Selbstlokalisierung ist weiterhin anzustreben, dass die explorierenden Roboter stets verschie-

dene lokale Handlungsalternativen zur Auswahl gestellt bekommen, um eine kontinuierliche Exploration auch in unstrukturiertem Gelände mit komplexen Hinderniskonfigurationen sicherzustellen. Die Entwicklung der dafür benötigten lokalen Pfadplanungsalgorithmen stellt ebenfalls ein Ziel dieser Arbeit dar. Das übergeordnete Ziel, einen Flugroboter als unterstützende, intelligente, externe Sensorplattform für den Bodenroboter einzusetzen, manifestiert sich wiederholt in den zentralen Teilen der vorliegenden Arbeit. Die komplementären Sensorinformationen, die so dem Bodenroboter zur Verfügung stehen, erlauben die deutliche Optimierung der entwickelten Algorithmen und erhöhen damit die Leistung des Roboterteams während der autonomen Operation nachhaltig.

## 1.2 Wissenschaftlicher Beitrag

Der wissenschaftliche Beitrag dieser Arbeit gliedert sich zunächst in zwei Hauptaspekte, die wiederum einige Unteraspekte aufweisen. Auf der einen Seite wurden im Rahmen dieser Arbeit Algorithmen neu- bzw. weiterentwickelt, die komplexe Probleme adressieren und neuartige Lösungen zuführen, die bei der autonomen Exploration mit mobilen Außenbereichsrobotern auftreten. Dieser Teil stellt den zentralen wissenschaftlichen Beitrag dieser Arbeit dar und ist durch zahlreiche internationale Publikationen dokumentiert, die sich auf die folgenden Themenbereiche verteilen:

- Lokalisierung [105] [109],
- Umgebungsmodellierung und Perzeption [104] [110],
- Pfadplanung [106] und
- Kooperation [105] [108] [107].

Die hier aufgelisteten Publikationen stellen die Arbeiten mit engstem Bezug zur vorliegenden Arbeit dar und sind alle vom Autor der vorliegenden Arbeit als Erstautor veröffentlicht worden. Sie werden durch zahlreiche weitere internationale Veröffentlichungen des Autors im Bereich der mobilen Außenbereichsrobotik ergänzt, sodass der wissenschaftliche Beitrag dieser Arbeit äußerst nachhaltig und fundiert dokumentiert worden ist.

Auf der anderen Seite wurde zur realen Demonstration der neu entwickelten Methoden ein voll funktionsfähiges und integriertes Robotersystem geschaffen, das unter Wettbewerbsbedingungen in der Lage ist, seine autonomen Fähigkeiten robust zu demonstrieren. Der Aspekt der Systemintegration dieser Arbeit wird an dieser Stelle speziell herausgestellt, da dieser im wissenschaftlichen Diskurs der Robotikforschung oft in den Hintergrund tritt. Es wird mittlerweile zwar anerkannt, dass nur experimentelle Nachweise in diesem Zusammenhang gültig sind. Der große experimentelle Aufwand erscheint jedoch häufig nur indirekt in den zugehörigen Publikationen.

### **1.2.1 Algorithmen**

Die autonome Exploration im nichturbanen Außenbereichsszenario mit mobilen, robotischen Systemen bietet zahlreiche Herausforderungen, die bewältigt werden müssen, aber gleichzeitig die Möglichkeit bieten, neuartige, in anderen Szenarien schwer umsetzbare Konzepte zu realisieren. Beide Aspekte finden sich im wissenschaftlichen Beitrag dieser Arbeit auf algorithmischer Seite wieder. Die Nutzung eines autonomen Flugroboters als externe Sensorplattform für den Bodenroboter stellt dabei einen stets präsenten Innovationsaspekt der Arbeit dar, der wiederholt in den zentralen Komponenten dieser Arbeit bearbeitet wurde.

### **GPS-unabhängige Lokalisierung im erdfesten Koordinatensystem**

Die Abhängigkeit von der GPS-Technologie als Quelle der Lokalisierung im erdfesten Koordinatensystem ist in der Außenbereichsrobotik allgegenwärtig. Um diese Einschränkung zu überwinden, beispielsweise in Situationen, in denen keine oder nur gestörte GPS-Signale zur Verfügung stehen, werden in dieser Arbeit zwei neue Verfahren eingeführt, die eine globale Lokalisierung für den Bodenroboter auch ohne GPS erreichen.

Das erste Verfahren realisiert einen probabilistischen Ansatz zur Selbstlokalisierung auf der Basis von metrisch-annotierten, topologischen Kartendaten, wie sie in weit verbreiteten Geodatenbanken heutzutage zur Verfügung stehen. Ein Partikelfilter wird eingesetzt, um die Pose des Roboters im erdfesten

Koordinatensystem ausschließlich mit Kartendaten und relativen Messungen aus der Koppelnavigation zu schätzen. Neben der vollständigen Unabhängigkeit von der GPS-Technologie hebt sich diese Methode von den vorherigen Arbeiten bezüglich der folgenden neuen Merkmale ab: Erhöhte Konvergenzgeschwindigkeit durch äquidistante Initialisierung von Partikeln im Straßennetzwerk, neuartige Partikelgewichtung durch Vergleich von Kartendaten und lokalem Bewegungspuffer sowie optimierte Resampling-Strategie zur Verbesserung der Verknüpfung von Partikeln und Straßensegmenten. Während verschiedener ausgedehnter Tests der Methode wurde zusätzlich nachgewiesen, dass eine erfolgreiche Lokalisierung selbst bei fehlerhaften Kartendaten jederzeit robust möglich ist.

Für das zweite Verfahren kommt ein kooperierender Flugroboter als externe Sensorplattform für den Bodenroboter zum Einsatz. Auch hier wird die Verfügbarkeit von Daten in Geodatenbanken ausgenutzt, um in diesem Fall eine Korrespondenz zwischen georeferenzierten Orthofotos und Live-Luftbildern des Flugroboters herzustellen und damit die Pose des Bodenroboters im erdfesten Koordinatensystem festzustellen. Diese Methode zur Lokalisierung im Außenbereich war zum Zeitpunkt der Veröffentlichung neu und unterstreicht den Mehrwert einer Kooperation zwischen Boden- und Luftroboter in besonderem Maße. Die Nutzung eines autonomen Flugroboters als externe Sensorplattform für den Bodenroboter stellt dabei einen stets präsenten Innovationsaspekt der Arbeit dar und findet sich auch im folgenden Bereich der Umgebungsmodellierung durch Sensorfusion wieder.

### **Kooperative Umgebungsmodellierung durch Sensorfusion**

Der wissenschaftliche Beitrag dieser Arbeit im Themenbereich der Umgebungsmodellierung ist geprägt von mehreren aufeinander aufbauenden Aspekten. Zunächst wird durch Fusion mehrerer geometrischer Sensoren ein detailliertes geometrisches Modell der Umgebung erstellt. Für die Umsetzung dieser Aufgabe werden verschiedene Kalibrierungsverfahren entwickelt, die es ermöglichen, die Daten der geometrischen Sensoren bezüglich eines einheitlichen Fahrzeugkoordinatensystems auch bei sich bewegendem Roboter zu nutzen. Die Nutzung der stark einfedernden Fahrwerkseigenschaften der mechanischen Platt-

form während des Kalibrierungsprozesses stellt in diesem Zusammenhang einen ansonsten vernachlässigten Aspekt dar. Zusätzlich zu der üblichen Repräsentation der geometrischen Umgebungsdaten als Punktwolke wird ein Oberflächenmodell in Form eines Dreiecksnetzes für die Beschreibung der Umgebung des Roboters eingeführt. Dabei werden sowohl Verfahren für die Triangulation geordneter als auch ungeordneter Punktwolken entwickelt und eingesetzt.

Der Vorteil der oberflächenförmigen Modellierung der geometrischen Sensordaten im Vergleich zu einer rein punktförmigen Beschreibung zeigt sich im Besonderen im Ergebnis des darauf folgenden Fusionschritts, der die Daten der auf den eingesetzten Robotern montierten visuellen Sensoren in das Umgebungsmodell integriert. Auf dem Bodenroboter kommen dabei Kameras mit sowohl entozentrischem als auch katadioptrischem Objektiv zum Einsatz. Im Gegensatz zur sonst äußerst sensorspezifischen Modellierung für die extrinsische Kalibrierung der visuellen Sensorik wird in dieser Arbeit ein Ansatz eingeführt, der es ermöglicht, die extrinsische Kalibrierung für die eingesetzten Sensoren einheitlich auf der Basis von 2D/3D-Punktkorrespondenzen durchzuführen. Für die Ermittlung der dafür benötigten Punktkorrespondenzen wurde eine Methode zur hochpräzisen Bestimmung entwickelt. Spezielle, für diesen Zweck entworfene Kalibrierobjekte im Zusammenspiel mit entsprechenden Algorithmen ermöglichen eine hochgenaue Kalibrierung trotz beschränkter Auflösung der eingesetzten Sensorik. Das Ergebnis der Fusion von geometrischen und visuellen Sensordaten wird in Form eines texturierten Dreiecksnetzes modelliert. Dieses ermöglicht im Vergleich zur rein punktförmigen Darstellung alternativer Ansätze eine visuell deutlich realistischere Abbildung der Welt, da die visuelle Information nicht ausschließlich an den projizierten Positionen der punktförmigen geometrischen Sensormessungen genutzt, sondern über die gesamte triangulierte Oberfläche des geometrischen Umgebungsmodells verteilt wird.

Durch die Integration von Sensorinformationen, welche die Bewegung des Roboters abbilden, kann der Aufbau des fusionierten Umgebungsmodells sowohl bei stehendem als auch bei bewegtem Roboter geschehen. Der Aufgabe der Erzeugung des Umgebungsmodells bei sich bewegendem Roboter kommt dabei speziell beim nun folgenden letzten Aspekt des wissenschaftlichen Beitrags im Bereich der Umgebungsmodellierung besondere Bedeutung zu. Dieser besteht

in der Integration visueller Sensordaten eines kooperierenden Flugroboters, der dem Bodenroboter ein Luftbild der Szene, in der sich der Bodenroboter gerade bewegt, zur Verfügung stellt. Dies ist ein neuer, innovativer Ansatz, der in dieser Form in der mobilen Außenbereichsrobotik bisher noch nicht gewählt worden ist. Die besondere Schwierigkeit in diesem Kontext besteht in der schritthaltenden Bestimmung der Abbildung zur Transformation von Sensormessungen im Kamerakoordinatensystem des Flugroboters in das Fahrzeugkoordinatensystem des Bodenroboters. Diese komplexe Aufgabe wird aufgrund mangelnder Genauigkeit der verfügbaren Lokalisierungsverfahren der beteiligten Roboter durch Analyse der Luftbilddaten mit Methoden der digitalen Bildverarbeitung gelöst. Die fundamental andere Perspektive, aus der das Luftbild aufgezeichnet wurde, bietet komplementäre visuelle Sensorinformationen, welche die visuelle Qualität des texturierten geometrischen Umgebungsmodells deutlich steigern.

### **Explorative lokale Pfadplanung**

Die Entwicklung von lokalen Pfadplanungsalgorithmen für die Außenbereichsrobotik ist ein Bereich anhaltender Forschungsbemühungen. Gerade die Frage, wie Pfadplanungsalgorithmen zu entwerfen sind, die eine effiziente und dennoch erschöpfende Exploration einer unbekanntes lokalen Szene erreichen, ist im unstrukturierten Außenbereichsszenario nicht endgültig geklärt. Der Großteil der bestehenden Ansätze zur Pfadplanung basiert auf der klassischen Start-Ziel-Planungsdirektive, die für das Explorationsszenario ungeeignet ist. Diese Erkenntnis aufgreifend, besteht der wissenschaftliche Beitrag dieser Arbeit im Bereich der lokalen Pfadplanung in der Entwicklung einer vielseitig einsetzbaren lokalen Pfadplanungsmethode für das unstrukturierte Außenbereichsszenario, die eine Auswahl an Pfaden generiert, die das Entkommen aus einer lokalen Szene ermöglichen. Auf der Grundidee der Rapidly Exploring Random Trees [120] aufbauend, werden ausschließlich Pfade geplant, die von der genutzten Fahrzeugplattform auch tatsächlich, eingeschränkt von deren kinematischen Eigenschaften, umgesetzt werden können.

In diesem Zusammenhang spielt der Entwurf der Methode zur Kollisionsprüfung während des inkrementellen Planungsbaumwachstums eine entscheidende

Rolle. In dieser Arbeit wird in diesem Kontext zusätzlich zur üblichen, rein geometrisch basierten Analyse der Umgebung ein neuer Algorithmus eingeführt, der sowohl geometrische als auch visuelle Informationen zur Detektion des von einem Bodenroboter befahrbaren Bereichs integriert. Die visuellen Sensordaten stammen dabei von einem kooperierenden Flugroboter, was einen der zentralen Innovationsaspekte auch im Bereich der lokalen Pfadplanung erneut aufgreift.

### 1.2.2 Systemintegration

Wie bereits einleitend angedeutet, wurde im Rahmen der Entwicklungen neuer innovativer Algorithmen für die autonome Exploration im Außenbereich in dieser Arbeit großer Wert auf die Erprobung der erarbeiteten Methoden im realen Anwendungskontext gelegt. Zu diesem Zweck wurde begleitend zu den algorithmischen Neuentwicklungen ein voll funktionsfähiges, integriertes Roboterteam geschaffen, das aus dem **Bodenroboter AMOR** und dem **Flugroboter PSYCHE** besteht. Mithilfe dieses Roboterteams wurden die entwickelten Komponenten in zahlreichen realen Anwendungsszenarien getestet und evaluiert. Neben ausgedehnten Testfahrten im ruralen und Offroad-Bereich wurden die entwickelten Methoden mit großem Erfolg unter Wettbewerbsbedingungen internationaler Roboterwettbewerbe erprobt (u.a. die European Land ROBot trials (ELROB) [170] [169]).

Dieser Ansatz der Erprobung stellt eine deutlich komplexere, jedoch oft auch aufschlussreichere Variante der Evaluation im Vergleich zur sonst meist üblichen isolierten Analyse einzelner Komponenten dar. Viele problematische Situationen ergeben sich während der autonomen Operation der Roboter erst im Zusammenspiel aller am Robotersystem beteiligten Komponenten. Da sich diese Vorgehensweise - aufgrund des deutlich höheren Aufwands - in vergleichbaren Forschungsarbeiten mehrheitlich nicht wiederfindet, wird dieser Aspekt der vorliegenden Arbeit hier herausgestellt. Zusammenfassend bleibt festzustellen, dass die harte Erprobung der Algorithmen mit einem integrierten Robotersystem in realen Szenarien die Qualität der entwickelten Methoden erheblich gesteigert hat, was sich konsequenterweise auch in der Qualität des wissenschaftlichen Beitrags dieser Arbeit widerspiegelt.

### 1.3 Aufbau der Arbeit

Die vorliegende Arbeit gliedert sich in die folgenden Kapitel: Das einleitende Kapitel 1 bietet einen Einstieg in die Thematik der autonomen Exploration im Außenbereich und stellt heraus, welches Ziel mit dieser Arbeit verfolgt wird und welchen wissenschaftlichen Beitrag sie zur Thematik leistet. Kapitel 2 gibt einen kurzen Überblick über bestehende Robotersysteme in der Außenbereichsrobotik und beschreibt weiterhin detailliert Entwurf und Implementierung der in dieser Arbeit eingesetzten Fahrzeugplattformen AMOR und PSYCHE, wobei der zentrale Fokus auf den mechanischen Fahrzeugplattformen und deren Sensorik bzw. Aktorik liegt. Kapitel 3 beschreibt die wichtigsten funktionalen Komponenten der entwickelten Roboter, die benötigt werden, um autonome Exploration zu realisieren. Dieses Kapitel bietet damit zusätzlich die inhaltliche Einordnung der drei folgenden Kapitel, die sich detailliert mit den drei zentralen Themen dieser Arbeit auseinandersetzen. Zunächst wird in Kapitel 4 die Lokalisierung von Robotern im Außenbereich besprochen und in diesem Zusammenhang werden zwei neue Methoden zur Lokalisierung eines Roboters im erdfesten Koordinatensystem eingeführt. Kapitel 5 befasst sich mit der Modellierung der Umgebung durch Sensorfusion. Multiple, verteilte, heterogene Sensoren werden genutzt, um einheitlich ein möglichst realistisches und hochqualitatives Modell der Umwelt bei sich bewegenden Robotern zu erzeugen. Kapitel 6 schließlich behandelt das Thema der lokalen explorativen Pfadplanung. Es wird eine neue Methode eingeführt, die alle relevanten Pfade generiert, die einen Bodenroboter aus einer lokalen Szene entkommen lassen. Jeweils begleitend werden die Ergebnisse der in dieser Arbeit in den Kapiteln 4, 5 und 6 neu eingeführten Methoden anhand der Analyse von autonomen Testfahrten und daraus extrahierter Beispielszenen demonstriert und evaluiert. Das abschließende Kapitel 7 umfasst eine Rekapitulation der zentralen Methoden und Ergebnisse dieser Arbeit. Zusätzlich werden die vorgestellten Methoden im Sinne einer kritischen Reflexion bewertet und ein Ausblick auf zukünftige Forschungstätigkeiten basierend auf den Ergebnissen dieser Arbeit gegeben.

# Kapitel 2

## Mobile Robotik im Außenbereich

In der mobilen Bodenrobotik sind verschiedene Szenarien zu unterscheiden. Es wird zwischen **Innen-** und **Außenbereichsrobotik** differenziert; weiterhin wird die Außenbereichsrobotik in die drei Unterkategorien **urban**, **rural** und **offroad** unterteilt.

Szenario/Eigenschaft	Innenbereich	Außenbereich		
		urban	rural	offroad
Strukturierte Umgebung	ja	ja	nein	nein
Umgebungsbeschreibung (2D vs. 3D)	2D	2D-3D	3D	3D
Äußere Störfaktoren	nein	ja	ja	ja
Befahrbares Wegenetz	ja	ja	eingeschränkt <sup>1</sup>	nein
GPS-Verfügbarkeit	nein	ja <sup>2</sup>	ja <sup>3</sup>	ja <sup>3</sup>
Geländegängigkeit erforderlich	nein	nein	teilweise	ja

Tabelle 2.1: **Szenario-Eigenschafts-Matrix der mobilen Bodenrobotik im Innen- bzw. Außenbereich.** <sup>1</sup> Wege sind existent, aber oft schwer erkennbar oder mit natürlichen Hindernissen behaftet. <sup>2</sup> Teilweise eingeschränkt durch hohe Bebauung. <sup>3</sup> Teilweise eingeschränkt durch natürliche Hindernisse.

Tabelle 2.1 gibt in diesem Zusammenhang einen Überblick über die unterschiedlichen Szenarien und wichtige Charakteristika, welche im Rahmen dieser

Arbeit den größten Einfluss auf den Entwicklungsprozess gehabt haben. Diese Eigenschaften haben entscheidenden Einfluss auf das Design eines mobilen Robotersystems, da sich aus ihnen direkte Anforderungen ergeben. Ihre Berücksichtigung ermöglicht die erfolgreiche Entwicklung eines mobilen Bodenroboters, welcher in einem oder mehreren der genannten Szenarien einsetzbar ist.

Im Unterschied zur Innenbereichsrobotik ist die Außenbereichsrobotik durch eine deutlich weniger strukturierte bis vollständig unstrukturierte Umgebung charakterisiert, die sich durch hohe Komplexität und geringe Vorhersagbarkeit auszeichnet. Im Außenbereich gibt es zwar selbstverständlich ebenfalls strukturierte Elemente, jedoch lässt die Dichte dieser Elemente besonders dann stark nach, wenn man sich vom urbanen Bereich in den ruralen bzw. sogar den Offroad-Bereich bewegt. Je weniger die Umgebung strukturiert ist, desto höher sind die Anforderungen an den Entwurf eines autonomen mobilen Roboters. Im Folgenden soll kurz anhand einiger Beispiele aus der Forschung auf den Entwurf der mechanischen Plattform mit der speziellen Fragestellung der richtigen Wahl der Lokomotionsform eingegangen werden.

## **Lokomotionsformen**

Die eingesetzte Lokomotionsform für die Fortbewegung eines mobilen robotischen Systems hat entscheidenden Einfluss auf die Leistungsfähigkeit des Gesamtsystems. Wird eine nicht zum Einsatzszenario passende Lokomotionsform gewählt, kann dies zu häufigem Scheitern des Roboters während seiner Operation führen. Haben sich in strukturierter Umwelt im überwiegenden Anteil Fahrzeuge mit Radantrieb durchgesetzt, existiert in unstrukturierter Umgebung eine Vielzahl von Ansätzen, die sich erfolgreich unterschiedlichster Lokomotionsformen bedienen. Abbildung 2.1 zeigt eine Auswahl von Robotern mit unterschiedlichen Lokomotionsformen.

Auch im unstrukturierten Außenbereich existieren Fahrzeuge mit Radantrieb, die in den meisten Fällen jedoch darauf optimiert sind, eine gewisse Geländegängigkeit aufzuweisen (Abb. 2.1 (a)). Vorteilhaft bei diesem Ansatz sind die präzise Regelbarkeit der Fortbewegung und die breite Verfügbarkeit ausgereifter robuster Fahrzeugplattformen, die dieses Prinzip der Fortbewegung

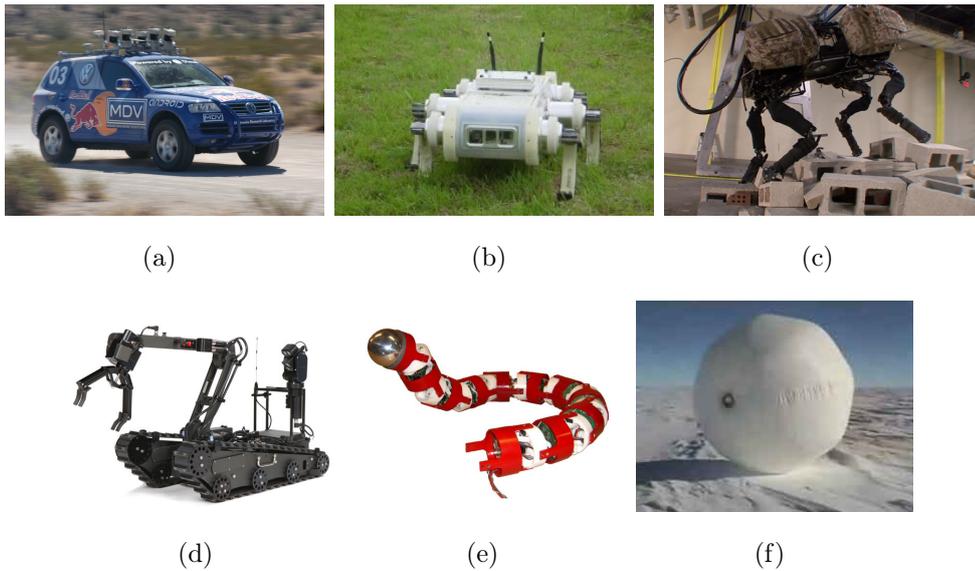


Abbildung 2.1: **Unterschiedliche Lokomotionsformen mobiler Roboter für den Einsatz im Außenbereichsszenario.** Stanley [193] (a). Rugged RHex [155] (b). BigDog [161] (c). Digital Vanguard ROV [44] (d). Aiko [197] (e). Tumbleweed [18] (f).

umsetzen. Es ist jedoch festzustellen, dass selbst geländegängige Fahrzeuge mit Radantrieb nur betrieben werden können, wenn (wenigstens minimal) befestigte Wege existieren. Antriebe, die ebenfalls um Fahrzeugachsen rotierende Komponenten besitzen, jedoch keine Radform aufweisen, orientieren sich an der Lokomotion kleiner Insekten (Abb. 2.1 (b)). Systeme dieser Art sind selbst in sehr unwegsamem Gelände fortbewegungsfähig. Allerdings ist der zu erwartende Pfad meist schlecht zu präzisieren. Laufende Plattformen, die sich durch koordinierte Einzelbeinbewegung fortbewegen, sind in unwegsamem Gelände ebenfalls äußerst mobil (Abb. 2.1 (c)). Bau und Regelung solcher Plattformen sind sehr aufwändig und stellen einen eigenständigen Forschungsbereich dar. Kettenfahrzeuge (Abb. 2.1 (d)) sind bei schwierigen Untergrundverhältnissen und extremen Steigungen Radfahrzeugen meist überlegen. Allerdings bedingt diese Lokomotionsform unruhigeres und unpräziseres Fahrverhalten des Roboters. Zusätzlich ist mit höherem Wartungsaufwand durch den Verschleiß der Ketten gerade bei höheren Geschwindigkeiten und einem deutlich gesteigerten Energiebedarf des Robotersystems zu rechnen. Aus dem Tierreich inspirierte Fortbewegungsformen, die beispielsweise die Lokomotion einer Schlange imitieren

ren (Abb. 2.1 (e)) sind für Spezialaufgaben konzipiert, wie etwa die Inspektion von schwer zugänglich Bereichen. Die Einsatzfähigkeit im Rahmen einer allgemeinen Explorationsaufgabe, bei der größere Strecken zurückgelegt werden sollen, ist nicht gegeben. Schließlich gibt es noch Ansätze für die Konstruktion von Außenbereichsrobotern, die keine aktive Form der Fortbewegung vorsehen. Der in Abbildung 2.1 (f) gezeigte kugelförmige Roboter bewegt sich stets in die Richtung, in die er durch die natürlichen Windkräfte gedrängt wird. Damit ist ein solcher Ansatz für die zielgerichtete Exploration ungeeignet.

Die vorangehenden Überlegungen zu den Unterschieden zwischen verschiedenen Lokomotionsformen in der Außenbereichsrobotik haben die Wahl einer Fahrzeugplattform mit Radantrieb für den Roboter in dieser Arbeit zur Folge gehabt. Es wird davon ausgegangen, dass sich der Roboter zwar in unstrukturierter Umgebung, jedoch auf einem (wenigstens minimal) befestigten Wegenetz bewegt. Weiterhin sollen moderate Bewegungsgeschwindigkeiten realisiert werden, wobei die Steuerung des Roboters nicht hauptsächlich reaktiv erfolgen soll. Dies erfordert eine höhere Ausführungspräzision von geplanten Pfaden. Schließlich ist die kostengünstige, breite Verfügbarkeit von robusten Radfahrzeugen für den Außenbereich ein weiteres wichtiges Entscheidungskriterium, da so die komplexe und langwierige Eigenkonstruktion einer mechanischen Fahrzeugplattform vermieden wird.

## **Robotische Radfahrzeuge**

Da in dieser Arbeit ein robotisches Radfahrzeug als mechanische Fahrzeugplattform gewählt wurde, folgt an dieser Stelle eine kurze Übersicht über einen Teil der sehr zahlreichen Forschungsfahrzeuge, in welchen ebenfalls dieser Ansatz gewählt wurde (siehe Abb. 2.2). Die Fahrzeugplattformen gliedern sich dabei grob in die drei folgenden Kategorien: geländegängige Straßenfahrzeuge, dedizierte Offroad-Fahrzeuge und vollständige Fahrzeugneukonstruktionen. Im folgenden Kapitel wird nun detailliert beschrieben, welche Designphilosophie für die Roboterfahrzeuge zugrunde gelegt wurde, die im Rahmen dieser Arbeit zum Einsatz gekommen sind.



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)



(j)



(k)



(l)

Abbildung 2.2: **Robotische Radfahrzeuge für den Einsatz im Außenbereichsszenario.** Stanley [193] (a). Sandstorm [199] (b). Boss [200] (c). Lama [87] (d). Demo III XUV [113] (e). Crusher [181] (f). LAGR [88] (g). Grizzly [164] (h). Mörri [194] (i). MuCAR-3 [79] (j). Hanna [116] (k). Ravon [6] (l).

## 2.1 AMOR: Der Bodenroboter

Nach Analyse der oben genannten Beispiele haben sich einige grundlegende Designprinzipien für die Entwicklung des **Bodenroboters AMOR** herauskristallisiert. Zunächst ist die Wahl der mechanischen Plattform als Basis für den Bodenroboter auf ein bewährtes Massenprodukt gefallen - ein benzinbetriebenes Quad vom Typ **Kodiak YFM 400** des Herstellers **Yamaha** (siehe Abb. 2.3).



Abbildung 2.3: **Bodenroboter AMOR** vor (a) und nach (b) Umbau der mechanischen Plattform.

Diese Wahl begründet sich durch mehrere Faktoren. Die Wahl eines erprobten, in Masse produzierten Fahrzeugs bedeutet, dass viele Fehler im iterativen Entwicklungsprozess durch den professionellen, darauf spezialisierten Fahrzeughersteller schon behoben worden sind. Fehler, die bei speziell für die Forschung in geringen Stückzahlen produzierten Fahrzeugen oder kompletten Eigenbau-Fahrzeugen mit größerer Wahrscheinlichkeit auftreten, können so schon bei der Fahrzeugwahl vermieden werden. Dies ist der Robustheit des Robotersystems und damit der Wartungsfreundlichkeit sehr zuträglich. Die Wartung beschränkt sich somit hauptsächlich auf die baulichen Veränderungen, die nötig sind, um das Fahrzeug mit einem Computer steuern zu können (siehe Kapitel 2.1.1). So kann der Entwicklungsfokus auf die Realisierung der autonomen Fähigkeiten des Roboters gelegt werden, ohne zu viel Zeit in die Entwicklung der mechanischen Plattform zu investieren. Zweitens ist die Geländegängigkeit für das in dieser Arbeit anvisierte Szenario vollständig gegeben. Auch schwierige

Untergrundverhältnisse, kleine bis mittlere Hindernisse und starke Steigungen können von der gewählten mechanischen Fahrzeugbasis bewältigt werden. Der benzinbetriebene Antrieb ist in der Lage, große Reichweiten ( $> 300$  km) des Roboters zu realisieren und sorgt gleichzeitig für eine ausreichende Stromversorgung der zusätzlichen Bauteile, die auf dem Roboter installiert werden müssen. In diesem Zusammenhang ist ebenfalls die Nutzlast des Fahrzeugs hervorzuheben, die groß genug ist, um sowohl eine große Anzahl verschiedener Sensoren als auch die entsprechende Elektronik zur Realisierung der gewünschten autonomen Fähigkeiten des Roboters unterzubringen.

### 2.1.1 Aktorik

Da die Aktorik der eingesetzten Fahrzeugplattform für die Benutzung durch einen menschlichen Bediener und damit nicht für den autonomen Betrieb, also die Steuerung durch einen Computer, ausgelegt ist, wurden Gas- und Bremszug, Lenkung sowie Gangschaltung so modifiziert, dass sie mit einem Computer angesteuert und bedient werden können. Abb. 2.4 zeigt die modifizierten Aktoren des Fahrzeugs nach dem Umbau.

Der Gaszug wird durch einen Servomotor bedient. Auf der Lenkstange wurden zwei Zahnräder montiert, die über Keilriemen mit je einem Elektromotor verbunden sind und so im Zusammenspiel mit einem ebenfalls auf der Lenkstange montierten Potentiometer die Einstellung des Lenkwinkels erlauben. Die Bremsen und die Gangschaltung werden durch pneumatische Komponenten bedient. Ein Pneumatikzylinder drückt auf die zuvor vom Hand-Bremshebel betätigte Schnittstelle zum hydraulischen Bremssystem des Fahrzeugs, welches ansonsten im Originalzustand belassen wurde. Ebenfalls ein Pneumatikzylinder bedient über ein Gestänge den Gangschaltungshebel des Fahrzeugs, an welchem auch keine weiteren Veränderungen vorgenommen werden mussten. Damit sind alle relevanten Aktorik-Komponenten über einen Computer bedienbar und somit ist das Fahrzeug für den autonomen Betrieb vorbereitet.

Um die Benutzung der Aktorik auf Software-Ebene möglichst einfach und sicher zu gestalten, ist die Ansteuerung der Aktorik so realisiert worden, dass alle Kommandos mit SI-Größen arbeiten. Das heißt, dass die Geschwindigkeit in Meter pro Sekunde, der Lenkwinkel in Radiant sowie der Gang (vorwärts

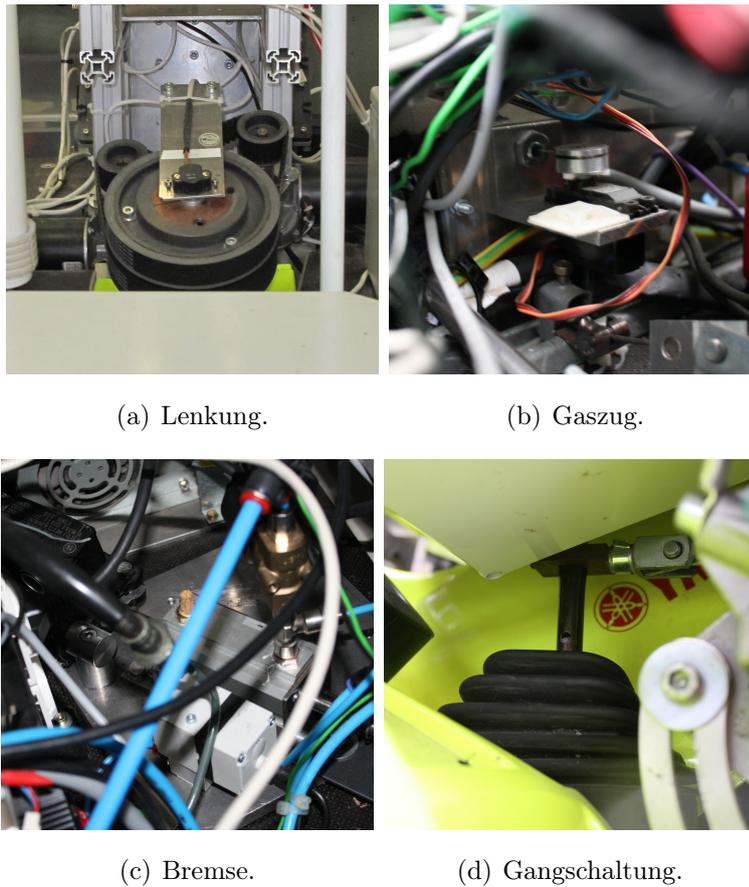


Abbildung 2.4: **Modifizierte Aktorik-Komponenten des Bodenroboters AMOR.**

oder rückwärts) als gewünschte Sollwerte kommandiert werden können. Diese Sollwerte werden dann zunächst auf Plausibilität geprüft und an entsprechende Regler für Gas, Bremse und Lenkung weitergegeben. Diese Regler setzen die Sollwerte in entsprechende, zur Aktor-Hardware passende Low-Level-Kommandos um. Zusätzliche Sicherheit der Aktorik-Benutzung wird durch einen Totmannschalter realisiert. Es wird gefordert, dass eine minimale Frequenz für die Kommandierung von Aktorik-Sollwerten eingehalten wird. Wird diese Frequenz unterschritten, wird davon ausgegangen, dass die autonome Steuerung des Roboters nicht korrekt funktioniert. In diesem Fall wird der Roboter aus Sicherheitsgründen augenblicklich gestoppt. Abb. 2.5 zeigt den Ablauf der Ansteuerung der Aktoren im Überblick.

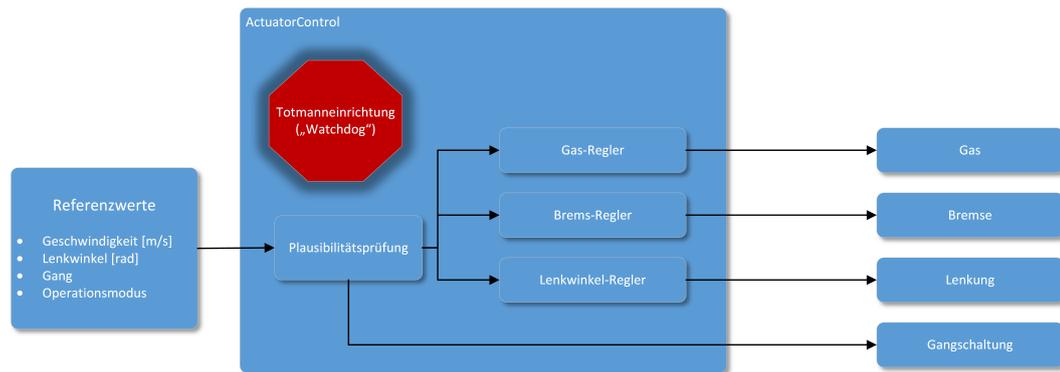


Abbildung 2.5: **Datenfluss bei der Ansteuerung der Aktorik-Komponenten.**

### 2.1.2 Sensorik

Die Auswahl der sensorischen Ausstattung des Bodenroboters AMOR gründet auf dem Streben nach einem hohen Grad an Redundanz bezüglich der eingesetzten Sensorik. So können messprinzipbedingte Schwächen der einzelnen Sensorsysteme kompensiert sowie Messfehler im Allgemeinen abgeschwächt werden. Dies ist gerade beim Einsatz im Außenbereich wichtig, da hier mit unterschiedlichsten äußeren Störquellen zu rechnen ist.

Abb. 2.6 zeigt eine Übersicht aller auf dem Roboter verbauten Sensoren. Zur Erfassung visueller Informationen sind unterschiedliche Kameras auf dem Roboter montiert. Eine Farbkamera erfasst den Bereich vor dem Roboter. Sie ist mit einem besonders weitwinkligen Objektiv ausgestattet, um einen möglichst großen Bereich der Umgebung vor dem Fahrzeug abzudecken. Oben auf dem Roboter ist eine Kamera mit katadioptrischem Objektiv angebracht, die durch die Nutzung eines konisch geformten Spiegels eine omnidirektionale Sicht auf die Umgebung rund um den Roboter erlaubt. Ergänzend zu diesen Kameras ist ein Stereokamera-System im unteren vorderen Bereich des Roboters angebracht, welches sowohl die Erzeugung von visuellen als auch geometrischen Informationen erlaubt.

Drei Laserscanner auf dem Fahrzeug dienen der geometrischen Vermessung der Umgebung des Roboters. Sie generieren jedoch jeweils ausschließlich einen Ebenenschnitt durch die geometrischen Strukturen in der Umgebung des Roboters.

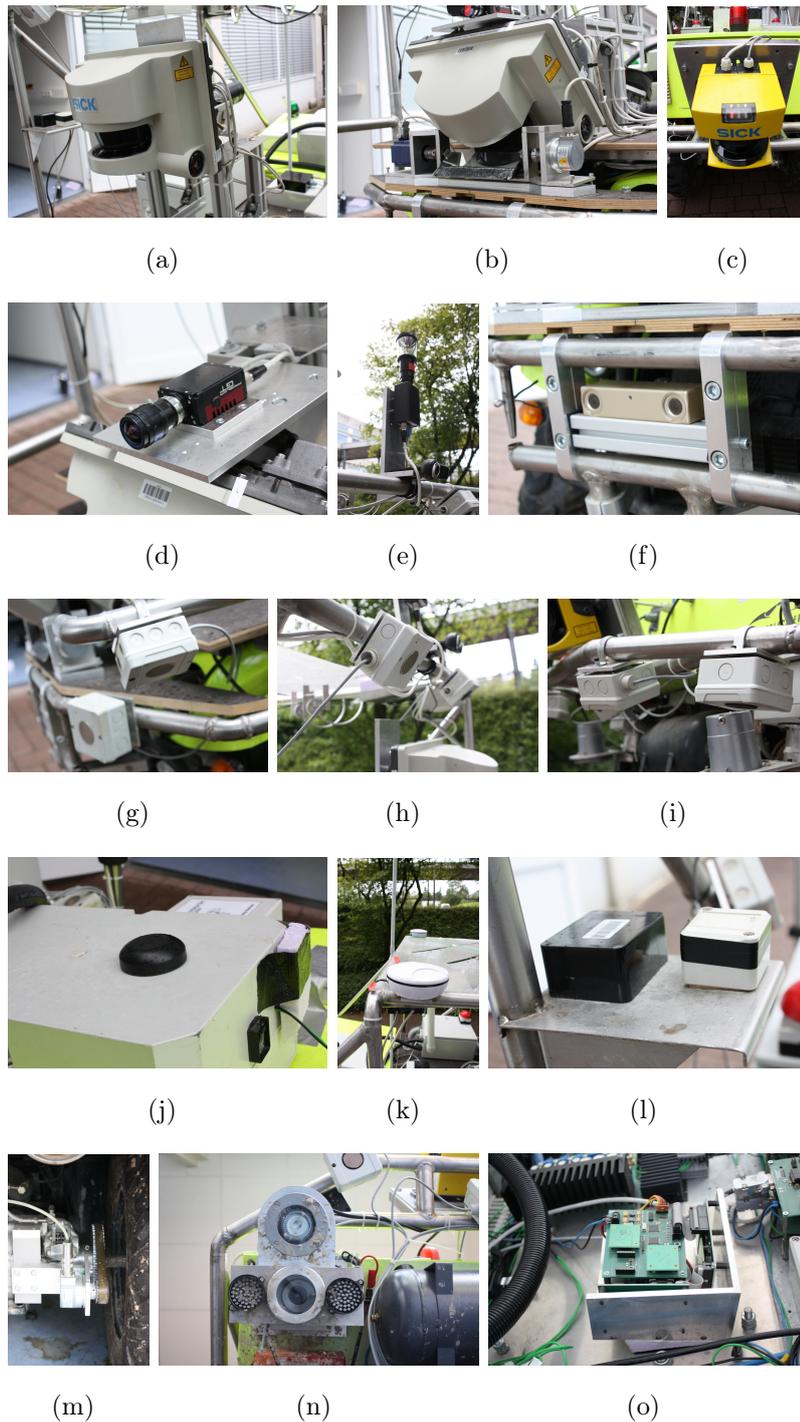


Abbildung 2.6: **Sensor-Ausstattung des Bodenroboters AMOR.** Laserscanner (a)-(c). Kameras (d)-(f). Ultraschallsensoren (g)-(i). GPS und Kompass (j)-(l). Radencoder, optische Bewegungsmessung und Inertialsystem (m)-(o).

Das heißt es werden nur dort Messpunkte erzeugt, wo die Messebene des Sensors die Umgebung schneidet. Um jedoch die geometrische Nachbarschaft von Messpunkten untersuchen zu können, was im komplexen Außenbereichsszenario zur Umgebungsbeschreibung und Hindernisdetektion und -klassifikation nahezu unumgänglich ist, gibt es unterschiedliche Herangehensweisen.

Der ganze Roboter kann mit dem darauf fest montierten Laserscanner bewegt werden. Dies führt hauptsächlich zu einer Translation (teilweise auch Rotation, je nach Beschaffenheit des befahrenen Bodens) der Messebene des Sensors und somit mit fortlaufender Bewegung des Roboters zu dem gewünschten vertikalen Abtasten der Umgebung. Dieser Ansatz wurde beim rückwärtig fest angebrachten Laserscanner gewählt. Nachteilig bei dieser Herangehensweise ist, dass ohne Bewegung des Roboters keine Abtastung in vertikaler Richtung und zusätzlich keine Mehrfach-Abtastung der Umgebung stattfindet, was beispielsweise bei Messfehlern oder sich bewegenden Hindernissen nicht sehr hilfreich ist.

Aus diesem Grund wurden die nach vorne messenden Laserscanner so konzipiert, dass sie nicht fest montiert sind, sondern jeweils eine Rotation ihrer Messebene auch ohne Bewegung des Roboters durchgeführt werden kann. Dabei wurden zwei unterschiedliche Prinzipien umgesetzt. Im einen Fall wird der gesamte Scanner inklusive Gehäuse rotiert. Dies ist ein weit verbreiteter Ansatz, der auch von einigen anderen Forschungsgruppen gewählt wurde ([179][210]). Das verhältnismäßig große Gewicht des Sensor-Gehäuses (ca. 9 kg) beschränkt jedoch die Rotationsgeschwindigkeit der Konstruktion. Für langsame Schwenkvorgänge ist dies ausreichend, sollen jedoch höhere Aktualisierungsraten realisiert werden (beispielsweise  $\geq 10$  Hz), kommt dieser Ansatz an seine Grenzen. Aus diesem Grund wurde für den Bodenroboter AMOR zusätzlich eine alternative Methode zur Rotation der Messebene eines Zeilen-Laserscanners entwickelt, welche höhere Rotationsgeschwindigkeiten der Messebene ermöglicht. Nicht der Sensor samt Gehäuse wird dabei rotiert, sondern ein im Vergleich deutlich leichter Spiegel, der vor dem Scanner im Strahlengang der messenden Laserstrahlen positioniert wird. So kann zwar nicht die Messgeschwindigkeit des Scanners selber erhöht werden, jedoch kann die Frequenz durchführbarer Schwenkvorgänge zur Erfassung einer kompletten Szene deutlich gesteigert werden. Eine solche Konstruktion bringt jedoch den Nachteil mit, dass im Ge-

gensatz zur spiegellosen Variante der Öffnungswinkel des Sensors von  $180^\circ$  auf etwa  $110^\circ$  eingeschränkt wird. Eine detaillierte Beschreibung des neuartigen Sensorsystems findet sich in [168].

Aus Gründen der Vollständigkeit sollen hier auch noch die zum Zwecke der Überwachung des extremen Nahbereichs angebrachten Ultraschall-Sensoren genannt werden. Da aufgrund der Montagepositionen der Laserscanner einige Hindernisse im Nahbereich des Roboters übersehen werden könnten, wurden Ultraschall-Sensoren an besonders gefährdeten Punkten montiert. Sie werden rein reaktiv zur Vermeidung von Gefahrensituationen eingesetzt.

Neben Sensoren zur Erfassung der unmittelbaren Umgebung sind diverse Sensoren zur Bestimmung der Raumlage bzw. zur Lokalisierung des Roboters im Weltkoordinatensystem hinzugefügt worden. Drehratensensoren (Gyroskope), Beschleunigungssensoren (Accelerometer) und Magnetfeldsensoren (Magnetometer) werden kombiniert verwendet, um die Orientierung des Roboters zu bestimmen. Diese auf dem Roboter verbaute Kombination aus Sensoren wird auch als *inertiales Navigationssystem* bezeichnet. Das inertielle Navigationssystem ist zwar in der Lage, die Orientierung des Roboters im dreidimensionalen Raum zu bestimmen, kann allerdings nicht isoliert zur Bestimmung der dreidimensionalen Position des Roboters bez. des globalen Weltkoordinatensystems genutzt werden. Der Roboter wurde zu diesem Zweck zusätzlich mit einem Rad-Encoder und einem optischen Bewegungserfassungssensor zur Messung der Überboden-Bewegung des Fahrzeugs ausgestattet. Die Sensorik ist bewusst auch an dieser Stelle redundant ausgelegt worden. In diesem Fall, um störende Effekte wie beispielsweise Schlupf der Räder detektieren und behandeln zu können.

Der optische Bewegungserfassungssensor ist eine Eigenentwicklung und bez. der Funktionsweise vereinfacht gesagt mit einer optischen Computer-Maus zu vergleichen. Aufbau und genaue Funktionsweise werden in [173] beschrieben. Zusätzlich zur Sensorik, welche die Überboden-Bewegung des Roboters erfasst, wurden sowohl ein einfacher GPS<sup>1</sup>-Empfänger als auch ein genauerer DGPS<sup>2</sup>-Empfänger verbaut. An dieser Stelle soll der Einsatz der GPS-Technologie in der Bodenrobotik im Außenbereich kurz näher beleuchtet werden, da der Ver-

---

<sup>1</sup>Global Positioning System

<sup>2</sup>Differential Global Positioning System

meidung der Abhängigkeit von dieser Technologie für die Lokalisierung des Bodenroboters AMOR in dieser Arbeit eine zentrale Rolle zukommt. GPS-Empfänger sind in der Lage, die globale Position des Roboters im Weltkoordinatensystem zu bestimmen. Die GPS-Technologie ist mittlerweile sehr bekannt, weit verbreitet, gut erprobt und kommt in vielen unterschiedlichen Anwendungen zum Einsatz. Die Lokalisierung mittels GPS stützt sich auf den Empfang und die Auswertung von Signalen, die von Satelliten auf einer Erdumlaufbahn ausgesendet werden. Prinzipiell ist dies eine äußerst komfortable Möglichkeit, die Position eines Fahrzeugs auf der Erde zu bestimmen. Wie jedoch in Tabelle 2.1 bereits angedeutet wurde, kann die Verfügbarkeit und/oder die Genauigkeit der Lokalisierung durch GPS stark eingeschränkt sein. Dies kann an der fehlenden Sichtbarkeit von Satelliten oder der Störung bzw. Reflexion der von den Satelliten ausgesendeten Signale liegen. Aus diesem Grund wird in dieser Arbeit die GPS-Technologie auf dem Bodenroboter AMOR nur verwendet, um eine sehr grobe Schätzung der Position des Roboters im erd-festen Koordinatensystem zu ermitteln. In der konkreten Planung und Umsetzung von autonomen Fahrmanövern wird die GPS-Technologie jedoch nicht verwendet, um nicht von dieser besonders im Gelände unzuverlässigen Lokalisierungsmethode abhängig zu sein. Mit dieser Arbeit wird somit u.a. auch der Nachweis erbracht, dass zielgerichtetes, autonomes Fahren über längere Strecken im Außenbereich auch ohne den ständigen Einsatz der GPS-Technologie auf dem Bodenfahrzeug möglich ist. Kapitel 4 gibt in diesem Zusammenhang detaillierten Einblick in die Umsetzung der Lokalisierung des Bodenroboters AMOR.

### 2.1.3 Sensor/Aktor Abstraktion

Die einzelnen Komponenten der Aktorik als auch die verschiedenen verbauten Sensoren verwenden nativ verschiedenste Kommunikationsprotokolle zur Übermittlung der Eingabe- bzw. Ausgabedaten (u.a. Ethernet, USB, CAN, I<sup>2</sup>C, RS232 etc.). Daher besteht eine weitere zentrale Designentscheidung in der vereinheitlichten Zusammenführung aller Aktoren und Sensoren in einer Sensor-Aktor-Abstraktionsebene (SAAE). Als einheitliches Kommunikationsprotokoll kommt dabei das Ethernet-Protokoll zum Einsatz (bzw. Protokolle,

die auf dem Ethernet-Protokoll aufbauen), d.h., alle Daten werden so umgewandelt, dass sie, sobald sie die SAAE erreichen, per Ethernet kommuniziert werden. Ob dabei das TCP- oder das UDP-Protokoll eingesetzt wird, ist nicht festgelegt. Das Format, das den Inhalt der Nachricht definiert, ist ebenfalls an dieser Stelle noch nicht fest definiert. Wichtig ist lediglich, dass die Daten per Ethernet kommuniziert werden. Diese Entscheidung wird durch das Streben nach einer starken Modularisierung des Gesamtsystems motiviert. So können die einzelnen Komponenten des Gesamtsystems über mehrere Computer und Microcontroller verteilt werden, ohne darauf zu achten, welche Prozessor-Architektur oder welches Betriebssystem jeweils eingesetzt wird. Es muss lediglich eine Ethernet-Schnittstelle vorhanden sein.

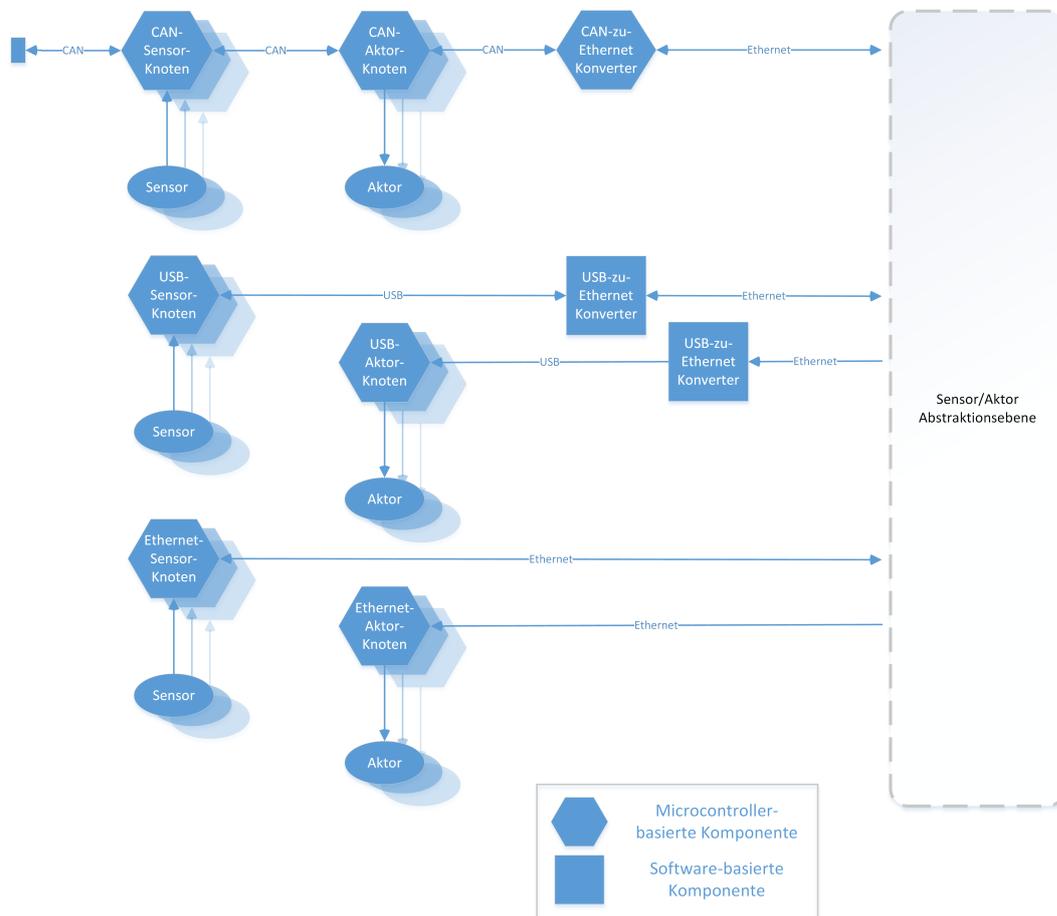


Abbildung 2.7: **Abstraktion der Aktorik- und Sensorikkomponenten.**

Abbildung 2.7 zeigt, wie der Datenfluss von Sensor bzw. Aktor zur SAAE

auf dem Roboter AMOR realisiert ist. Dabei kommen drei Kommunikationsprotokolltypen zum Einsatz. Sensoren bzw. Aktoren, die bereits direkt ein auf Ethernet basierendes Protokoll verwenden, bedürfen keiner weiteren Komponente zur Umwandlung des Übertragungsprotokolls. Per USB angeschlossene Komponenten sind direkt an einen der in AMOR verbauten Computer angeschlossen. Dort findet die Umwandlung des Übertragungsprotokolls nach Ethernet mithilfe einer Treiber-Softwarekomponente statt. Alle weiteren Sensoren bzw. Aktoren, die mittels eines anderen seriellen Protokolls (I<sup>2</sup>C, RS232, etc.) kommunizieren, besitzen jeweils eine Sensor- bzw. Aktor-nahe Verarbeitungskomponente (Microcontroller), die zunächst eine Umwandlung der Daten in ein proprietäres CAN-Bus-Protokoll durchführt. Die Umwandlung der auf dem CAN-Bus versendeten Daten übernimmt dann ein ebenfalls mit einem Microcontroller realisierter CAN-zu-UDP-Ethernet-Umwandler. Neben der Vereinheitlichung des Kommunikationsmechanismus<sup>3</sup> für Sensor- bzw. Aktor-Daten liegt auch eine Vereinheitlichung des Nachrichtenformats nahe. Daher ist auch der inhaltliche Aufbau der Nachrichten einheitlich festgelegt worden. Dabei wird sich in dieser Arbeit des Robotik-Software-Frameworks ROS<sup>3</sup> bedient [160].

## 2.2 Das Team: Unterstützung aus der Luft

Neben Bodenrobotern gewinnen in der Außenbereichsrobotik zunehmend auch andere Robotertypen an Bedeutung. Gerade der Einsatz von Flugrobotern drängt sich aufgrund der komplementären Eigenschaften im Vergleich mit am Boden operierenden Robotern geradezu auf. Tabelle 2.2 gibt einen Überblick der Eigenschaften in diesem Zusammenhang.

Aufgrund der komplementären Eigenschaften der beiden unterschiedlichen Robotertypen werden im Rahmen dieser Arbeit Bodenroboter und Flugroboter in Form eines kooperierenden Teams eingesetzt. Dabei ist sehr wichtig hervorzuheben, dass der **klare Entwicklungsfokus** auf dem Bodenroboter AMOR lag. Der Flugroboter PSYCHE wurde von Anfang an als **externe mobile Sensorplattform** für den Bodenroboter AMOR konzipiert. Die überlegene

---

<sup>3</sup>Robot Operating System. <http://www.ros.org>

<b>Robotertyp/Eigenschaft</b>	Bodenroboter	Flugroboter
Mobilität	-	+
Reichweite/Ausdauer	+	-
Rechenleistung	+	-
Nutzlast	+	-
GPS-Verfügbarkeit	-	+
Hindernisse	-	+

Tabelle 2.2: Vergleich von Boden- und Flugroboter im Außenbereichs-Szenario.

Mobilität des Flugroboters und die daraus resultierende vorteilhafte Übersicht über die aktuelle Szene, in welcher der Bodenroboter operiert, bieten enormen Mehrwert. Ergänzt wird dies durch die minimale Störung absoluter Lokalisierungssensorik (bspw. GPS-Empfänger), die ab einer gewissen Operationshöhe des Flugroboters angenommen werden kann.

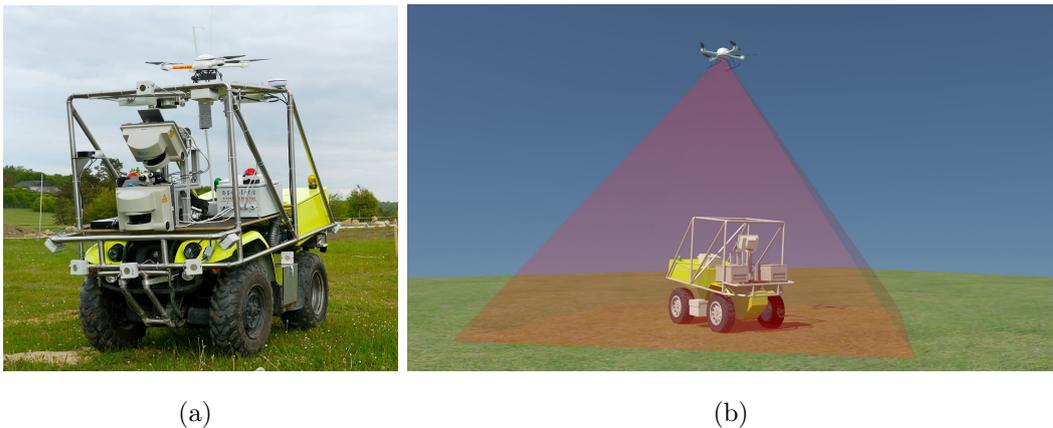


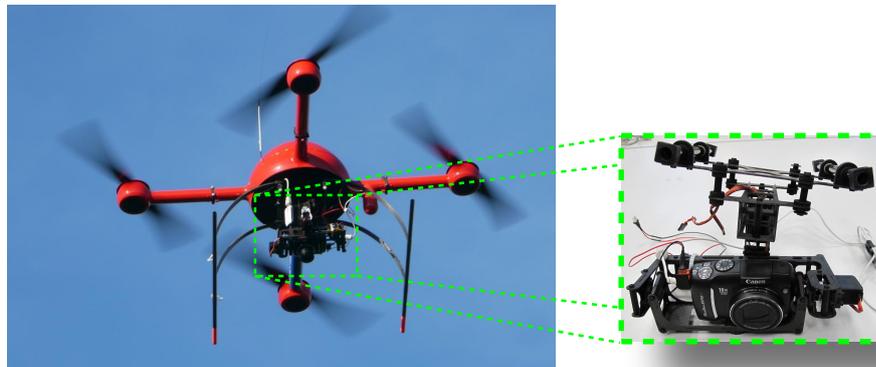
Abbildung 2.8: Das kooperierende Team bestehend aus dem Bodenroboter AMOR und dem Flugroboter PSYCHE.

Aufgrund der äußerst beschränkten Energiekapazität, die auf dem Flugroboter zur Verfügung steht, hat sich die Verarbeitung von Sensordaten mittels komplexer Algorithmen direkt auf dem Flugroboter als nicht effektiv erwiesen. Daher werden alle in der Luft generierten Daten an den Bodenroboter gesendet, um dort mithilfe der vorhandenen überlegenen Rechenleistung verarbeitet werden zu können. Die Nutzlast ist je nach Typ des Flugroboters im Vergleich

zu der Nutzlast eines Bodenroboters äußerst beschränkt (siehe Kapitel 2.3). Dies bedeutet eine starke Einschränkung bei der Auswahl der mitgeführten Sensorik, sodass bez. dieses Punktes sehr selektiv vorgegangen werden muss. Das Nichtvorhandensein von physischen Hindernissen, solange der Flugroboter in einer gewissen Mindesthöhe operiert, stellt dabei eine große Erleichterung dar. Denn so muss keine aufwendige Sensorik auf dem Flugroboter mitgeführt werden, um Hindernissen ausweichen zu können. Dies stellt selbstverständlich einen weiteren großen Unterschied zu einem Bodenroboter dar, welcher auf äußerst komplexe Hindernisse im Außenbereich stoßen kann.

## 2.3 PSYCHE: Der Flugroboter

Ähnlich wie für den Bodenroboter AMOR fiel auch für den **Flugroboter PSYCHE** [11] die Wahl auf ein erprobtes, kommerziell erhältliches Gerät als Basis für den Flugroboter. Zwar kann der Quadrokoopter *microdrones md4-1000* (siehe Abb. 2.9) zum Zeitpunkt der Entstehung dieser Arbeit noch nicht als Massenprodukt bezeichnet werden, die Entwicklungsarbeit des Herstellers hat dennoch zu einem ausgereiften, leicht benutzbaren Fluggerät geführt, welches für den Einsatzzweck in dieser Arbeit sehr gut geeignet ist.



(a)

Abbildung 2.9: **Flugroboter PSYCHE**, ausgerüstet mit hochauflösender Luftbildkamera in nivellierender Aufhängung als Nutzlast.

Der Flugroboter kann mit verschiedenen Nutzlasten ausgestattet werden, wobei der begrenzende Faktor bei der Wahl einer Nutzlast hauptsächlich deren

Gewicht darstellt. In dieser Arbeit wird eine hochauflösende Luftbildkamera als Nutzlast verwendet, welche in einer aktiv nivellierenden Aufhängung angebracht ist (siehe auch [103] für ein weiteres Nutzlastbeispiel). Dies bedeutet, dass die optische Achse der Kamera und der Gravitationsvektor stets näherungsweise parallel sind, was eine deutliche Vereinfachung in der Nutzung der Sensordaten der Luftbildkamera durch den Bodenroboter bedeutet (siehe Abb. 5.26 in Kapitel 5.4 für eine detaillierte Erörterung des beschriebenen geometrischen Zusammenhangs).

Wie schon zuvor betont, fungiert der Flugroboter ausschließlich als externe Sensorplattform für den Bodenroboter. Dabei folgt der Flugroboter dem Bodenroboter und transferiert kontinuierlich Luftbilder an den Bodenroboter. Der Flugroboter muss, um diese Aufgabe umsetzen zu können, über gewisse autonome Fähigkeiten verfügen.

Die Umsetzung der Autonomie des Flugroboters ist jedoch ungleich einfacher als die des Bodenroboters. Dies liegt in folgenden zentralen Eigenschaften der Operation eines autonomen Quadropters begründet:

Erstens ist der Flugroboter ein holonomes System. Dies vereinfacht die Pfadplanung, da u.a. keine kinematischen Einschränkungen für die Bewegung des Roboters in Betracht gezogen werden müssen.

Zweitens kann ab einer ausreichenden Mindestflughöhe davon ausgegangen werden, dass sich der Flugroboter mithilfe von satellitenbasierten Lokalisierungsmethoden robust lokalisieren kann und keinerlei Hindernisse in der Luft zu erwarten sind. Abbildung 2.10 bietet einen Gesamtüberblick über die für die Umsetzung der autonomen Fähigkeiten relevanten Systemkomponenten des Flugroboters PSYCHE sowie den Datenaustausch mit dem Bodenroboter AMOR.

Die Basis der autonomen Operation liegt in der automatischen Lageregelung des Luftfahrzeugs, die unter Verwendung von Inertialsensorik (Sensoren für Beschleunigungen, Magnetfelder und Winkelgeschwindigkeiten) eine stabile Fluglage in der Luft auch unter Störeinflüssen (Wind etc.) erzeugt. Die Positionsregelung teilt sich, was die genutzte Sensorik angeht, in zwei Komponenten: Zum einen wird unter Zuhilfenahme eines Barometers die Höhe des Roboters bezogen auf den Startpunkt reguliert, zum anderen wird die Position des Roboters im erdfesten Koordinatensystem mithilfe eines GPS-Empfängers geregelt.

Der Flugroboter hält mithilfe der Positionsregelung seine Position im erdfesten Koordinatensystem, solange nicht eine andere Referenzposition an den Flugroboter gesendet wird. Damit hat der Flugroboter genügend Autonomie, um als externe Sensorplattform dem Bodenroboter folgen zu können.

Das Starten und Landen des Flugroboters schließlich wird in dieser Arbeit nicht autonom, sondern stets manuell durchgeführt, da diese Aufgaben ein eigenständiges Forschungsthema darstellen und nicht in den Hauptfokus dieser Arbeit fallen.

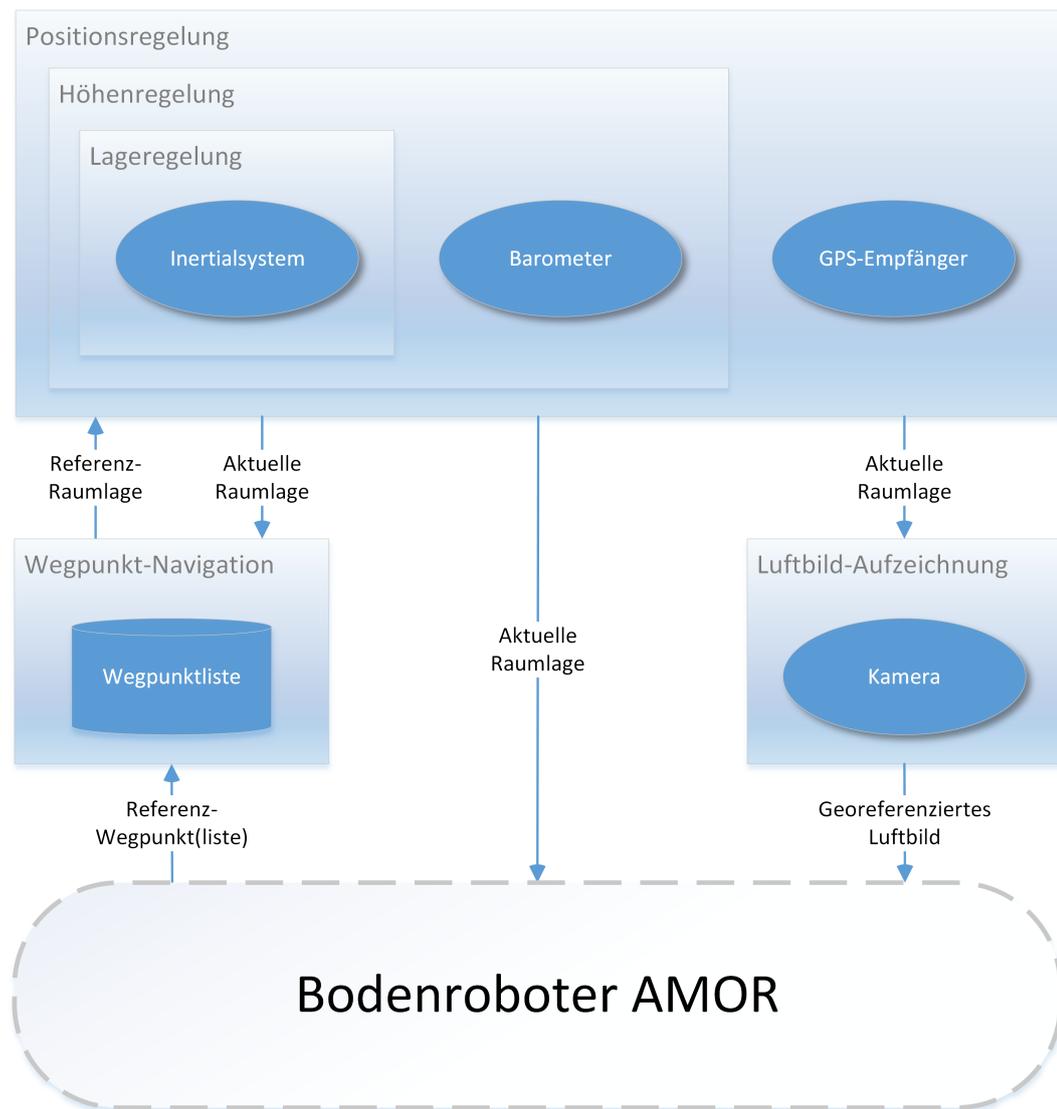


Abbildung 2.10: Systemkomponenten des Flugroboters PSYCHE.



# Kapitel 3

## Systemarchitektur des Roboterteams

Dieses Kapitel gibt einen Gesamtüberblick über alle Systemkomponenten, die für die Umsetzung des autonomen Explorationsszenarios mit dem Roboterteam AMOR und PSYCHE im Rahmen dieser Arbeit implementiert wurden. Darunter befinden sich die Komponenten Lokalisierung, Umgebungsmodellierung, Pfadplanung und Kooperation, im Kontext derer der überwiegende Anteil des wissenschaftlichen Beitrags dieser Arbeit erarbeitet wurde (siehe Kapitel 1.2). Aus diesem Grund werden diese detailliert in den Kapiteln 4, 5 und 6 beschrieben. Um jedoch eine erklärende Einbettung dieser zentralen Komponenten in die gesamte Systemarchitektur zu leisten, werden in diesem Kapitel zusätzlich alle anderen beteiligten Systemkomponenten mit reduzierter Detailfülle beschrieben. Dabei liegt der **Fokus** stets auf der **Funktion einer Komponente im Gesamtzusammenhang des Systems** und nicht auf der detaillierten inhaltlichen Beschreibung der jeweiligen Komponenten. Die Abhängigkeiten der wichtigsten Systemkomponenten des komplexen Multirobotersystems sind in Abb. 3.1 grafisch zusammengefasst. Dabei wurde die Darstellung auf die zentralen Schlüsselkomponenten reduziert, die sich in den folgenden Unterkapiteln wiederfinden.

Die in Abbildung 3.1 dargestellten Systemkomponenten sind in zwei Hauptkategorien gegliedert, **globale** und **lokale** Komponenten. Die Kategorisierung orientiert sich an den Referenzkoordinatensystemen (siehe Kapitel 4), bezüg-

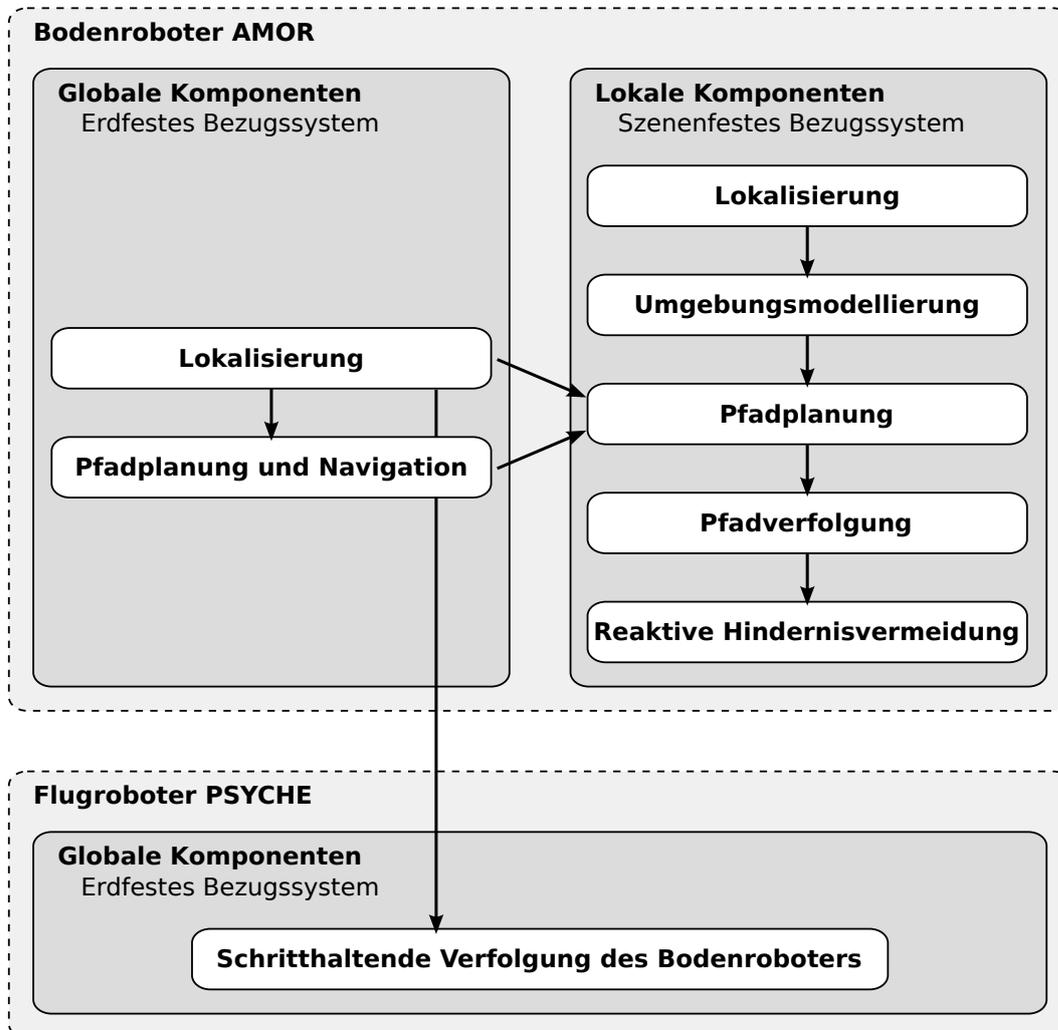


Abbildung 3.1: Gesamtübersicht der Abhängigkeiten der wichtigsten funktionalen Systemkomponenten des Multirobotersystems, bestehend aus Bodenroboter AMOR und Flugroboter PSYCHE.

lich derer die Daten in den jeweiligen Komponenten definiert sind. In globalen Komponenten sind die Daten bezüglich eines erdfesten ("globalen") Koordinatensystems definiert. In lokalen Komponenten hingegen sind die Daten bezüglich eines szenenfesten ("lokalen") Koordinatensystems definiert. Die Nutzung verschiedener Referenzkoordinatensysteme motiviert sich aus unterschiedlichen Ansprüchen der Komponenten an globale Eindeutigkeit einerseits und lokale Konsistenz und Genauigkeit andererseits. Dies wird in den folgenden Kapiteln dieser Arbeit weiter detailliert.

## 3.1 Globale Komponenten

Die Aufgabe der globalen Komponenten ist die Lokalisierung der Roboter im erdfesten Koordinatensystem, um darauf aufbauend die globale Navigation des Roboterteams zu realisieren. Dies beinhaltet die Planung einer Sequenz von Wegpunkten zu einer oder mehreren manuell festgelegten Zielkoordinaten. Dies geschieht auf Basis eines Netzes aus Wegsegmenten, sofern ein solches Netz vorhanden ist. Ist kein Wegenetz vorhanden, reduziert sich die Liste der Wegpunkte auf die manuell eingegebenen Zielkoordinaten. Unabhängig von der Existenz eines Wegenetzes wird also grundsätzlich der aktuell anzusteuernde Wegpunkt ermittelt, um so die aktuelle Zielpose im erdfesten Koordinatensystem zu erhalten. Dieses globale Ziel stellt schließlich die einzige Schnittstelle zu den lokalen Komponenten des Systems dar. Somit ist, abgesehen von dieser Schnittstelle, eine klare Trennung zwischen globalen und lokalen Komponenten realisiert. Das bedeutet eine konsequente Umsetzung des Konzepts der lokalen Szene für die autonome Exploration in dieser Arbeit (siehe Kapitel 5.1 für die Definition des Begriffs der Szene).

### 3.1.1 Lokalisierung

Für das Problem der Lokalisierung eines Fahrzeugs existiert eine Vielzahl an Lösungsansätzen. Bei der Auswahl der richtigen Methode sind diverse Kriterien in Betracht zu ziehen. Da der Fokus dieser Arbeit ausschließlich auf der Außenbereichsrobotik liegt, scheiden etablierte Methoden aus, die ausschließ-

lich für die Nutzung im Innenbereich konzipiert sind, wie etwa die Nutzung von synthetischen oder natürlichen (bspw. linienförmigen oder planaren) visuellen Landmarken [211][3], aktiven (Ultra-)Schall- oder Funksendern [77][35] oder präzisen, apriori bekannten (3D-)Karten [83].

In der Außenbereichsrobotik ist der gebräuchlichste Ansatz für die Lokalisierung bezüglich eines erdfesten Koordinatensystems die Nutzung von satellitenbasierten Methoden (bspw. GPS [80]). Der Grund hierfür liegt in der Ausgereiftheit der zugehörigen Empfänger und der Einfachheit ihrer Benutzung. Um Ausfälle zu kompensieren, werden die Messungen meist mit einem Kalman-Filter oder alternativen Filtermethoden verarbeitet [91]. Adaptive Filterverfahren zeigen darauf aufbauend ein verbessertes Verhalten bei spontanen Richtungs- oder Geschwindigkeitsänderungen [85]. Auch die Integration von Messungen, die von relativen Lokalisierungsverfahren generiert wurden, ist üblich, um Störungen und Ausfälle zu behandeln [143][205]. Neben satellitengestützten Verfahren sind in der Außenbereichsrobotik visuelle Methoden zur Lokalisierung weit verbreitet. Die meisten Ansätze dieser Art befassen sich mit der relativen Lokalisierung einer bewegten Kamera [146] (oder eines bewegten Stereokamerasystems [64]) auf Einzelbild-zu-Einzelbild-Basis und sind damit nicht direkt für die absolute Lokalisierung geeignet. Einige Ansätze erreichen eine absolute Lokalisierung durch die visuelle Wiedererkennung von Orten, die zuvor in Form eines visuellen Alphabets oder einer topologischen Karte mit visueller Annotation je Knoten der Karte abgelegt wurden.

In dieser Arbeit wurden für die globale Lokalisierung neue Methoden entwickelt und genutzt. Erstens wird die Abhängigkeit von satellitenbasierten Lokalisierungsmethoden durch die Entwicklung eines Verfahrens, das ausschließlich Daten aus einem digitalen Wegenetz und der Koppelnavigation des Bodenroboters verwendet, nahezu vollständig eliminiert [109]. Zweitens wird die Verfügbarkeit eines kooperierenden Flugroboters genutzt, um den Bodenroboter durch Registrierung von Live-Luftbildern und georegistrierten Orthofotos global zu lokalisieren [105]. Die Details beider Verfahren sowie ausführliche Grundlagen zum Themenbereich Lokalisierung finden sich in Kapitel 4. Die Ausgabe des globalen Lokalisierungsmoduls ist, unabhängig von der gewählten Methode der Erzeugung, stets eine 3D-Position im erdfesten Koordinatensystem.

### 3.1.2 Pfadplanung und Navigation

Wie bereits zu Beginn des Kapitels angesprochen wurde, basiert die globale Pfadplanung des in dieser Arbeit entwickelten Systems auf der Annahme der Existenz eines Wegenetzes, das einer topologischen Karte mit metrischen Annotationen je Knoten entspricht. Die Kanten der topologischen Karte entsprechen befahrbaren Wegen. Durch die metrischen Annotationen der Knoten wird der Zusammenhang zum erdfesten Koordinatensystem hergestellt und gleichzeitig sind implizit die Distanzen zwischen den Knoten im erdfesten Koordinatensystem gegeben, die als Annotationen der Kanten bzw. Kantengewichtungen genutzt werden. Die Verfügbarkeit von Datenbanken mit annotierten Wegenetzen, wie etwa die in dieser Arbeit genutzte Datenbank *OpenStreet-Map*<sup>1</sup>, ermöglichen den einfachen Zugriff auf Wegenetzdaten weltweit (siehe Abb. 3.2). Bei eventuell fehlenden Wegsegmenten im Operationsgebiet des Bodenroboters können diese ohne großen Aufwand auch manuell hinzugefügt werden. Somit kann im Folgenden von der Verfügbarkeit eines Wegenetzes für die globale Pfadplanung ausgegangen werden.

Das Ziel der globalen Pfadplanung besteht nun in der Ermittlung einer Sequenz von durch Kanten verknüpften Knoten aus dem Wegenetz, die den kürzesten Pfad von der aktuellen Position des Roboters zu einer oder mehreren Zielkoordinaten realisiert. Da die topologische Karte einem Graphen entspricht, kann das globale Pfadplanungsproblem durch lange bekannte und erprobte Algorithmen aus der Graphentheorie gelöst werden. Der Dijkstra-Algorithmus [49] berechnet einen Pfad minimaler Länge von einem Startknoten zu einem Zielknoten mit der Komplexität  $\mathcal{O}(n \log n + m)$ , wobei  $n$  der Anzahl der Knoten des Graphs und  $m$  der Anzahl der Kanten des Graphs entspricht. Der Nachteil dieses Algorithmus ist, dass keine zielgerichtete Suche durchgeführt wird. An diesem Punkt setzt der A\*-Algorithmus von Hart et al. [76][145] an, der die Pfadsuche nicht ausschließlich auf Basis der Kantengewichtungen des Graphen, sondern zusätzlich auf Basis einer frei definierbaren Heuristik umsetzt. Damit wird während der Suche nicht der Pfad mit der geringsten Kantengewichtung erweitert, sondern vielmehr der Pfad, der unter Zuhilfenahme der Heuristik aktuell am günstigsten erscheint. In der Praxis einer metrisch annotierten to-

---

<sup>1</sup><http://www.openstreetmap.org>

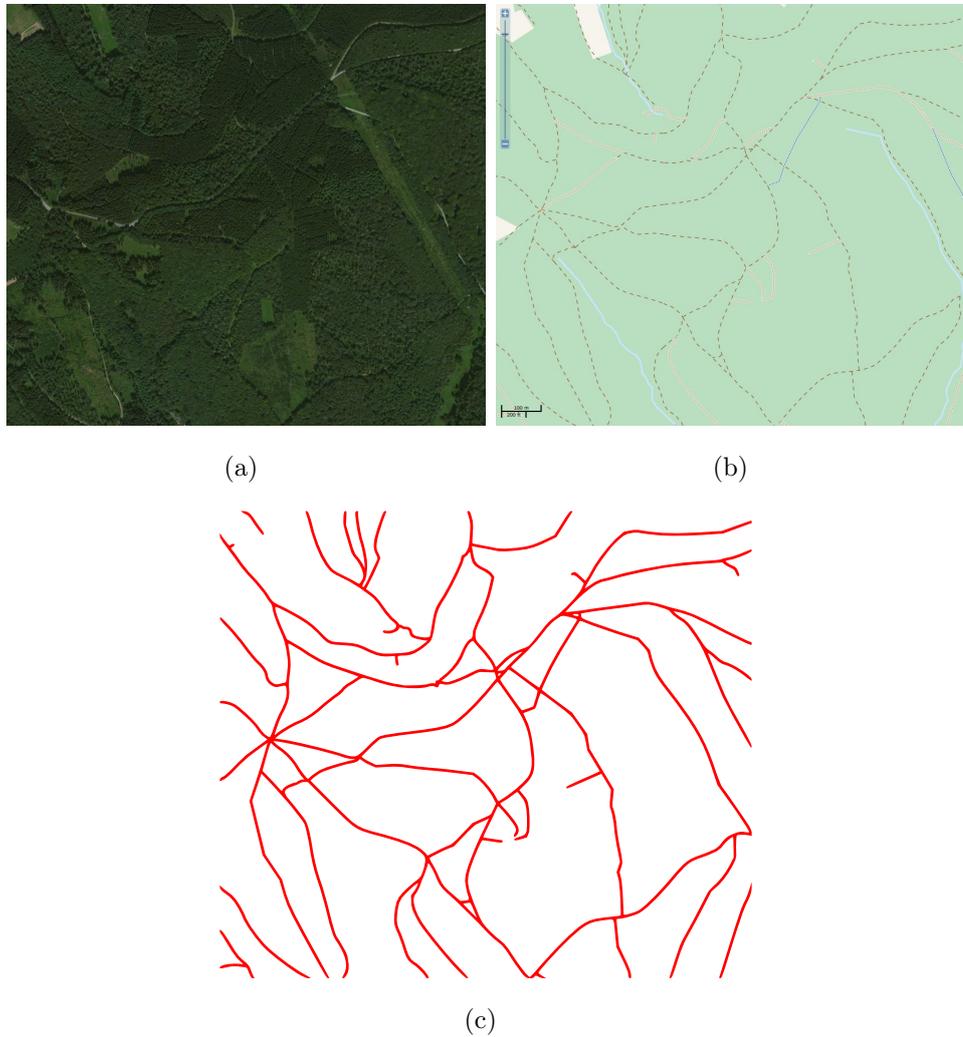


Abbildung 3.2: **Datengrundlage der kartenbasierten globalen Planung.** Orthofoto entnommen aus der Datenbank *Bing Maps* (<http://www.bing.com/maps>) (a). Zugehöriger gerendertes Ausschnitt von Kartendaten, entnommen aus der Datenbank *OpenStreetMap* (b). Visualisierung der extrahierten Wegsegmente, die für die globale Planung eingesetzt werden (c). Copyright des Orthofotos: Microsoft Inc.

pologischen Karte wird die euklidische Distanz vom gerade betrachteten Knoten zum Zielknoten (also die Luftlinie) als Heuristik gewählt. So wird eine zielgerichtete und in den meisten Fällen deutlich effizientere Suche des minimalen Pfads realisiert. Aufgrund der höheren Leistungsfähigkeit wird dieses Verfahren in dieser Arbeit zur globalen Pfadplanung im vorgegebenen Wegesetz verwendet. Der A\*-Algorithmus findet zwar nicht immer den optimalen Weg; in den verwendeten realen Karten sind die Unterschiede zum wesentlich aufwendigeren Dijkstra-Algorithmus jedoch fast immer sehr gering.

Die globale Planung wird zu Beginn jeder Explorationsfahrt durchgeführt und stellt der nachfolgenden globalen Navigationskomponente eine Sequenz von Wegpunkten im erdfesten Koordinatensystem zur Verfügung.

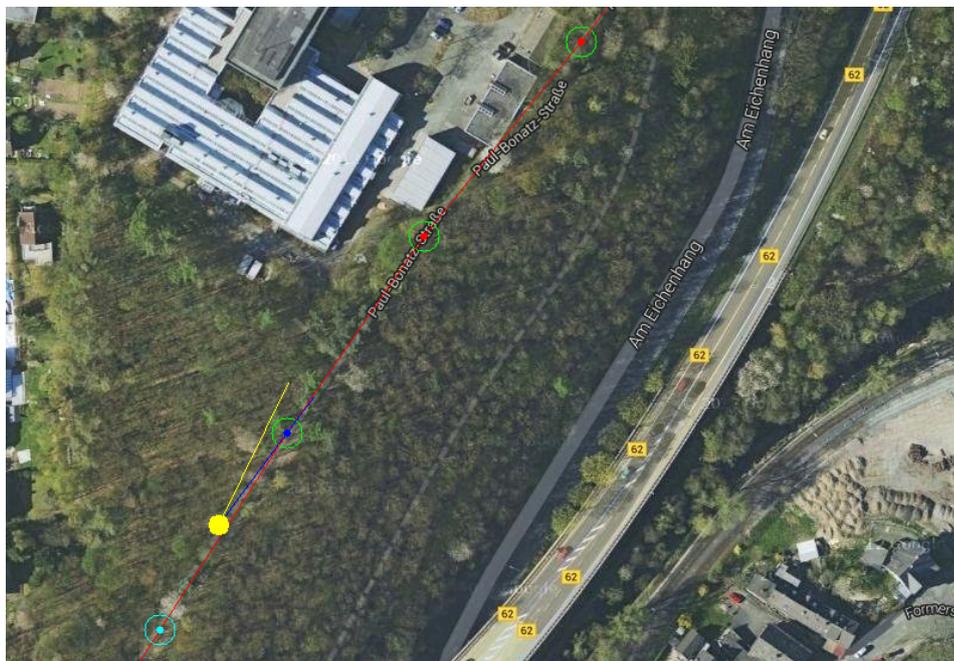


Abbildung 3.3: **Visualisierung der globalen Planungskomponente während einer Testfahrt.** Aktueller Wegpunkt (blau). Zukünftige Wegpunkte (rot). Bereits erreichte Wegpunkte (cyan). Wegpunktradien (grün). Aktuelle Fahrzeugpose (gelb). Copyright des Luftbilds: Google Inc.

Die globale Navigationskomponente ermittelt den aktuell anzusteuern den Wegpunkt aus der Liste der zuvor durch die globale Planung erzeugten Wegpunkte (siehe Abb. 3.3).

---

**Algorithmus 1** Globale Wegpunktnavigation

---

```
for  $j \leftarrow 0$  to  $n$  do  
  |  $waypoints[j].reached \leftarrow False$   
end  
 $i \leftarrow 0$   
 $d_{min} \leftarrow \infty$   
while  $waypoints[n].reached \neq True$  do  
  |  $d_{current} \leftarrow distance(waypoints[i], position_{robot})$   
  |  $d_{min} \leftarrow \min(d_{min}, d_{current})$   
  | if  $d_{current} > d_{min} + d_{tolerance}$  then  
    |  $waypoints[i].radius \leftarrow waypoints[i].radius + r_{increment}$   
  | else  
    |  $reset(waypoints[i].radius)$   
  | end  
  | if  $d_{current} < waypoints[i].radius$  then  
    |  $waypoints[i].reached \leftarrow True$   
    |  $i \leftarrow i + 1$   
    |  $d_{min} \leftarrow \infty$   
  | end  
  |  $publish(waypoints[i])$   
end
```

---

Algorithmus 1 zeigt den gesamten Ablauf der globalen Wegpunktnavigation in kompakter Pseudocode-Form. Beginnend beim ersten Wegpunkt der Liste wird der jeweils aktuelle Wegpunkt darauf überprüft, ob er bereits erreicht worden ist. Ist dies der Fall, d.h., ist der Abstand des Fahrzeugs zum aktuellen Wegpunkt geringer als der Wegpunktradius ("*radius*") des aktuellen Wegpunkts, wird dieser als erreicht ("*reached*") markiert und der nächste Wegpunkt der Sequenz als aktueller Wegpunkt ausgewählt. Dies wird solange wiederholt, bis die Wegpunktsequenz vollständig abgearbeitet wurde. Die globalen Wegpunkte sind grundsätzlich nicht als präzise anzufahrende Positionen zu sehen, sondern eher als grobe Navigationsrichtlinie. Es kann durch bestimmte lokale Hinderniskonfigurationen oder Ungenauigkeiten im Wegenetz durchaus dazu kommen, dass ein globaler Wegpunkt aufgrund eines zu geringen Wegpunktradius nicht erreicht werden kann. Damit dies nicht zum vollständigen Abbruch des Explorationsvorhabens des Roboterteams führt, wurde dem Navigationsalgorithmus ein Mechanismus zur Behandlung dieser Problematik hinzugefügt. Es wird stets die minimale Distanz zum aktuellen Wegpunkt vorgehalten. Wird die Distanz des Roboters zum aktuellen Wegpunkt zu einem gewissen Zeitpunkt größer als die minimale Distanz zuzüglich einer Fehlertoleranz, wird davon ausgegangen, dass der Wegpunkt nicht erreicht werden kann. Um dieser Tatsache zu begegnen, wird ab dem Zeitpunkt der Detektion der Nichterreichbarkeit des Wegpunkts die Wegpunktdistanz pro Durchlauf des Navigationsalgorithmus deutlich erhöht. So wird sichergestellt, dass jeder Wegpunkt prinzipiell erreichbar bleibt.

### 3.1.3 Schritthaltende Verfolgung des Bodenroboters durch den Flugroboter

Der Flugroboter hat als externe Sensorplattform für den Bodenroboter die Aufgabe, aktuelle Luftbilder aufzuzeichnen. Zur Erfüllung dieser Aufgabe folgt der Flugroboter dem Bodenroboter und nimmt währenddessen Luftbilder auf, die dem Bodenroboter zur Weiterverarbeitung zur Verfügung gestellt werden (siehe Abb. 3.4 für eine beispielhafte Außenbereichsszene mit dem Roboterteam AMOR und PSYCHE).



Abbildung 3.4: **Schritthaltende Verfolgung des Bodenroboters AMOR durch den Flugroboter PSYCHE in einer Außenbereichsszene.**

Der Flugroboter verfügt über die Fähigkeit, autonom Wegpunkte anzufliegen, die im erdfesten Koordinatensystem definiert sind. In diesem Zusammenhang wird in dieser Arbeit angenommen, dass sich ein Flugroboter, der in einer gewissen Mindestflughöhe operiert. Zusätzlich kann er sich stets (bis auf wenige sehr seltene Ausnahmen) mit Methoden der satellitenbasierter Lokalisierung (bspw. GPS) bezüglich eines erdfesten Koordinatensystems lokalisieren. Im Gegensatz zum Bodenroboter, der häufig Störungen oder Ausfällen von Satellitensignalen durch Verdeckungen ausgesetzt ist, werden für die Lokalisierung des Flugroboters im erdfesten Koordinatensystem daher satellitengestützte Lokalisierungsmethoden verwendet. Daher kann stets davon ausgegangen werden, dass die Kommunikation der aktuellen Position des Bodenroboters an den Flugroboter als anzufliegender Wegpunkt ausreichend ist, um eine schritthaltende Verfolgung des Bodenroboters zu implementieren.

Als Optimierung wird zusätzlich eine Verschiebung des Wegpunkts entlang der Richtung der positiven Längsachse des Bodenroboters durchgeführt, um ein Vorausfliegen des Flugroboters zu erreichen (siehe Abb. 3.5). Dies liegt

in der implementierten Methode der Luftbildaufzeichnung begründet, da der Flugroboter für die Aufzeichnung seine aktuelle Position in der Luft hält, sich also für den Aufzeichnungszeitraum nicht mit dem Bodenroboter mitbewegt. So können für die Aufzeichnung möglichst ideale Verhältnisse (Minimierung der Bewegungsunschärfe, ruhige Fluglage etc.) garantiert werden. Durch den verschobenen Wegpunkt ist der Flugroboter zum Aufzeichnungspunkt in etwa über und nicht deutlich hinter dem Bodenroboter positioniert. So wird eine gute Abdeckung des Umgebungsmodells des Bodenroboters durch das Luftbild erreicht. Für die Auslösung einer Luftbildaufzeichnung gelten die beiden Bedingungen, dass erstens der Flugroboter den aktuellen Wegpunkt mit Berücksichtigung einer gewissen Toleranz erreicht haben muss, und zweitens, dass die Kamera bereit für eine erneute Auslösung sein muss. Sind beide Bedingungen erfüllt, hält der Flugroboter kurz seine Position und es wird ein Luftbild aufgenommen, das direkt nach der Aufzeichnung an den Bodenroboter gesendet wird.

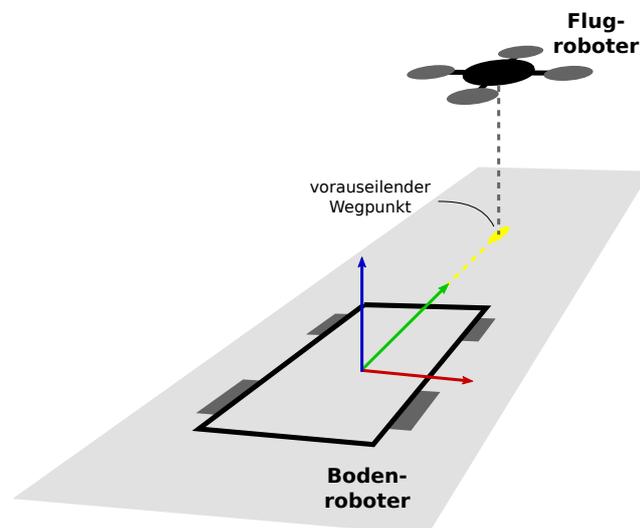


Abbildung 3.5: **Vorseilender Wegpunkt für die Navigation des Flugroboters.**

## 3.2 Lokale Komponenten

Die lokalen Komponenten befassen sich inhaltlich mit allen Aufgaben, die für die autonome Operation des Bodenroboters, basierend auf der aktuell von der auf dem Roboter verbauten Sensorik erfassbaren Umgebung, also der **lokalen Szene**, notwendig sind. Dies umfasst eine Verarbeitungskette, die sich aus den Komponenten Lokalisierung, Umgebungsmodellierung, Hindernisdetektion, Pfadplanung und Pfadverfolgung zusammensetzt. Die Lokalisierung ermittelt die Pose des Roboters im Szenenkoordinatensystem.

Die Umgebungsmodellierung erzeugt durch Fusion multipler geometrischer und visueller Sensoren ein lokales Umweltmodell, das als Basis für die folgende Detektion von Hindernissen im lokalen Umfeld des Roboters dient. Das Ergebnis der Hindernisdetektion wird in Form einer lokalen Hinderniskarte repräsentiert, die in der darauf aufbauenden lokalen Pfadplanung genutzt wird, um einen kollisionsfreien Pfad zu planen, der mit der Kinematik des Bodenroboters umsetzbar ist.

Die Pfadverfolgung übernimmt daraufhin die Regelung von Geschwindigkeit und Lenkwinkel des Fahrzeugs und versucht dabei, ein minimales Abweichen vom geplanten Pfad zu erreichen.

Hinzukommt schließlich die reaktive Hindernisvermeidung, die eine wichtige Rolle am Ende der Verarbeitungskette des Gesamtsystems spielt. Sie kompensiert etwaige Ungenauigkeiten vorgelagerter Komponenten, die bspw. aufgrund von Messfehlern der beteiligte Sensoren entstehen und erlaubt zusätzlich das spontane Reagieren auf sich bewegende Hindernisse im Nahbereich des Roboters.

### 3.2.1 Lokalisierung

Der lokalen Lokalisierung kommt in der Systemarchitektur des autonomen Bodenroboters AMOR eine zentrale Bedeutung zu, da diese Komponente die jeweiligen Referenzkoordinatensysteme für Sensorinformationen, Szenen und Pfade ermittelt. Damit ist diese Komponente an einem Großteil der lokalen Komponenten des Bodenroboters mitbeteiligt. Die sorgfältige Wahl adäquater Methoden für die lokale Lokalisierung ist damit von entscheidender Bedeutung.

Wie in Kapitel 4.1 ausführlich beschrieben, wird in dieser Arbeit das weit verbreitete Prinzip der Koppelnavigation für die lokale Lokalisierung im Szenenkoordinatensystem des Bodenroboters AMOR eingesetzt [16]. Die gesuchte Position des Fahrzeugs wird, ausgehend von einer initialen Startpose, inkrementell zusammengesetzt bzw. fortgeführt. Die Orientierung der linearen Elemente wird mittels der Inertialsensorik des Roboters bestimmt. Die Länge der linearen Elemente wird üblicherweise durch Radencodier bzw. Odometer ermittelt, die durch die Messungen von Radumdrehungen auf die zurückgelegte Strecke des Fahrzeugs schließen lassen. Nicht befestigte Straßen und Wege mit anspruchsvollen Untergrundbedingungen können in diesem Zusammenhang eine große Herausforderung darstellen, da Radschlupf eine ernstzunehmende Fehlerquelle bedeutet (siehe Abb. 3.6).

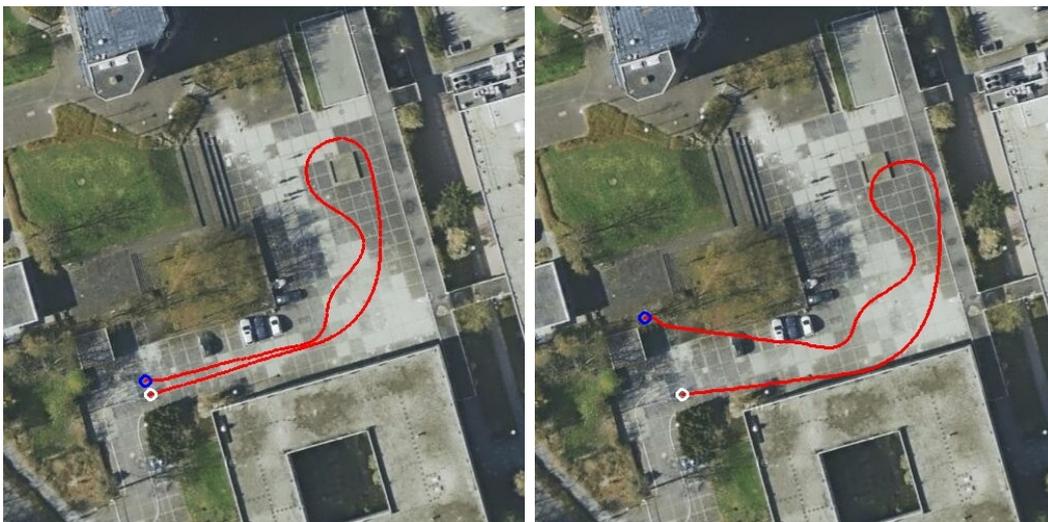


**Abbildung 3.6: Anspruchsvolle Bodenverhältnisse erzeugen Radschlupf bei einer autonomen Testfahrt des Roboters AMOR.**

Um diesem Problem entgegenzuwirken, werden zusätzlich zum Odometer zwei optische Eigenbau-Sensoren zur Messung der Bewegung des Fahrzeugs eingesetzt, die nicht vom Problem des Radschlupfs betroffen sind. Das Funktionsprinzip entspricht, vereinfacht gesagt, dem einer optischen Computermaus und wird in [173] detailliert besprochen. Die Fusion beider Messprinzipien (Odo-

meter mit Radencoder und optische Bewegungsmessung) erlaubt sowohl die Erkennung als auch die Minimierung von Positionierungsfehlern durch Rad-schlupf.

Auch visuelle Lokalisierungsmethoden eignen sich zur lokalen bzw. relativen Lokalisierung. Daher wurde im Rahmen dieser Arbeit auch diese Möglichkeit in Betracht gezogen, um die lokale Pose des Bodenroboters zu ermitteln. Evaluert wurde der Ansatz von Geiger et al. [64], der einer der aktuell leistungsfähigsten, echtzeitfähigen Algorithmen ist (siehe KITTI Vision Benchmark Suite<sup>2</sup>). Der Algorithmus basiert ausschließlich auf Kameradaten und der Quellcode ist verfügbar. Somit kann er als gute Referenz für die Evaluation des Konzepts der visuellen Odometrie im Vergleich zum klassischen Koppelnavigationsansatz dienen. Aufgrund der Rechenintensität und vor allem der fehlenden Robustheit wurde dieser Ansatz für die praktische Umsetzung der relativen Lokalisierung in dieser Arbeit jedoch verworfen (siehe Abb. 3.7).



(a) Klassische Koppelnavigation.

(b) Visuelle Odometrie.

**Abbildung 3.7: Vergleich der Lokalisierungsergebnisse bei klassischer Koppelnavigation bzw. visueller Odometrie während einer kurzen Testfahrt.** Copyright der Luftbilder: Google Inc.

<sup>2</sup>[http://www.cvlibs.net/datasets/kitti/eval\\_odometry.php](http://www.cvlibs.net/datasets/kitti/eval_odometry.php)

### 3.2.2 Umgebungsmodellierung

Die Umgebungsmodellierung befasst sich mit der Erzeugung eines Modells der - mit der Sensorik des Roboters erfassbaren - Umwelt des Bodenroboters. Dabei kommen sowohl multiple geometrische als auch visuelle Sensoren zum Einsatz, die durch verschiedene Methoden der Sensorfusion zu einem einheitlichen Modell zusammengeführt werden. Das Ergebnis der Umgebungsmodellierung ist ein geometrisches Oberflächenmodell, das mit visuellen Informationen angereichert wird. Konkret bedeutet dies die Färbung der erzeugten Oberfläche mit Daten aus verschiedenen visuellen Sensorenquellen.

Eine Reduktion des Umgebungsmodells auf zwei Dimensionen, wie sie oft im Innenbereich vorgenommen wird, ist gerade im nichturbanen Außenbereich nicht mehr ausreichend. Eine Erweiterung des Umgebungsmodells auf drei Dimensionen ist deshalb unumgänglich. Vereinfachungen wie etwa die Beschreibung von Boden und Wänden durch planare Oberflächen werden der Struktur der Umgebung im Außenbereich nicht gerecht. Aus diesem Grund wird das Umgebungsmodell konsequent in 3D umgesetzt. Dies bedingt gleichzeitig die Beschreibung des Robotersystems als Objekt mit sechs anstatt nur drei Freiheitsgraden.

Um den dafür erforderlichen Fusionsprozess erfolgreich durchführen und damit die Sensor-Rohdaten aufeinander abbilden zu können, müssen verschiedene Kalibrierungsschritte für die Sensoren realisiert werden. Im Fall der auf dem Bodenroboter montierten Sensoren gilt es zunächst, die statischen Transformationen zwischen den jeweiligen nativen Sensorkoordinatensystemen und dem fahrzeugfesten Koordinatensystem zu bestimmen. Dabei werden je Sensortyp verschiedene Kalibrierungsverfahren angewendet.

Die geometrischen Sensoren werden sowohl miteinander als auch mit dem Weltkoordinatensystem registriert. Unterschiedliche visuelle Sensoren (entzentratisch und katadioptrisch) bedürfen klassischerweise unterschiedlicher Registrierungsverfahren. Im Rahmen dieser Arbeit ist es jedoch gelungen, diesen Prozess zu generalisieren und eine einheitliche Methode zur Registrierung der genutzten unterschiedlichen visuellen Sensortypen mit den geometrischen Sensoren zu kreieren (siehe Kap. 5.3.3). Weiterhin sind die Sensordaten mithilfe der Daten der relativen Lokalisierung in das szenenfeste Koordinatensystem

zu transformieren, um die Sensordaten jeweils in einer Szene zusammenfassen zu können.



Abbildung 3.8: Mit Luftbilddaten texturiertes Umgebungsmodell des Bodenroboters.

Abschließend bleibt festzuhalten, dass die Texturierung des Oberflächenmodells des Bodenroboters mit Live-Luftbilddaten des Flugroboters in besonderem Maße hervorzuheben ist. Diese Form der kooperativen Umgebungsbeschreibung stellt aufgrund des Innovationsgrads und der Neuartigkeit des Ansatzes einen zentralen wissenschaftlichen Beitrag dieser Arbeit dar. Die Ermittlung der Transformation zwischen Fahrzeugkoordinatensystem des Bodenroboters und Sensorkoordinatensystem der Luftbildkamera bei sich bewegenden Robotern ist ein komplexes Problem, das nicht unter Zuhilfenahme der üblichen Lokalisierungs- bzw. Registrierungsmethoden gelöst werden kann. Daher wurde ein neues Verfahren entwickelt, das basierend auf Methoden der digitalen Bildverarbeitung eine schritthaltende Registrierung der Daten der Luftbildkamera mit dem Umgebungsmodell des Bodenroboters erreicht. So kann das Umgebungsmodell mit aktuellen Luftbilddaten texturiert werden (siehe

Abb. 3.8). Alle Details der in diesem Abschnitt kurz zusammengefassten Methoden zur Umgebungsmodellierung durch Sensorfusion werden in Kapitel 5 ausführlich dargelegt.

### 3.2.3 Pfadplanung

In Kapitel 3.1.2 wurde das in dieser Arbeit genutzte Verfahren zur globalen Pfadplanung und Navigation im erdfesten Koordinatensystem dargestellt. Die beschriebene Methode erzeugt jedoch nur einen groben Pfad, der als ungefähre Navigationsrichtlinie auf globaler Ebene zu sehen ist. Dieser Pfad zieht weder die genauen lokalen Gegebenheiten noch die kinematischen Fahrzeugeigenschaften in Betracht. Aus diesem Grund ist es unumgänglich, eine lokale Feinplanung durchzuführen. Die Effektivität einer Aufteilung in globale Grobplanung und lokale Feinplanung wurde in vielen Arbeiten nachgewiesen, die sich mit der Entwicklung von mobilen Robotern für die Außenbereichsrobotik befassen [94][175][31][180].

Die Entwicklung von lokalen Pfadplanungsalgorithmen für die Außenbereichsrobotik ist ein Bereich anhaltender Forschungsbemühungen. Gerade die Frage, wie Pfadplanungsalgorithmen zu entwerfen sind, die eine effiziente und dennoch erschöpfende Exploration einer unbekannt lokalen Szene erreichen, ist im unstrukturierten Außenbereichsszenario nicht endgültig geklärt. Es wurde daher in dieser Arbeit eine neue Methode der explorativen lokalen Pfadplanung entwickelt, die zum Ziel hat, alle Pfade zu ermitteln, die den Roboter aus der lokalen Szene entkommen lassen, also in noch nicht erkundete Bereiche führen. Weiterhin werden die kinematischen Eigenschaften des Roboters bei der Planung berücksichtigt, sodass sichergestellt ist, dass die geplanten Pfade auch tatsächlich von der Mechanik des Roboters ausgeführt werden können.

Die Methode der Rapidly Exploring Random Trees (RRT) nach LaValle [120] bietet in diesem Zusammenhang die ideale Basis für eine schnelle Exploration einer aktuellen lokalen Szene. Abbildung 3.9 zeigt verschiedene Wachstumsstufen eines einfachen RRT.

Die übliche Nutzung eines RRTs in der Pfadplanung sieht eine Start-Ziel-Suche vor. Das heißt, während des Aufbaus des Baums wird stets auf Basis einer manuell zu definierenden Metrik geprüft, ob ein vorher festgelegter Zielzustand

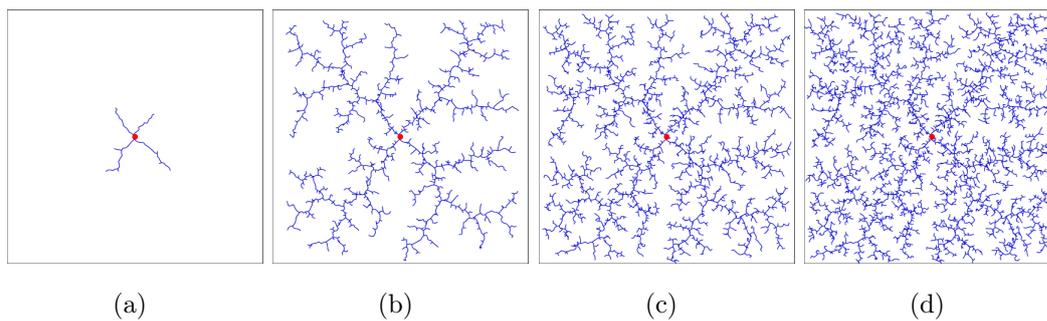


Abbildung 3.9: **Schritte im Wachstum eines Rapidly Exploring Random Trees.** Quelle: [http://msl.cs.uiuc.edu/rrt/gallery\\_2drdt.html](http://msl.cs.uiuc.edu/rrt/gallery_2drdt.html)

erreicht worden ist. Dieser Suchprozess kann beispielsweise durch simultanes Wachstum zweier RRTs vom Start- bzw. Zielzustand aus beschleunigt werden [102].

Dies ändert allerdings nichts an der Tatsache, dass im Fall des in dieser Arbeit vorherrschenden Explorationsszenarios vor der Durchführung der Planung keine Zielzustände ermittelt werden können. Die Zielzustände mit zugehörigen Pfaden sind vielmehr das Ergebnis des Planungsprozesses selbst. Zwar wäre es möglich, die Karte der lokalen Szene auf Übergänge von bekanntem zu unbekanntem Terrain zu untersuchen und dort Zielzustände zu platzieren, jedoch kann in diesem Fall nicht geklärt werden, ob diese Zielzustände auch tatsächlich erreichbar sind. Sowohl Hindernisse im Fahrzeugumfeld als auch die Fahrzeugkinematik machen dies in vielen Fällen unmöglich.

Daher wird in dieser Arbeit keine klassische Start-Ziel-Planung durchgeführt, sondern das Baumwachstum solange fortgeführt, bis eine ausreichende Abdeckung des gesamten bekannten lokalen Szenenbereichs erreicht worden ist. Eine Untersuchung der Knoten des Baums nach Beendigung der Wachstumsphase ermöglicht es, kollisionsfreie, kollidierende und explorierende Knoten zu unterscheiden (siehe Abb. 3.10).

Explorierende Knoten sind Endpunkte von Pfaden, die den Roboter in unbekanntes Terrain führen. Durch Bestimmung des explorierenden Knotens, der den Roboter den groben globalen Pfad am besten ausführen lässt, ist der lokale Pfad gefunden, den es auszuführen gilt. Eine detaillierte Beschreibung von Grundlagen und algorithmischen Details der in dieser Arbeit entwickelten Methode zur lokalen Pfadplanung finden sich in Kapitel 6.

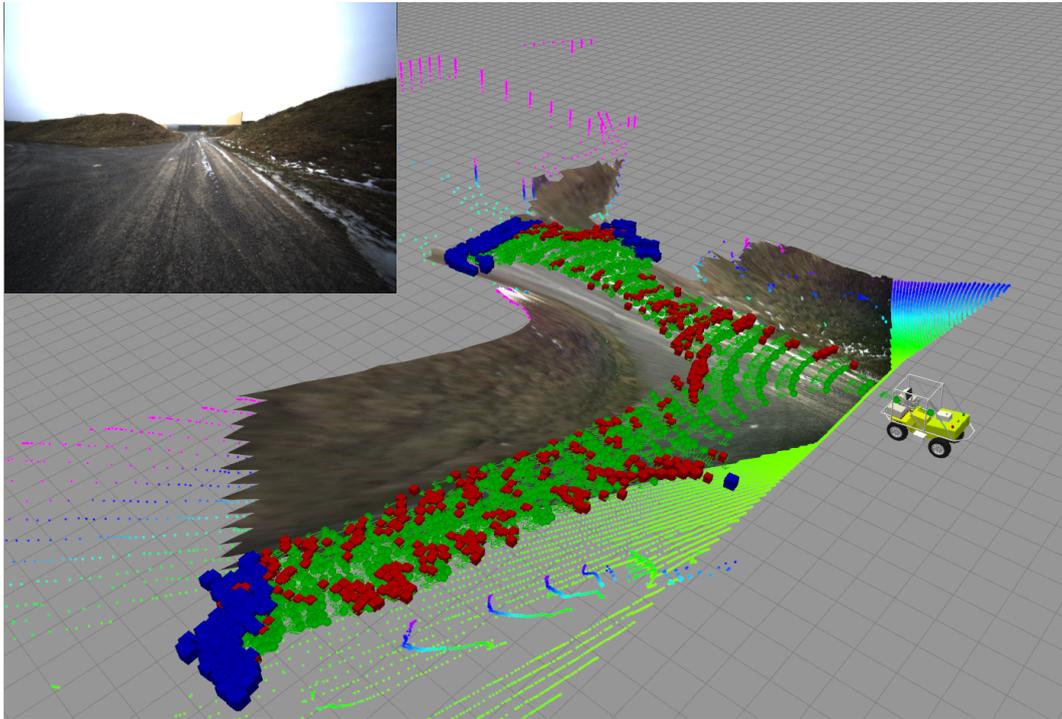


Abbildung 3.10: **Ergebnis der lokalen explorativen Pfadplanung in einer Außenbereichsszene.** Kollisionsfreie (grün), kollidierende (rot) und explorierende (blau) Knoten bilden jeweils die Endpunkte von ausführbaren lokalen Pfaden.

### 3.2.4 Pfadverfolgung

Wie in [177] in einem detaillierten Vergleich festgestellt wird, ist die Wahl der Methode zur Pfadverfolgung stark vom Einsatzszenario des ausführenden Fahrzeugs abhängig. Die hauptsächlichen Unterscheidungskriterien sind in diesem Zusammenhang die zu erwartende Geschwindigkeit des Fahrzeugs und die Beschaffenheit des zu verfolgenden Pfads (siehe Tabelle 3.1).

Bei niedrigen Geschwindigkeiten und Pfaddiskontinuitäten ist demnach als Methode der sogenannte *Pure Pursuit-Algorithmus* [4][40] zu favorisieren, der in die Kategorie der rein geometrischen Methoden fällt. Andere Methoden

<b>Pfadverfolgungsmethode</b>	<b>Bevorzugtes Anwendungsszenario</b>
Pure Pursuit [4][40]	Geringe Geschwindigkeiten und/oder nicht kontinuierliche Pfade
Stanley [193]	Kontinuierliches Fahren auf Schnellstraßen und/oder Einparkmanöver
Kinematisches Modell [45]	Kontinuierliche Einparkmanöver
Linear-quadratischer Regulator mit Vorsteuerungsfunktion [162]	Kontinuierliches Fahren mit hohen Geschwindigkeiten
Optimaler Vorschau-Regler [152][174]	Kontinuierliches Fahren auf Schnellstraßen bei relativ konstanter Geschwindigkeit

Tabelle 3.1: **Vergleich verschiedener Methoden zur Pfadverfolgung anhand ihrer Einsatzszenarios.** Quelle: [177]

sind für höhere Geschwindigkeiten bei kontinuierlichen Pfadverläufen entwickelt worden. Sie unterscheiden sich vor allen Dingen im zugrunde gelegten Modell, das ebenfalls entweder geometrisch [193], kinematisch [45] oder dynamisch [162][152][174] sein kann. Da aufgrund der unstrukturierten Umgebung im Einsatzszenario der Roboter in dieser Arbeit sowohl vergleichsweise niedrige Geschwindigkeiten als auch abrupte Änderungen in der Kontinuität der geplanten Pfade zu erwarten sind, wurde der *Pure Pursuit-Algorithmus* als passendste Methode der Pfadverfolgung gewählt.

Abbildung 3.11 fasst die relevanten Größen des *Pure Pursuit-Algorithmus* grafisch zusammen. Als Grundlage zur Beschreibung der Methode dient das Zweirad-Modell, das durch Zusammenlegen der Vorder- bzw. Hinterräder eines vierrädrigen Fahrzeugs mit Ackermann-Lenkung eine zulässige Vereinfachung der geometrischen Beschreibung erreicht. Das Grundprinzip der *Pure Pursuit-Methode* besteht in der Regelung des Lenkwinkels des Fahrzeugs, sodass dieses auf einen ständig aktualisierten Vorausschau-Punkt auf dem auszuführenden Pfad zusteuert. Der Vorausschau-Punkt entsteht durch den Schnitt der Kreisbahn, auf der sich das Fahrzeug bewegt, mit dem auszuführenden Pfad. Ziel ist es nun, stets diejenige Kreisbahn (bzw. den zugehörigen Lenkwinkel) zu ermitteln, die den Vorausschau-Punkt erzeugt, der in einer vorher definierten festen

Vorausschau-Distanz zur Hinterachse des Fahrzeugs liegt. Die Vorausschau-Distanz entspricht der Länge des Vorausschau-Vektors, der den Mittelpunkt der Hinterachse mit dem Vorausschau-Punkt verbindet.

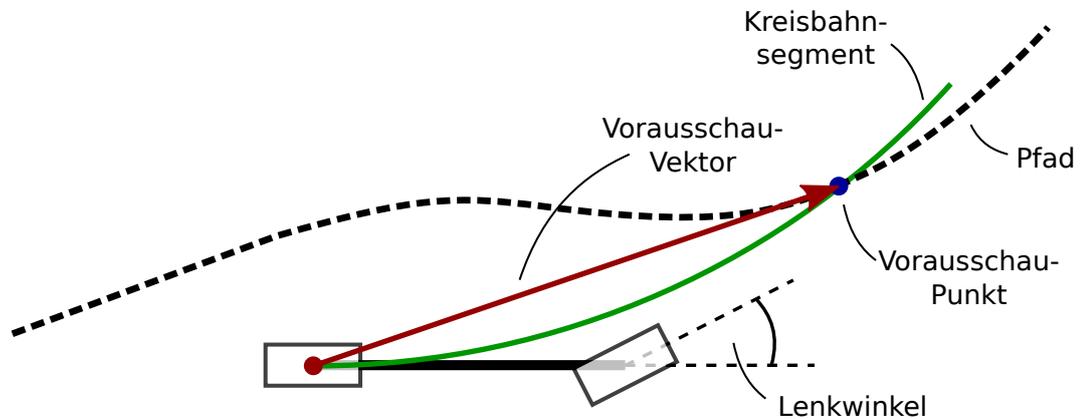


Abbildung 3.11: Pfadverfolgung nach der *Pure Pursuit*-Methode.

Der Wahl der Vorausschau-Distanz kommt bei Nutzung dieser Methode der Pfadverfolgung eine zentrale Rolle zu. Es gilt, diesen Parameter abhängig von den Fahrzeugeigenschaften und dem gewünschten, an das Einsatzszenario angepassten Fahrzeugverhalten einzustellen. Abbildung 3.12 verdeutlicht in diesem Zusammenhang die Auswirkung unterschiedlicher Vorausschau-Distanzen auf das Fahrzeugverhalten. Zu geringe Vorausschau-Distanzen erzeugen schnell ein Schwingen des Fahrzeugs um den Pfad. Zu große Vorausschau-Distanzen bedeuten ein zu langsames Annähern der Fahrzeugposition an den geplanten Pfad und häufiges Schneiden von Pfadelementen mit hoher Krümmung.

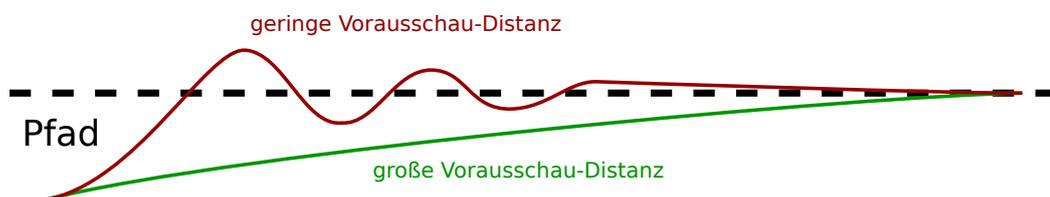


Abbildung 3.12: Auswirkungen verschiedener Vorausschau-Distanzen auf das Fahrzeugverhalten während der Pfadverfolgung mit der *Pure Pursuit*-Methode.

### 3.2.5 Reaktive Hindernisvermeidung

Die in den zwei vorherigen Abschnitten beschriebenen Methoden der lokalen Pfadplanung und -verfolgung erlauben es, kollisionsfreie Pfade zu planen und auszuführen. Allerdings kann nicht davon ausgegangen werden, dass der jeweils geplante Pfad während der gesamten Dauer der Ausführung kollisionsfrei bleibt, da die Häufigkeit der Aktualisierung des lokalen Umgebungsmodells und damit des aktuell geplanten Pfads zu gering ist, um Hindernisse zu erfassen, die sich schnell bewegen.

Weiterhin sind sowohl Sensorik als auch Aktorik des Roboters nicht perfekt. Daher kommt es zu Ungenauigkeiten bei der Erzeugung und Ausführung von kollisionsfreien Pfaden. Beide Aspekte stellen ein ernstzunehmendes Problem für die sichere Operation des Robotersystems dar.

Aus diesem Grund wurde eine weitere Systemkomponente entwickelt, die reaktive Hindernisvermeidung. Sie überprüft am Ende der Verarbeitungskette die zuvor bestimmte Soll-Trajektorie ständig auf Gültigkeit. Wird festgestellt, dass die Soll-Trajektorie nicht kollisionsfrei ausführbar ist, wird eine möglichst kostengünstig auszuführende Alternativ-Trajektorie ermittelt. Damit reiht sich das Verfahren in eine ganze Reihe von Arbeiten ein, die eine Menge vordefinierter Trajektorien auf deren Tauglichkeit für die Ausführung eines vorher geplanten Pfads überprüfen [95][25][115][204].

Der implementierte Algorithmus zur reaktiven Hindernisvermeidung fusioniert im ersten Schritt die Daten verschiedener Sensoren (Laserscanner und Ultraschallsensoren) in einer lokalen Gitterkarte, die deutlich kleinere Ausmaße hat als die Hinderniskarte, die für die lokale Pfadplanung verwendet wird. Im Gegenzug wird diese Gitterkarte jedoch deutlich häufiger aktualisiert. Da die Sensoren bezüglich des Fahrzeugkoordinatensystems registriert sind (siehe Kapitel 5.2.4), liegen die Daten in Form einer Punktwolke - bestehend aus Abtastpunkten der Umgebung - vor, die bezüglich des Fahrzeugkoordinatensystems definiert ist. Aufgrund der relativ geringen Messfrequenz der geometrischen Sensoren (Zeilen-Laserscanner und Ultraschallsensoren) hat die Punktwolke nur wenige Elemente. Durch Zuordnung der Abtastpunkte zu passenden Gitterzellen entsteht eine dünn besetzte Gitterkarte, wobei jeder Zelle der maximale absolute Höhenwert über Grund zugeordnet wird. Zur Unterscheidung von freien

Zellen und Hinderniszellen wird durch Anwendung einer Schwellwertfunktion mit binärem Ergebniswert je Gitterzelle die Karte von einer Höhenkarte in eine Hinderniskarte transformiert. Überschreitet der Höhenwert einer Zelle den definierten Schwellwert, wird die Zelle als Hinderniszelle markiert. Die restlichen Zellen werden als frei, also befahrbar, markiert.

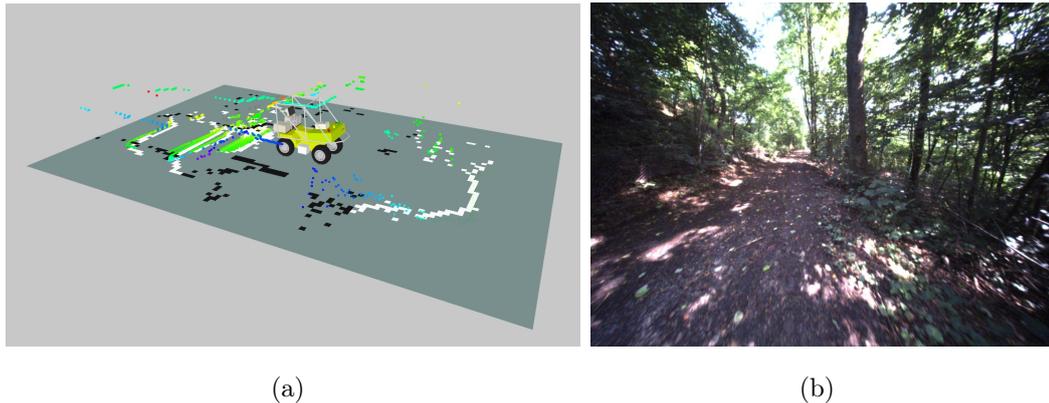


Abbildung 3.13: **Hinderniskarte der reaktiven Hindernisvermeidung.** Karte mit integrierten Umgebungsmesspunkten in einer Beispielszene (a). Schwarze Zellen kennzeichnen Hinderniszellen. Weiße Zellen sind frei befahrbare Zellen. Graue Zellen werden aufgrund von fehlenden Sensorinformationen nicht ausgewertet und ignoriert. Kameraaufnahme der Beispielszene (b).

Die Karte wird mit einer hohen Frequenz erneuert, wobei nur die Messungen seit der letzten Erneuerung berücksichtigt werden. Das somit fehlende "Gedächtnis" der reaktiven Hindernisvermeidung hat sich, bei Hindernissen im äußersten Nahbereich des Bodenroboters, als problematisch herausgestellt. Dies ist in der Tatsache begründet, dass die eingesetzten Sensoren (Laserscanner und Ultraschallsensoren) aufgrund ihrer Montageposition und -orientierung im unmittelbaren Nahbereich des Roboters "blind" sind. Alternative Montagepositionen der Sensoren würden die Vorausschau und damit die Zeit, die zur Verfügung steht, um auf ein Hindernis zu reagieren, so stark verringern, dass nur sehr geringe Geschwindigkeiten des Bodenroboters realisiert werden könnten. Da eine Verringerung der maximal möglichen Geschwindigkeit des Roboters nicht gewünscht ist, wird zusätzlich die Information aus der Hinderniskarte der Pfadplanung in die Karte der reaktiven Hindernisvermeidung integriert.

Durch Projektion von Hindernissen aus der Hinderniskarte der Pfadplanung in die Hinderniskarte der reaktiven Hindernisvermeidung kann dem Problem des "Vergessens" von Hindernissen aufgrund der hohen Aktualisierungsrate im äußersten Nahbereich des Roboters begegnet werden.

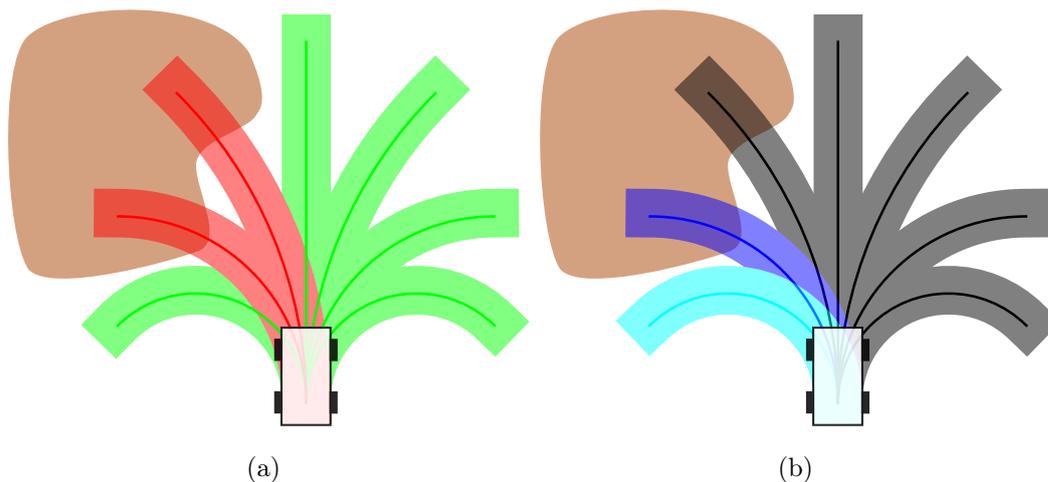


Abbildung 3.14: **Trajektorienanalyse in der reaktiven Hindernisvermeidung.** Überlagerung von Trajektorien und Hinderniskarte zur Unterscheidung von kollisionsfreien (grün) und kollidierenden (rot) Trajektorien (a). Auswahl der günstigsten kollisionsfreien Alternativ-Trajektorie (cyan) bei kollidierender Soll-Trajektorie (blau) (b).

Auf Basis der erzeugten Hinderniskarte wird nun eine Analyse der, aus verschiedenen Lenkwinkeln entstehenden, möglichen Trajektorien des Bodenroboters durchgeführt. Die Hinderniskarte wird mit einer vordefinierten Menge an ausführbaren Trajektorien überlagert. Die Trajektorien, die dabei Hinderniszellen überlagern, werden als kollidierend markiert, die restlichen als kollisionsfrei (siehe Abb. 3.14 (a)). Ist die von der Pfadverfolgungs-Komponente vorgeschlagene Soll-Trajektorie als kollidierend markiert, muss eine alternative, kollisionsfreie Trajektorie ermittelt werden. Es wird die kollisionsfreie Trajektorie gewählt, welche die geringste Lenkwinkeländerung erfordert (siehe Abb. 3.14 (b)).

### 3.3 Missionsplanung

Die in den bisherigen Abschnitten dieses Kapitels beschriebenen Komponenten realisieren jeweils Aufgaben, die nahezu ausschließlich von funktionaler Bedeutung für das Gesamtsystem sind. Die erfolgreiche Umsetzung der autonomen Operation des Roboterteams im Explorationsszenario erfordert jedoch auch verwaltende Tätigkeiten. Zu diesem Zweck wurde im Rahmen dieser Arbeit die Architektur CAPTAIN<sup>3</sup> geschaffen, welche die Sequenzierung von Unteraufgaben eines abstrakten Missionsszenarios organisiert. Gleichzeitig wird dadurch die Arbitrierung des Zugriffs auf die Aktorik des Bodenroboters geregelt.

Missionen in der vorgeschlagenen Architektur werden durch hierarchische Zustandsautomaten beschrieben, die in Form eines Baums von endlichen Automaten implementiert sind (ähnlich dem Grundansatz der SMACH-Architektur von Bohren und Cousins [22]). Die hierarchische Organisation von Missionen ermöglicht es, Missionen in Untermissionen von geringerem Abstraktionsniveau zu unterteilen. So kann die Komplexität der Gesamtmission deutlich besser überschaut und beherrscht werden und zusätzlich können Untermissionen in anderen Missionen wiederverwendet werden. Der Entwicklungsprozess von komplexen Missionen wird dadurch stark vereinfacht und bleibt sehr transparent.

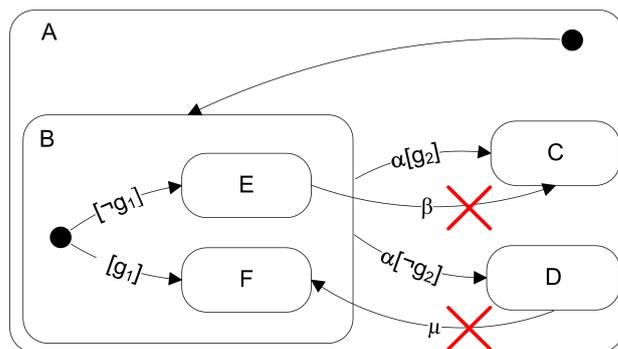
Zur Beschreibung der Zustandsautomatenhierarchie wird auf die Statechart-Notation nach Harel [75] zurückgegriffen, die es im Gegensatz zur klassischen Notation für endliche Zustandsautomaten erlaubt, hierarchische Zusammenhänge zwischen Zustandsautomaten auszudrücken. Die Statechart-Notation sieht dafür zwei Zustandstypen vor, zusammengesetzte (A und B in Abb. 3.15) und atomare (C, D, E und F in Abb. 3.15). Zusammengesetzte Zustände werden auf ihrer Hierarchieebene als atomare Zustände behandelt, ermöglichen jedoch die Realisierung einer Hierarchie, da diese wiederum beliebig viele atomare und zusammengesetzte Zustände beinhalten können.

Zustandsübergänge werden durch Ereignisnachrichten ( $\alpha$ ,  $\beta$  und  $\mu$  in Abb. 3.15) ausgelöst, die von Zuständen der Missionshierarchie selbst oder von allen

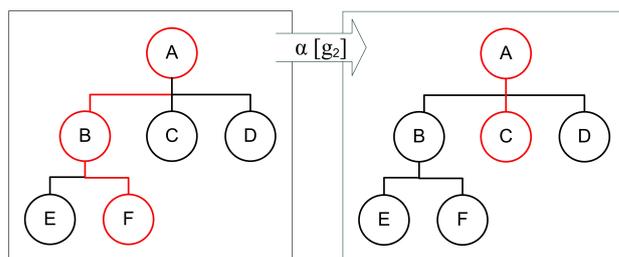
---

<sup>3</sup>kurz für: Control Architecture Providing Task Arbitration and Sequencing for Intelligent Robot Navigation

übrigen Komponenten des Systems generiert werden können. Ereignisnachrichten sind also als Indikatoren für eine relevante Änderung der aktuellen Missionsgegebenheiten zu interpretieren. Sie bedürfen eines veränderten Verhaltens des Roboters. Eingehende Ereignisnachrichten werden im Allgemeinen sequenziell abgearbeitet. Sie können mit Bedingungen versehen ( $g_1$  und  $g_2$  in Abb. 3.15) oder bei hoher Dringlichkeit (bspw. zur Durchführung eines Notbremsmanövers) als höher priorisiert markiert werden. Im Unterschied zur Statechart-Notation sind in der CAPTAIN-Architektur Zustandsübergänge nur auf der gleichen Hierarchieebene zugelassen. Ist ein Zielzustand ein zusammengesetzter Zustand, entscheidet dieser stets selbst, welcher seiner Unterzustände aktiv wird. Die Ereignisnachrichten  $\beta$  und  $\mu$  wurden zur Verdeutlichung verbotener Zustandsübergänge zur Abb. 3.15 hinzugefügt. So wird eine strikte Kapselung von Hierarchieebenen gewährleistet. Weiterhin transportie-



(a)



(b)

Abbildung 3.15: **Hierarchische Zustandsautomaten** in Statechart-Notation (a). **Beispielhafter Zustandsübergang** beim Eintreffen einer Ereignisnachricht (b). [101]

ren Ereignisnachrichten niemals zusätzliche für die Ausführung der Mission

relevante Daten. Das Zusammentragen von relevanten Daten ist Aufgabe des aktiven Zustands der aktuell ausgeführten Mission, was die Struktur der Ereignisnachrichten einfach und damit effizient belässt. Es gibt grundsätzlich nur einen aktiven Zustand in der Missionshierarchie. Nur dieser aktive Zustand hat Zugriff auf die Aktorik des Roboters. Diese Grundbedingung verhindert den konkurrierenden Zugriff auf die aktorischen Komponenten des Roboters, ohne dabei den modularen Aufbau der funktionalen Komponenten des Systems zu beeinträchtigen.

Die Abbildungen 3.16 und 3.17 geben abschließend einen anschaulichen Einblick in die Strukturierung von Missionen mit CAPTAIN. Hierzu wird eine zweistufigen Hierarchie von Zustandsautomaten verwendet, welche die Mission der autonomen Navigation einschließlich einer Untermission zur Behandlung einer Sackgassen-Situation implementiert.

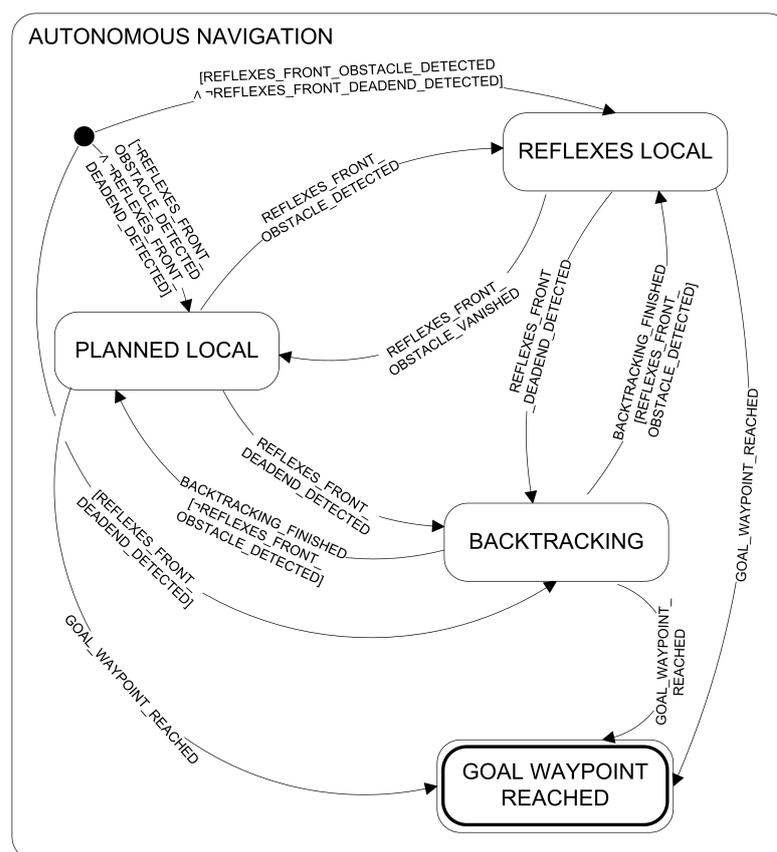


Abbildung 3.16: Hierarchische Zustandsautomaten zur Umsetzung der Mission: Autonome Navigation. Hauptmission.

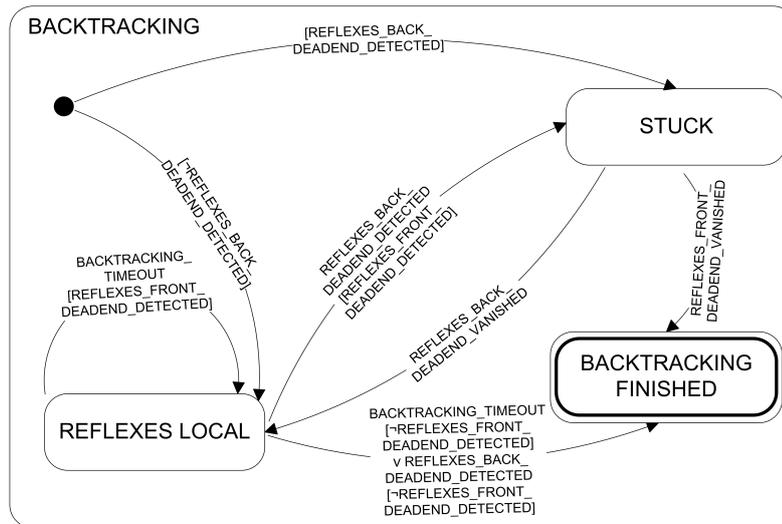


Abbildung 3.17: **Hierarchische Zustandsautomaten zur Umsetzung der Mission: Autonome Navigation. Untermission: Sackgassen-Behandlung.**

# Kapitel 4

## Lokalisierung im Außenbereich

Die Lokalisierung eines mobilen Roboters ist eine fundamental wichtige Komponente, auf deren Ergebnisse zahlreiche folgende Komponenten angewiesen sind. Sie befasst sich mit der Bestimmung bzw. mit der Schätzung der Position und der Orientierung, also der Pose oder Raumlage, des Roboters. Die mathematische Repräsentation von Position und Orientierung wird durch die Anzahl der Freiheitsgrade definiert, die für die Beschreibung des Roboters berücksichtigt werden. Ein als starrer Körper beschriebener Roboter hat sechs Freiheitsgrade, wenn er sich frei in einer dreidimensionalen Welt bewegen kann. Daher kann die Pose des Roboters mit einem 6-Tupel

$$\rho = (p_x, p_y, p_z, \phi_x, \phi_y, \phi_z), \quad (4.1)$$

beschrieben werden. Das 6-Tupel ist aus den zwei Tripeln  $(p_x, p_y, p_z)$  und  $(\phi_x, \phi_y, \phi_z)$  zusammengesetzt. Sie beschreiben die Position mit drei Translationen entlang der Achsen des zugrunde gelegten kartesischen Koordinatensystems und die Orientierung mit drei Rotationen um die Koordinatensystemsachsen. Alternativ zur vollständigen Beschreibung der Roboterpose mit einem 6-Tupel wird aus Gründen der Optimierung der Berechnungskomplexität häufig eine Reduktion auf drei Freiheitsgrade durchgeführt. Diesem Ansatz liegt der vornehmlich aus der Innenbereichsrobotik stammende Gedanke zugrunde, dass sich der Boden, auf dem sich der Roboter bewegt, lokal stark vereinfacht als Ebene beschreiben lässt. Der Roboter bewegt sich also in dieser

Bodenebene und die Orientierung kann daraufhin mit einem einzelnen Winkel beschrieben werden, der eine Rotation um die Normale der Bodenebene definiert. Im Fall der Reduktion auf drei Freiheitsgrade kann also die Pose eines Roboters mit einem Tripel

$$\rho' = (p_x, p_y, \phi) \quad (4.2)$$

beschrieben werden. Meist ist diese Form der Beschreibung allerdings gerade im Außenbereich nicht ausreichend, um die Komplexität der Umgebung adäquat zu repräsentieren, und daher nicht praktikabel einsetzbar. Eine Prüfung der Zulässigkeit dieser Vereinfachung für einzelne Anwendungsfälle im Außenbereich ist jedoch trotzdem lohnenswert, da die Einsparung von Rechenzeit die fehlenden Freiheitsgrade und die damit zusammenhängende vereinfachte Beschreibung von Roboter und Umwelt aufwiegen kann.

## Koordinatensysteme

Für die Beschreibung der Position eines Objektes auf oder über der Erdoberfläche existieren verschiedene Koordinatendefinitionen. Hier werden kartesische und ellipsoidische Koordinaten unterschieden.

Das sogenannte Earth-centered Earth-fixed Koordinatensystem (ECEF) ist ein weit verbreitetes kartesisches Erdkoordinatensystem. Wie der Name bereits nahelegt, liegt der Ursprung des Koordinatensystems im Erdmittelpunkt. Die positive Z-Achse entspricht der Rotationsachse der Erde und durchsticht die Erdkugel beim Nordpol. Die Kreise, die durch beide Pole laufen, werden Meridiane oder Längengrade genannt, wobei der sogenannte Nullmeridian durch Greenwich in England verläuft. Neben den Meridianen ist der Äquator wichtig, um das ECEF-Koordinatensystem zu definieren. Der Äquator ist der Größte der Schnittkreise, die entstehen, wenn man alle Ebenen mit der Erdkugel schneidet, die die Rotationsachse der Erde als Ebenennormale besitzen. Die Schnittkreise werden Breitengrade genannt. Die positive X-Achse des ECEF-Koordinatensystem zeigt in Richtung des Schnittpunktes von Nullmeridian und Äquator. Die positive Y-Achse wird aus dem Kreuzprodukt von X-Achse und Z-Achse konstruiert.

Das ECEF-Koordinatensystem ist nicht sehr intuitiv zu verwenden, da beispielsweise nicht direkt ersichtlich ist, ob sich ein Punkt im Erdinnern, auf der Erdoberfläche oder über der Erdoberfläche befindet. Daher werden deutlich häufiger ellipsoidische Koordinatensysteme verwendet. Ellipsoidische Koordinaten basieren auf der Darstellung eines Punktes auf der Erdoberfläche durch die Angabe von geographischer Länge und Breite. Die geographische Länge entspricht dem Längengrad, die geographische Breite entspricht dem Breitengrad, auf dem sich der Punkt befindet. So können alle Punkte auf der Erdoberfläche eindeutig beschrieben werden. Falls gewünscht, kann die orthometrische Höhe der Beschreibung hinzugefügt werden, sodass auch Punkte über der Erdoberfläche definiert werden können. Die orthometrische Höhe gibt dabei den Abstand eines Punktes zur Erdoberfläche in Lotrichtung an. Das Lot zeigt dabei in Richtung des Erdmittelpunktes. Zur finalen Bestimmung eines Punktes auf der Erdoberfläche wird ein geometrisches Modell zugrunde gelegt. Die Form der Erde entspricht keiner idealen Kugel, sondern vielmehr einem Rotationsellipsoid. Es sind verschiedene Versuche unternommen worden, die Parameter dieses Rotationsellipsoids zu bestimmen. Der wohl gängigste Referenzellipsoid in diesem Zusammenhang entstammt dem World Geodetic System 1984 (WGS 84) [176]. Dieser Referenzellipsoid wird unter anderem für die Bestimmung von Positionen durch das GPS verwendet.

Auch wenn ellipsoidische Koordinaten deutlich intuitiver zu verwenden sind als ein erdzentriertes, kartesisches Koordinatensystem, ist deren Nutzung in Kombination mit den meist kartesischen, lokal definierten Koordinatensystemen für Sensordaten oder Roboterbewegungen ebenfalls nur bedingt praktikabel. Das Universal Transversal Mercator (UTM) Koordinatensystem [72] bietet in diesem Zusammenhang eine komfortable Lösung, welche die Vorteile eines ellipsoidischen Koordinatensystems hat und gleichzeitig kartesische Koordinaten nutzt. Das UTM-System teilt den Erdellipsoid in 6 Grad breite Streifen auf und projiziert diese mithilfe einer transversalen Mercator-Projektion auf eine Ebene. Damit wird eine Abbildung auf kartesische Koordinaten erreicht, wobei die Polkappen davon ausgenommen sind und mittels einer anderen Abbildung, der Universalen Polaren Stereografischen Projektion (UPS) [72], projiziert werden. Das UTM-System wird in dieser Arbeit für die Beschreibung der Position der eingesetzten Roboter im erdfesten Koordinatensystem verwendet.

Ein weiteres kartesisches Koordinatensystem, das intuitiver zu verwenden ist als das ECEF-Koordinatensystem, ist das sogenannte Eastern-Northern-Up Koordinatensystem (ENU). ENU ist im Vergleich zu den bisher beschriebenen Koordinatensystemen kein Koordinatensystem mit einem festen Ursprung, sondern vielmehr eine Koordinatendefinition, die auf jeden Punkt  $P$  auf der Erdoberfläche angewendet werden kann und damit ein lokales Koordinatensystem mit  $P$  als Ursprung erzeugt. Konkret wird das lokale Koordinatensystem aus der Tangentialebene zur Erdoberfläche mit dem Berührungspunkt  $P$  erzeugt.

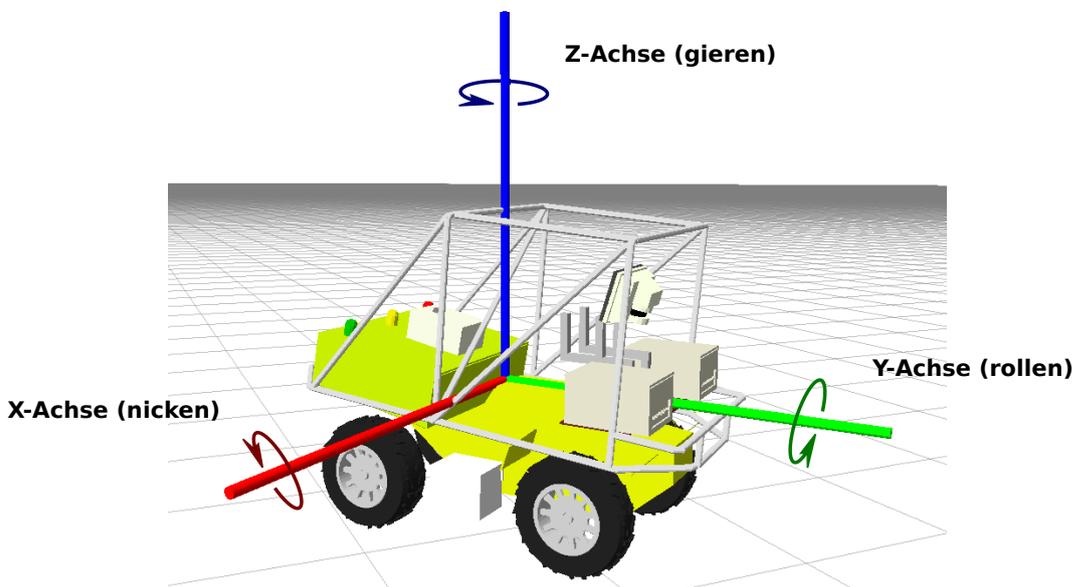


Abbildung 4.1: Koordinatensystem nach der ENU-Konvention am Beispiel des fahrzeugfesten Koordinatensystems des Roboters AMOR.

Die positive X-Achse, die in dieser Tangentialebene liegt, zeigt in Richtung Osten, die positive Y-Achse zeigt in Richtung Norden und die Z-Achse schließlich steht orthogonal auf der Tangentialebene, entspricht also der Normalen und zeigt weg vom Erdmittelpunkt. Dieses Koordinatensystem kommt in dieser Arbeit durchgehend für die Definition und Beschreibung von Sensordaten und Fahrzeugposen zum Einsatz (siehe Abb. 4.1).

## Beschreibung der Orientierung starrer Körper im dreidimensionalen Raum

Im vorhergehenden Abschnitt wurden verschiedene Koordinatensysteme besprochen und damit wurde die Beschreibung der Position eines starren Körpers im Raum erörtert. Zur Beschreibung der Orientierung existieren in der Mathematik verschiedene Ansätze. An dieser Stelle sollen zwei dieser Ansätze betrachtet werden, die in dieser Arbeit Anwendung finden - Eulersche Winkel und Quaternionen [48].

Die erste Darstellungsform, Eulersche Winkel, definiert eine Sequenz, bestehend aus drei aufeinanderfolgenden Rotationen, durch welche die Orientierung eines starren Körpers im dreidimensionalen Raum beschrieben werden kann. Prinzipiell wird so die Transformation von einem raumfesten in ein körperfestes Koordinatensystem beschrieben. Die Rotationssequenz bildet sich dabei aus drei einzelnen Rotationen, die jeweils durch einen Drehwinkel und die Achse um die gedreht definiert werden. Rotiert wird jeweils um eine der Achsen des aktuellen Koordinatensystems. Aktuell deshalb, weil nur die erste Rotation der Sequenz um eine der Achsen des raumfesten Koordinatensystems durchgeführt wird. Da die Rotationsachsen mitgedreht werden, werden die zwei folgenden Rotationen um die durch die vorherige Rotation transformierten Achsen ausgeführt. Jede der Rotationen wird üblicherweise durch eine Rotationsmatrix dargestellt. Da die Hintereinanderausführung von Rotationen durch die Multiplikation der einzelnen Rotationsmatrizen erreicht wird, ist aufgrund der fehlenden Kommutativität der Matrixmultiplikationsoperation auf die richtige Reihenfolge der einzelnen Transformationen zu achten. Einige Sequenzen haben sich dabei als oft verwendete Standard-Sequenzen durchgesetzt. Die x- bzw. y-Konvention bezeichnet die Sequenzen, in denen zunächst um die Z-Achse, dann um die rotierte X- bzw. Y-Achse und schließlich um die zweimal rotierte Z-Achse gedreht wird.

Daneben findet gerade in der Fahrzeug- und Luftfahrttechnik die sogenannte Roll-Nick-Gier<sup>1</sup>-Konvention häufig Anwendung. Diese Konvention ist dadurch charakterisiert, dass Rotationen um alle drei Koordinatenachsen enthalten sind, was sich gut auf das dreiachsige Messprinzip von Sensoren zur Raum-

---

<sup>1</sup>engl.: roll-pitch-yaw

lagemessung abbilden lässt. Aus diesem Grund bildet diese Konvention auch in dieser Arbeit eine der zwei Grundlagen zur Beschreibung der Orientierung starrer Körper im dreidimensionalen Raum. Die zugehörige Rotationsmatrix bildet sich wie folgt aus den drei Lagewinkeln  $\phi_x$ ,  $\phi_y$  und  $\phi_z$  der Fahrzeugpose  $\rho$ . Die resultierende Rotationsmatrix  $R_{\text{RNG}}$  hat die folgende Form:

$$R_{\text{RNG}}(\phi_x, \phi_y, \phi_z) = R_R(\phi_x)R_N(\phi_y)R_G(\phi_z) = \begin{pmatrix} c_{\phi_y}c_{\phi_z} & c_{\phi_y}s_{\phi_z} & -s_{\phi_y} \\ s_{\phi_x}s_{\phi_y}c_{\phi_z} - c_{\phi_x}s_{\phi_z} & s_{\phi_x}s_{\phi_y}s_{\phi_z} + c_{\phi_x}c_{\phi_z} & c_{\phi_y}s_{\phi_x} \\ c_{\phi_x}s_{\phi_y}c_{\phi_z} + s_{\phi_x}s_{\phi_z} & c_{\phi_x}s_{\phi_y}s_{\phi_z} - s_{\phi_x}c_{\phi_z} & c_{\phi_y}c_{\phi_x} \end{pmatrix}, \quad (4.3)$$

wobei die Abkürzungen  $s_{\phi_x}$ ,  $c_{\phi_x}$  usw. für  $\sin \phi_x$ ,  $\cos \phi_x$  usw. genutzt werden. Der zweite Ansatz der Darstellung basiert auf Quaternionen. Der 4-dimensionale reelle Vektorraum

$$\mathbb{H} := \{q = q_0 + i \cdot q_1 + j \cdot q_2 + k \cdot q_3 \mid q_0, q_1, q_2, q_3 \in \mathbb{R}\} \quad (4.4)$$

definiert die sogenannten Hamiltonschen Quaternionen.  $i$ ,  $j$  und  $k$  sind imaginäre Einheiten, wie sie auch in der Definition des Zahlbereichs  $\mathbb{C}$  der komplexen Zahlen verwendet werden. Der Vektorraum der Hamiltonschen Quaternionen ist somit als Erweiterung des Vektorraums der komplexen Zahlen zu sehen. Ein Quaternion  $q$  ist durch

$$q = q_0 + q_1i + q_2j + q_3k \quad (4.5)$$

definiert, wobei gilt

$$i^2 + j^2 + k^2 = -1. \quad (4.6)$$

In Vektordarstellung ist das Quaternion  $q$  ein 4-elementiger Vektor

$$q = \begin{pmatrix} q_0 \\ q_{1:3} \end{pmatrix}. \quad (4.7)$$

Die Konjugierte  $\bar{q}$ , die Norm  $\|q\|$  und die Inverse  $q^{-1}$  des Quaternion  $q$  ist

definiert durch

$$\bar{q} = \begin{pmatrix} q_0 \\ -q_{1:3} \end{pmatrix}, \quad (4.8)$$

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}, \quad (4.9)$$

$$q^{-1} = \frac{\bar{q}}{\|q\|}. \quad (4.10)$$

Der Bezug zur Darstellung von Rotationen im dreidimensionalen Raum wird durch die Einheitsquaternionen hergestellt. Ein Einheitsquaternion  $q$  ist ein Quaternion mit

$$\|q\| = 1. \quad (4.11)$$

Die Rotation eines Vektors  $v \in \mathbb{R}^3$  im dreidimensionalen Raum kann mit dem Einheitsquaternion  $q$  wie folgt dargestellt werden

$$\check{v}' = q \cdot \check{v} \cdot \bar{q}, \text{ mit } \check{v} = \begin{pmatrix} 0 \\ v \end{pmatrix} \mid \check{v} \in \mathbb{H}. \quad (4.12)$$

Anschaulich stellt ein Einheitsquaternion im Zusammenhang mit der Darstellung von Rotationen im dreidimensionalen Raum die Rotation eines Vektors mit dem Winkel  $\phi$  um eine beliebige Rotationsachse beschrieben durch den Einheitsvektor  $a = (a_x a_y a_z)^T$  dar. Aus Rotationsachse  $a$  und Winkel  $\phi$  wird das zugehörige Quaternion  $q$  mit

$$q = \begin{pmatrix} \cos \frac{\phi}{2} \\ a_x \sin \frac{\phi}{2} \\ a_y \sin \frac{\phi}{2} \\ a_z \sin \frac{\phi}{2} \end{pmatrix} \quad (4.13)$$

konstruiert. Diese Konstruktionsmethode ist allerdings oft nur schwer auf die auf einem Roboter verwendete Sensorik abzubilden, die zur Bestimmung der Raumlage verwendet wird. Die entsprechende Sensorik misst Beschleunigungen

oder Winkelgeschwindigkeiten entlang oder um drei feste orthogonal zueinander stehende Koordinatenachsen. Alle Eulerwinkel-Sequenzen können jedoch problemlos in Einheitsquaternionen überführt werden. Für die Gier-Nick-Roll-Konvention ergibt sich bspw. das Quaternion

$$q = \begin{pmatrix} \cos \frac{\phi_x}{2} \cos \frac{\phi_y}{2} \cos \frac{\phi_z}{2} + \sin \frac{\phi_x}{2} \sin \frac{\phi_y}{2} \sin \frac{\phi_z}{2} \\ -\cos \frac{\phi_x}{2} \sin \frac{\phi_y}{2} \sin \frac{\phi_z}{2} + \cos \frac{\phi_y}{2} \cos \frac{\phi_z}{2} \sin \frac{\phi_x}{2} \\ \cos \frac{\phi_x}{2} \cos \frac{\phi_z}{2} \sin \frac{\phi_y}{2} + \sin \frac{\phi_x}{2} \cos \frac{\phi_y}{2} \sin \frac{\phi_z}{2} \\ \cos \frac{\phi_x}{2} \cos \frac{\phi_y}{2} \sin \frac{\phi_z}{2} - \sin \frac{\phi_x}{2} \cos \frac{\phi_z}{2} \sin \frac{\phi_y}{2} \end{pmatrix}. \quad (4.14)$$

Die durch ein Quaternion beschriebene Rotation ist auch mit einer Rotationsmatrix darstellbar. Die passende Rotationsmatrix  $R_q$  zum Quaternion  $q$  wird durch

$$R_q = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \quad (4.15)$$

definiert.

Im Rahmen dieser Arbeit wurde besonderer Wert darauf gelegt, Rotationen sehr präzise über kurze Zeiträume, bezogen auf ein lokales Referenzkoordinatensystem ("lokale Szene"), nachvollziehen zu können. Ziel ist es, bei der Integration von zeitlich aufeinanderfolgenden Sensormessungen in ein konsistentes Umgebungsmodell möglichst hohe Genauigkeiten erreichen zu können. Die Motivation hinter dieser Vorgehensweise wird in Kapitel 5.1 detailliert erörtert. Aus diesem Grund wird nun abschließend eine weitere Methode zur Konstruktion eines Quaternionen beschrieben. Sie basiert auf der direkten Integration von Winkelgeschwindigkeiten, ohne einen Rechenweg über Eulerwinkel. So werden Singularitäten vermieden und die benutzte Eulerwinkel-Sequenz muss nicht zusätzlich zu den drei Eulerwinkeln kommuniziert werden.

Sei  $\omega = (0 \ \omega_x \ \omega_y \ \omega_z)^T$  ein Quaternion, das die aktuell gemessenen Winkelgeschwindigkeitskomponenten enthält, dann gilt

$$\dot{q}(t) = \frac{\delta q(t)}{\delta t} = \frac{1}{2} \omega(t) q(t) \quad (4.16)$$

für die zeitliche Ableitung des Quaternions  $q$ . Aus abgeleitetem Quaternion  $\dot{q}(t)$  und aktuellem Quaternion  $q(t)$  wird das aktualisierte Quaternion  $q(t+dt)$  durch

$$q(t+dt) = q(t) + \dot{q}(t) \delta t \quad (4.17)$$

bestimmt, womit schließlich Folgendes gilt:

$$q(t+dt) = q(t) + \frac{1}{2} \delta t \omega(t) q(t) \quad (4.18)$$

Die Voraussetzung für den Einsatz dieser Methode ist, dass  $\delta q(t)$  klein ist, da ansonsten ohne die Verwendung eines geeigneten numerischen Integrationsverfahrens (wie beispielsweise des Runge-Kutta-Verfahrens [165, 112, 34]) zu große Ungenauigkeiten entstehen können. Bei typischen Winkelgeschwindigkeiten, die bei einem mobilen Roboter im Außenbereich auftreten, bedeutet diese Anforderung konkret, dass  $dt \lll 1$  sec gelten sollte. Die auf AMOR und PSYCHE eingesetzten Sensoren zur Messung von Winkelgeschwindigkeiten haben jeweils Messfrequenzen von mindestens 100 Hz und erreichen somit eine ausreichende Präzision mit der soeben beschriebenen Methode.

Einheitsquaternionen können dazu genutzt werden können, Rotationen zu beschreiben. Daher kann, analog zur Definition der Pose eines Roboters mit einem 6-Tupel, das drei Winkel zur Beschreibung der Orientierung nutzt (siehe Ausdruck 4.1), eine Definition auf Basis der Komponenten eines Einheitsquaternions  $q$  durchgeführt werden. Das 7-Tupel

$$\check{\rho} = (p_x, p_y, p_z, q_0, q_1, q_2, q_3) \quad (4.19)$$

definiert demnach eine Pose eines starren Körpers mit sechs Freiheitsgraden, die ein Einheitsquaternion zur Beschreibung der Orientierung nutzt. Dies ist die bevorzugte Art der Beschreibung von geometrischen Zuständen und Zusammenhängen in dieser Arbeit und wird in vielen der folgenden Beschreibungen aufgegriffen.

## 4.1 Odometrie und Koppelnavigation

Odometrie bezeichnet die Bestimmung der relativen Position eines Fahrzeugs durch Messung seiner Radumdrehungen [28]. Der zurückgelegte Weg des Fahrzeugs  $\Delta s$  wird durch

$$\Delta s = \pi d N \quad (4.20)$$

beschrieben, wobei  $d$  dem Raddurchmesser des Fahrzeugs und  $N$  der Anzahl der Radumdrehungen entspricht. Zur Steigerung der Genauigkeit können für  $N$  auch Radteilumdrehungen eingesetzt werden. Fehler in der Positionsbestimmung mit dieser Methode können aufgrund von mangelhafter Rad- oder Fahrwerksgeometrie entstehen. Daneben kann die Beschaffenheit des Untergrundes ebenfalls eine Fehlerquelle darstellen.

Formel 4.20 beschreibt jedoch nur die Berechnung des zurückgelegten Wegs und erlaubt keine Aussage über die Ausrichtung des Fahrzeugs. Sie ist somit für eine mehr als eindimensionale Positionsbestimmung nicht ausreichend. Soll die Position eines Fahrzeugs in der Ebene oder sogar im dreidimensionalen Raum bestimmt werden, ist es nötig, den Richtungsvektor des Fahrzeugs mit in die Berechnung einzubeziehen. Diese Form der Positionsbestimmung wird als Koppelnavigation<sup>2</sup> bezeichnet [16]. Der prinzipielle Ansatz dieser Methode besteht darin, den Pfad des Fahrzeug aus stückweise linearen Teilelementen zu rekonstruieren. Wenn die Messfrequenz sowohl des Odometers als auch des Sensors zur Bestimmung der Ausrichtung des Fahrzeugs hoch genug sind, wird so eine gute Näherung des realen Fahrzeugpfads erzeugt. Der aktuelle Orientierungsvektor  $v_o$  des Fahrzeugs wird durch Rotation des Null-Orientierungsvektors  $v_0$  bestimmt. Mit  $e_y = (0 \ 1 \ 0)^T$  im dreidimensionalen bzw.  $e_y = (0 \ 1)^T$  im zweidimensionalen Fall gilt

$$v_o = q_o e_y \bar{q}_o, \quad (4.21)$$

wobei  $q_o$  das Einheitsquaternion darstellt, das iterativ nach Gleichung 4.18 konstruiert wird. Die aktuelle Position  $p = (p_x \ p_y \ p_z)^T$  wird schritthaltend

---

<sup>2</sup>engl.: dead reckoning

bestimmt aus der Position des Fahrzeugs  $p(t)$  zum letzten Messzeitpunkt  $t - 1$  und dem aktuellen Orientierungsvektor  $v_o(t)$ , der mit dem zurückgelegten Weg  $\Delta s$  skaliert wurde.

$$p(t) = p(t - 1) + \Delta s v_o(t) \quad (4.22)$$

Wie in Kapitel 2.1.2 bereits beschrieben wurde, wird eine Kombination aus verschiedenen Sensoren zur Bestimmung der Orientierung des Roboters AMOR genutzt. An dieser Stelle wird kurz der Einfluss der unterschiedlichen Eigenschaften der einzelnen Sensoren bzw. ihrer Messprinzipien auf die Orientierung eines Fahrzeugs beleuchtet. Denn daraus ergibt sich die Nutzung verschiedener Varianten der Koppelnavigation in dieser Arbeit. Ein unbewegtes Fahrzeug kann mithilfe von Beschleunigungssensoren den eigenen Nick- und Rollwinkel relativ zum Gravitationsvektor der Erde bestimmen. Dabei vermisst der dreiaxige Beschleunigungssensor entlang der einzelnen orthogonalen Koordinatenachsen den Gravitationsvektor. Mit bekannter Erdbeschleunigung können aus dem resultierenden dreielementigen Beschleunigungsvektor zwei der drei Lagewinkel des Roboters bestimmt werden. Der Gierwinkel kann mit dieser Methode nicht festgestellt werden. Dazu kann ein Magnetfeldsensor genutzt werden, der das Erdmagnetfeld an der aktuellen Position des Roboters misst, um die Ausrichtung des Roboters zu bestimmen. Die Kombination dieser beiden Sensortypen erlaubt die Bestimmung aller drei Lagewinkel des Roboters, solange der Roboter sich nicht in Bewegung befindet oder das Erdmagnetfeld lokal nicht durch äußere Einflüsse gestört wird. Der Roboter darf deshalb nicht in Bewegung sein, weil sonst nicht eindeutig festgestellt werden kann, welchen Anteil die Bewegung des Roboters bzw. die Erdbeschleunigung an der auf dem Roboter gemessenen Gesamtbeschleunigung hat. Drehratensensoren können in diesem Zusammenhang eingesetzt werden, um diese Uneindeutigkeit näherungsweise aufzulösen. Durch Integration der gemessenen Winkelgeschwindigkeiten wird die relative Änderung der einzelnen Lagewinkel ermittelt. So kann die durch Beschleunigungs- und Magnetfeldsensoren ermittelte Orientierung des Roboters bei äußeren Störeinflüssen korrigiert werden. Im Umkehrschluss ist eine Integration von gemessenen Drehraten ohne die Einbeziehung von Mes-

sungen aus Beschleunigungs- und Magnetfeldsensoren nur unter zwei Bedingungen sinnvoll und praktikabel: Wenn erstens rein relative Winkeländerungen gemessen werden sollen; und zweitens nur über kürzere Zeiträume<sup>3</sup> gemessen werden soll, da sich sonst durch die zeitliche Integration fehlerbehafteter Winkelgeschwindigkeitsmessungen ein Fehler akkumuliert. Aufgrund dieser Überlegungen und der Anforderungen der in Kapitel 5 beschriebenen Methoden zur Modellierung der Umgebung unter Nutzung von Sensordaten aus verschiedenen Sensorquellen wurden in dieser Arbeit zwei unterschiedliche Varianten der Koppelnavigation implementiert, die jeweils unterschiedliche Orientierungsvektoren verwenden. Zum einen wird der driftende, ausschließlich auf integrierten Winkelgeschwindigkeitsmessungen basierende Orientierungsvektor  $\tilde{v}_o$  verwendet. Zum anderen wird der nichtdriftende, aus der Fusion von Sensoren für Beschleunigungen, Magnetfelder und Drehraten erzeugte Orientierungsvektor  $\hat{v}_o$  verwendet. Daraus resultieren nach Gleichung 4.22 die folgenden Ausdrücke zur Evolution der Position des Roboters per Koppelnavigation.

$$\hat{p}(t) = \hat{p}(t-1) + \Delta s \hat{v}_o(t) \quad (4.23)$$

$$\tilde{p}(t) = \tilde{p}(t-1) + \Delta s \tilde{v}_o(t) \quad (4.24)$$

Da sich die Motivation der Nutzung eines driftenden, aber wenig störungsanfälligen Orientierungsvektors  $\tilde{v}_o$  neben seinem nicht driftenden Pendant  $\hat{v}_o$  für eine der Varianten der implementierten Koppelnavigation eventuell intuitiv nicht direkt erschließt, folgt nun eine kurze Erklärung in diesem Zusammenhang. Das Ziel ist, möglichst präzise lokale, in sich konsistente Modelle der Umgebung zu erzeugen. Dafür muss die Bestimmung der aktuellen Pose des Roboters gleichzeitig robust gegen Störungen und auch bei Fahrzeugbewegungen stabil sein. Solange der Zeitraum der Posenbestimmung nicht zu lang ist, ist unter diesen Voraussetzungen die Bestimmung der relativen Pose des Roboters, basierend auf der reinen Integration von Drehraten, ideal geeignet für die Unterstützung der Fusion von Sensordaten zur Erzeugung eines lokalen Umgebungsmodells.

---

<sup>3</sup>abhängig von der Qualität der eingesetzten Drehratensensoren

## 4.2 Positionsbestimmung bezüglich erdfester Koordinatensysteme

Die im vorherigen Abschnitt beschriebenen Methoden der Positionsbestimmung mittels Odometrie bzw. Koppelnavigation können nur relative Positionsmessungen erzeugen. Der in der Außenbereichsrobotik verbreitetste Ansatz zur Erzeugung von absoluten Positionsmessungen bez. eines erdfesten Koordinatensystems besteht in der Nutzung des satellitengestützten Navigationssystems NAVSTAR-GPS<sup>4</sup> [80]. Das Funktionsprinzip des GPS-Systems bedingt, dass Objekte in der Sichtlinie zwischen Empfängerposition und Navigationssatelliten sowie von Objekten reflektierte Signale die Genauigkeit der Position beeinträchtigen. Sie können es sogar gänzlich unmöglich machen, eine Positionsbestimmung durchzuführen. Dies kann durch den Einsatz von Filtermethoden und die Integration von Daten aus der Koppelnavigation verbessert werden. Die Arbeiten von Mohamed und Schwarz und Wang et al. sind nur zwei der vielen Beispiele für diese Vorgehensweise [143][205] (siehe auch Kapitel 3.1.1 für weitere ausführliche Details zu diesem Themenkomplex).

Nichtsdestotrotz sind so auch nur zeitlich begrenzte Ausfälle oder Störungen der GPS-Signale zu überbrücken. Daher wurde im Rahmen dieser Arbeit ganz bewusst nicht auf die GPS-Technologie als Quelle für absolute Positionsinformationen für den Bodenroboter vertraut. Es wurden stattdessen zwei neuartige, alternative Methoden der absoluten Lokalisierung für AMOR entwickelt, die gar nicht oder nur in sehr geringem Maße (bspw. ausschließlich während einer Initialisierungsphase mit sehr geringen Genauigkeitsanforderungen) von einem GPS-Empfänger auf dem Bodenroboter abhängig sind. Zum einen eine kartenbasierte Lösung, welche die Position des Roboters auf Basis einer topologischen Straßenkarte mit metrischen Annotationen ermittelt. Zum anderen eine Lösung, die eine Kooperation zwischen Boden- und Luftroboter implementiert und auf der Annahme basiert, dass ein Flugroboter, der in einer gewissen

---

<sup>4</sup>Der Vollständigkeit halber sollte festgestellt werden, dass neben dem NAVSTAR-GPS-System weitere satellitengestützte Navigationssysteme in Betrieb genommen wurden oder im Begriff sind in Betrieb genommen zu werden: GLONASS (Russland), Galileo (Europa), Beidou/COMPASS (China). Bezogen auf die Verbreitung in Robotikanwendungen ist die GPS-Technologie im Vergleich mit den anderen Ansätzen derzeit deutlich überrepräsentiert.

Höhe operiert, nahezu immer eine Positionsbestimmung per GPS durchführen kann. Beide Methoden stellen einen Teil des neuartigen wissenschaftlichen Beitrags dieser Arbeit dar und werden in den folgenden zwei folgenden Kapiteln detailliert erörtert. Die zugehörigen Veröffentlichungen finden sich in [109] und [105].

### 4.3 Straßenkarten-basierte Lokalisierung

Der Umfang an Informationen, die in frei zugänglichen Geoinformationssystemen (GIS) integriert und zugreifbar gemacht werden, wächst stetig. Er hat mittlerweile ein Maß erreicht, das den Einsatz in der autonomen Außenbereichsrobotik sehr attraktiv macht. Neben georegistrierten Luftbildern stehen auch Daten über Straßen und Wege zur Verfügung. Ganze Straßennetzwerke sind in Form von topologischen Karten mit metrischer Annotation je Knoten abgelegt. Diese Datenquelle lässt sich ergänzend, aber auch alternativ zu satellitengestützten Methoden wie GPS für die Lösung des Lokalisierungsproblems in der mobilen Außenbereichsrobotik einsetzen.

Der Abgleich von Daten aus einer Straßenkarte mit schritthaltend ermittelten (relativen oder absoluten) Positionsinformationen eines Fahrzeugs wird im Allgemeinen als *Map Matching* bezeichnet. Dieser Grundansatz zur Kompensation von Fehlern in den schritthaltend durch Sensoren bestimmten Positionsmessungen kann dabei auf unterschiedliche Art umgesetzt und implementiert werden.

Die ersten Arbeiten im Bereich der Map Matching-Algorithmen sind hauptsächlich geometrisch motiviert gewesen. Dies bedeutet, dass die geometrische Form des Wegenetzes mit der Position, der Pose oder der Trajektorie eines Fahrzeugs verglichen wird. Der einfachste Ansatz in diesem Zusammenhang ist der Punkt-zu-Punkt-Abgleich [21]. Durch Suche nach dem der Fahrzeugposition nächsten Knoten im Wegenetz kann der Zusammenhang zwischen Fahrzeugposition und Wegenetz hergestellt werden. Die Zuordnung ist bei dieser Methode jedoch auf die Knotenpunkte der Karte reduziert. Um dieses Manko zu überkommen, wurde der Punkt-zu-Linie-Abgleich eingeführt [207]. Dabei wird die aktuelle Fahrzeugposition auf das nächstgelegene Liniensegment des

digitalen Wegenetzes projiziert. Die Folge ist eine Verbesserung des Lokalisierungsergebnisses gerade bei geringer Knotendichte in der digitalen Wegekarte. Wird die Historie der vergangenen Fahrzeugpositionen als Linie interpretiert, kann der Punkt-zu-Linie-Abgleich zum Linie-zu-Linie-Abgleich erweitert werden. Dies stellt eine weitere Verbesserung dar, wie in [153] nachgewiesen werden konnte. Nichtsdestotrotz sind rein geometrische Verfahren bei uneindeutigen lokalen Kartenkonfigurationen und fehlerhaften Sensormessungen unzuverlässig.

Eine Möglichkeit, rein geometrische Verfahren robuster zu gestalten, ist die Einbeziehung der Topologie des digitalen Wegenetzes. Ist die Kante des Wegenetzes bekannt, auf der sich das Fahrzeug aktuell bewegt, können durch Ermittlung der Kanten, die mit dieser Kante verbunden sind, alle anderen Kanten für den Abgleich zu diesem Zeitpunkt ausgeschlossen werden. Dies verhindert in rein geometrischen Verfahren häufig auftretende Fehler, wie etwa die falsche Zuordnung bei parallel verlaufenden Kanten, zwischen denen das Fahrzeug durch fehlerhafte Sensorinformationen lokalisiert wurde. Yu [214] stellt in diesem Zusammenhang ein Verfahren vor, das zunächst topologische Merkmale, wie Abbiegungen und Kreuzungen, aus Kartendaten und Fahrzeugtrajektorie extrahiert und diese danach gegeneinander abgleicht, um die Kante des Wegenetzes ausfindig zu machen, auf der sich das Fahrzeug aktuell bewegt. Quddus et al. [159] definieren andere Merkmale, die als gewichtete Summe zu einem einheitlichen Vergleichskriterium fusioniert werden, das den Abgleich von Fahrzeugtrajektorie und Wegenetz erlaubt. Die Merkmale integrieren dabei Informationen über die Orientierung von Fahrzeug und Wegenetzkante, den Abstand zwischen Fahrzeugposition und Wegenetzkante und schließlich die relative Lage der Fahrzeugposition zum nächsten Knoten des Wegenetzes. Zwar bedeutet die Nutzung der Topologie des Wegenetzes eine Verbesserung der Ergebnisse im Vergleich zu rein geometrischen Methoden. Jedoch werden in diesen Verfahren weiterhin die Wichtigkeit der Modellierung des Rauschens von Sensordaten und die Einbeziehung von deutlich mehr als nur einer kurzen Historie von Messungen außer Acht gelassen. Die Methoden der rekursiven Bayes-Schätzung bieten eine ideale Grundlage für die kontinuierliche Integration von verrauschten Sensormessungen mit dem Ziel, die aktuelle Fahrzeugposition zu schätzen. Ein weit verbreiteter und ausgereifter Ansatz in dieser

Kategorie von Algorithmen ist die Nutzung eines Kalman-Filters [92] zur Integration von Messungen aus GPS-Sensorik und Koppelnavigation und Daten eines digitalen Wegenetzes. Die Arbeiten von Kim et al. [96], Li et al. [126] und Obradovic et al. [150] sind nur einige Beispiele aus der Forschung in diesem Kontext.

Als Alternative zum Kalman-Filter hat sich der Partikelfilter [7] als weitere Methode der rekursiven Bayes-Schätzung etabliert. Im Unterschied zum Kalman-Filter ist es mit dem Partikelfilter u.a. möglich, auch multimodale Wahrscheinlichkeitsverteilungen zu modellieren. Bonnifait et al. [26] [61] präsentieren einen Multi-Hypothesen-Ansatz auf Basis eines Partikelfilters zur simultanen Schätzung der aktuell befahrenen Kante des Wegenetzes und der Position des Fahrzeugs auf dieser Kante. Das dem Partikelfilter zugrunde liegende Beobachtungsmodell integriert zu diesem Zweck Pseudo-Entfernungen und Doppler-Messungen eines GPS-Empfängers. Bonnifait et al. können in ihrer Arbeit nachweisen, dass die Initialisierung des Filters mit weniger als vier sichtbaren Satelliten und die Verfolgung des Fahrzeugs sogar mit nur zwei Satelliten möglich ist.

Toledo-Moreo et al. können in ihren Arbeiten [195] und [196] zeigen, dass sogar fahrspurgenaue Lokalisierung durch Nutzung eines Partikelfilters realisierbar ist. Es werden Daten eines GPS-Empfängers, eines Odometers und eines Gyroskops mit Daten aus einem digitalen Wegenetz fusioniert, wobei die Kanten des Wegenetzes im Gegensatz zu anderen Arbeiten nicht durch gerade Linien, sondern durch Klothoiden beschrieben werden.

Eine der ersten Arbeiten, die sich mit der Partikelfilter-basierten Lokalisierung ohne die Nutzung von Daten eines satellitengestützten Lokalisierungssystems beschäftigt, ist der sogenannte MAP<sup>5</sup>-Ansatz von Gustafsson et al. [71]. Dieser Ansatz entspricht der Grundidee der kartenbasierten Lokalisierungsmethode, die in dieser Arbeit entwickelt wurde. Es werden ausschließlich per Koppelnavigation erzeugte Positionierungsdaten und ein digitales Wegenetz in einem Partikelfilter zusammengeführt. Im Unterschied zu der in dieser Arbeit entwickelten Methode jedoch, wird die Gewichtung von Partikeln ausschließlich durch Ermittlung des euklidischen Abstands zur nächstgelegenen Kante des Wegenetzes durchgeführt und die zur Evaluation genutzte Teststrecke besteht

---

<sup>5</sup>Map Aided Positioning

hauptsächlich aus einer Sequenz von rechtwinkligen Abbiegevorgängen. Auch Davidson et al. [43] verwenden Daten der Koppelnavigation als einzige Sensorquelle in einem kartenbasierten Partikelfilter. Der vorgeschlagene Algorithmus implementiert zwei Operationsmodi: Es wird zwischen Geradeaus-Fahrt und Abbiegevorgang unterschieden, wobei der Moduswechsel durch Untersuchung der Gierwinkelgeschwindigkeit des Fahrzeugs getriggert wird. Die Idee dabei ist, während der Geradeaus-Fahrt die Partikel an die Straße zu binden, um ein Abdriften zu verhindern, und während des Abbiegevorgangs freie Partikelbewegung zu ermöglichen, um den Abbiegevorgang besser auf die abstrakte Darstellung der digitalen Straßenkarte abbilden zu können. Experimente mit dieser Vorgehensweise in dieser Arbeit haben gezeigt, dass die harte Trennung zwischen den vorgeschlagenen zwei Operationsmodi eine zusätzliche Fehlerquelle darstellt, welche die Robustheit der Lokalisierungsmethode negativ beeinflusst. Gerade bei einem autonom agierenden Fahrzeug ist zu erwarten, dass es häufiger als bei einem von einem menschlichen Benutzer gesteuerten Fahrzeug zu spontanen Gierwinkeländerungen kommt, da Algorithmen zur autonomen Navigation der menschlichen Antizipationsfähigkeit bei lokalen Ausweichmanövern unterlegen sind. Diese spontanen Gierwinkeländerungen führen zu fehlerhaft detektierten Moduswechseln. Daher sieht die in dieser Arbeit entwickelte Methode ausschließlich einen einzigen Operationsmodus vor. Sowohl die Grundlagen als auch die speziellen Implementierungsdetails dieser Methode werden nun in den folgenden Unterkapiteln erörtert.

### 4.3.1 Repräsentation von digitalen Wegenetzen

Da die in diesem Kapitel beschriebene Lokalisierungsmethode auf der Nutzung von digitalen Wegenetzdaten basiert, folgt an dieser Stelle zunächst eine kurze Einführung der Beschreibungs- bzw. Darstellungsform dieser Daten. Wegenetzdaten sind in aktuellen Geoinformationssystemen stets in Form eines klassischen Graphen abgelegt. Die Knoten entsprechen Orten im erdfesten Koordinatensystem und die Kanten definieren die Topologie der Knoten und damit das eigentliche Wegenetz. Ist also ein Knoten von einem anderen Knoten aus über einen Weg zu erreichen, sind beide mit einer Kante verbunden. Abgebildet auf die klassische Graphentheorie bedeutet dies, dass das Wegenetz

als Graph

$$\mathbb{M} = (V, E) \quad (4.25)$$

beschrieben werden kann, wobei

$$V = \{v^i = (v_x^i \ v_y^i) \in \mathbb{R}^2 \mid i = 1, 2, \dots, N_V\} \quad (4.26)$$

der Menge der Knoten des Graphen

$$E = \{e^j \in [V]^2 \mid j = 1, 2, \dots, N_E\} \quad (4.27)$$

der Menge der Kanten des Graphen entspricht und  $E \subseteq [V]^2$ . Die Elemente von  $E$  sind also 2-elementige Teilmengen von  $V$ .

In der Praxis der Lokalisierung eines mobilen Roboters ist es meist nötig, die Gesamtheit der Wegenetzdaten, die sich in aktuellen Geoinformationssystemen mittlerweile über die ganze Erde erstrecken, auf ein verarbeitbares Maß einzuschränken. Ein lokaler Ausschnitt der Wegenetzdaten sei daher durch den Graph

$$\mathbb{M}^{[1]} = (V^{[1]}, E^{[1]}) \quad \text{mit} \quad V^{[1]} \subseteq V, E^{[1]} \subseteq E \quad (4.28)$$

definiert. Die Elemente der Mengen  $V^{[1]}$  und  $E^{[1]}$  werden dabei jeweils auf ihre Zugehörigkeit zum lokalen Ausschnitt der Wegenetzdaten geprüft, indem festgestellt wird, ob sie innerhalb eines begrenzenden Rechtecks liegen.

Eine weitere relevante Größe ist die Länge

$$l(e^j) : E \rightarrow \mathbb{R} \quad (4.29)$$

der Kante  $e^j$ , die durch Bestimmung des euklidischen Abstands der Knoten, die durch die Kante verbunden werden, ermittelt wird.

### 4.3.2 Rekursive Bayes-Schätzung

Die rekursive Bayes-Schätzung ist ein probabilistischer Ansatz zur Zustandsschätzung eines Systems unter Berücksichtigung bzw. Modellierung der dem System zugrunde liegenden Unsicherheit. An dieser Stelle werden die grundlegende Begriffe dieser Theorie eingeführt. Die verwendete Notation orientiert sich dabei an der Arbeit von Thrun et al. [192].

Den folgenden Ausführungen liegt die Annahme zugrunde, dass der Prozess der rekursiven Bayes-Schätzung einen Markov-Prozess erster Ordnung darstellt. Ein Markov-Prozess zeichnet sich durch die Tatsache aus, dass es ausreichend ist, einen Teil der vorherigen Systemzustände in die Schätzung des zukünftigen Systemzustands einzubeziehen. Das Ergebnis ist dabei gleichwertig mit dem Ergebnis, das entstanden wäre, hätte man alle vorherigen Systemzustände einbezogen. Im Fall eines Markov-Prozesses erster Ordnung bedeutet dies sogar, dass die Schätzung nur vom aktuellen Zustand des Systems abhängig ist.

Für die folgenden Erklärungen wird die Methode der rekursiven Bayes-Schätzung auf das Problem der Lokalisierung eines mobilen Roboters angewendet. Der Zustand des Roboters zum Zeitpunkt  $t$  wird durch den Zustandsvektor  $x_t$  definiert, der dem Zustandsraum  $\chi$  entstammt. Messungen bspw. von Sensoren des Roboters zum Zeitpunkt  $t$  werden mit  $z_t$  und Aktionen im Sinne von Steuerdaten werden mit  $u_t$  bezeichnet.

Der Ablauf der rekursiven Bayes-Schätzung gliedert sich in zwei Schritte, den Prädiktions- und den Filterschritt, die in Abb. 4.2 schematisch dargestellt sind. Die Propagation des Systemzustands wird modelliert durch das Bewegungsmodell auf der einen Seite und das Beobachtungsmodell auf der anderen Seite. Das Bewegungsmodell beschreibt, wie sich der bisherige Systemzustand  $x_{t-1}$  unter Berücksichtigung der Steuerdaten  $u_t$  und des Systemrauschens in den aktuellen Systemzustand  $x_t$  transformieren lässt. Die zugehörige Wahrscheinlichkeitsdichte  $p(x_t|x_{t-1}, u_t)$  wird als Zustandsübergangswahrscheinlichkeit bezeichnet. Das Beobachtungsmodell modelliert die Erzeugung einer Messung  $z_t$  abhängig vom aktuellen Zustand  $x_t$  und dem Messrauschen mit der zugehörigen Wahrscheinlichkeitsdichte  $p(z_t|x_t)$ , die als Beobachtungswahrscheinlichkeit definiert ist. Es ist wichtig festzustellen, dass der genaue Systemzustand zu keinem Zeitpunkt des Schätzprozesses direkt beobachtet werden kann. Es ist

lediglich möglich, diesen anhand der Mess- und Steuerdaten abzuleiten. Während des zweischrittigen, rekursiven Schätzprozesses wird zwischen der Apriori- und der Aposteriori-Wahrscheinlichkeitsdichte unterschieden. Die Apriori-Wahrscheinlichkeitsdichte  $p(x_t|z_{1:t-1}, u_{1:t})$  beschreibt die Wahrscheinlichkeitsdichte vor dem Eintreffen der nächsten Messung, wobei schon die aktuellen Steuerdaten berücksichtigt wurden. Die Aposteriori-Wahrscheinlichkeitsdichte  $p(x_t|z_{1:t}, u_{1:t})$  bezieht unter Berücksichtigung der Apriori-Wahrscheinlichkeitsdichte die aktuelle Messung ein, um den aktuellen Systemzustand zu schätzen. Dabei gilt  $z_{1:k} = \{z_i\}_{i=1}^k$  und  $u_{1:k} = \{u_i\}_{i=1}^k$ .

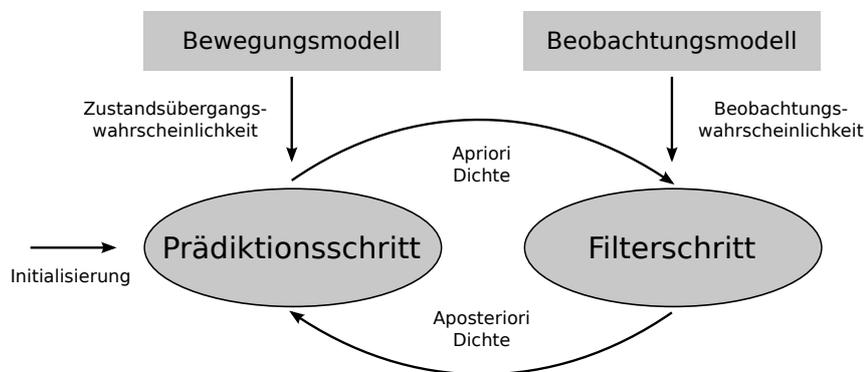


Abbildung 4.2: Grundlegender Ablauf der rekursiven Bayes-Schätzung.

## Prädiktionsschritt

Der Prädiktionsschritt ist demnach als Zwischenschritt zu sehen, der aus der Aposteriori-Wahrscheinlichkeitsdichte  $p(x_{t-1}|z_{1:t-1}, u_{1:t-1})$  des vorherigen Zeitschritts die Apriori-Wahrscheinlichkeitsdichte  $p(x_t|z_{1:t-1}, u_{1:t})$  des aktuellen Zeitschritts ableitet. Da für Markov-Prozesse die Markov-Eigenschaft gilt, kann durch Integration des zugehörigen Terms die Chapman-Kolmogorow-Gleichung auf das Problem der rekursiven Bayes-Schätzung angewendet werden [98], um die Apriori-Wahrscheinlichkeitsdichte  $p(x_t|z_{1:t-1}, u_{1:t})$  zu bestimmen durch

$$p(x_t|z_{1:t-1}, u_{1:t}) = \int p(x_t|x_{t-1}, u_t) p(x_{t-1}|z_{1:t-1}, u_{1:t-1}) dx_{t-1}. \quad (4.30)$$

## Filterschritt

Aus der im vorherigen Schritt ermittelten Apriori-Wahrscheinlichkeitsdichte  $p(x_t|z_{1:t-1}, u_{1:t})$  wird mit Eintreffen einer neuen Messung  $z_t$  die neue Aposteriori-Wahrscheinlichkeitsdichte  $p(x_t|z_{1:t}, u_{1:t})$  durch

$$p(x_t|z_{1:t}, u_{1:t}) = \eta p(z_t|x_t) p(x_t|z_{1:t-1}, u_{1:t}) \quad (4.31)$$

ermittelt, wobei  $p(z_t|x_t)$  die Beobachtungswahrscheinlichkeit der Messung  $z_t$  abhängig von Zustand  $x_t$  und  $\eta$  einen Faktor zur Normalisierung darstellt.

## Partikelfilter

In der Praxis ist die geschlossene Berechnung der zuvor beschriebenen Wahrscheinlichkeitsdichten nicht praktikabel. Daher wurden Methoden zur näherungsweisen Modellierung des allgemeinen rekursiven Bayes-Filters entwickelt, die in der Praxis effizient implementiert werden können.

Der Kalman-Filter nach Kalman [92] ist ein Beispiel in diesem Kontext. Es werden die drei folgenden Grundannahmen getroffen:

- Bewegungs- und Beobachtungsmodell sind durch lineare Funktionen modelliert.
- Die Aposteriori-Wahrscheinlichkeitsdichte  $p(x_t|z_{1:t}, u_{1:t})$  wird als Normalverteilung modelliert.
- Die Rauschterme von Bewegungs- und Beobachtungsmodell sind durch eine mittelwertfreie Normalverteilung modelliert.

Dies sorgt für eine starke Vereinfachung der Berechnung der gesuchten Wahrscheinlichkeitsdichten. Sind jedoch die Grundannahmen des Kalman-Filters nur unzureichende Näherungen des realen Systemverhaltens, ist mit einer schlechten Leistungsfähigkeit des Schätzers zu rechnen. Gerade falls multimodale Wahrscheinlichkeitsdichten zu erwarten sind, reicht das unimodale Modell der Normalverteilung des Kalman-Filters zur adäquaten Beschreibung der Aposteriori-Wahrscheinlichkeitsdichte  $p(x_t|z_{1:t}, u_{1:t})$  nicht aus

Einen anderen Ansatz wählt der in dieser Arbeit genutzte Partikelfilter [7], der ebenfalls als Spezialisierung des allgemeinen Bayes-Filters zu sehen ist. Allerdings wird ein probabilistischer Ansatz zur Reduktion der Komplexität des allgemeinen Bayes-Schätzers gewählt. Es wird weiterhin keine parametrische Darstellung der Wahrscheinlichkeitsdichten eingesetzt. Die Wahrscheinlichkeitsdichten werden vielmehr durch eine endliche Menge von zufällig aus der zu beschreibenden Wahrscheinlichkeitsverteilung gezogenen Abtastpunkten modelliert. Die Abtastpunkte werden als Partikel bezeichnet und entsprechen lokalen Stützstellen der durch die Partikel approximierten Wahrscheinlichkeitsdichte. Ein Partikel zum Zeitpunkt  $t$  wird durch das Tupel

$$s_t^i = (x_t^i, w_t^i) \quad (4.32)$$

beschrieben. Die Menge aller  $N$  Partikel des Filters wird mit  $S = \{s^i | i = 1, 2, \dots, N\}$  bezeichnet. Neben dem durch das Partikel repräsentierten Zustand  $x_t^i$  beinhaltet jedes Partikel-Tupel den Gewichtungsfaktor  $w_t^i$ . Der Gewichtungsfaktor  $w_t^i$  entspricht der Beobachtungswahrscheinlichkeit  $p(z_t | x_t^i)$  der aktuellen Messung  $z_t$ . Die Gesamtheit der gewichteten Partikel in  $S$  stellt damit eine Approximation der Aposteriori-Wahrscheinlichkeitsdichte  $p(x_t | z_{1:t}, u_{1:t})$  dar.

Der konkrete Ablauf der Filterung mit Partikelfilter sieht nun im Prädiktions-schritt die Bewegung jedes Partikels anhand des Bewegungsmodells des Filters mit den aktuellen Steuerdaten  $u_t$  vor. Ziel ist es, eine Approximation der Apriori-Wahrscheinlichkeitsdichte  $p(x_t | z_{1:t-1}, u_{1:t})$  zu erzeugen. Im folgenden Filterschritt werden die Gewichtungsfaktoren  $w_t^i$  anhand der aktuellen Messung  $z_t$  aktualisiert, um die approximierte Aposteriori-Wahrscheinlichkeitsdichte  $p(x_t | z_{1:t}, u_{1:t})$  zu erhalten.

Auch bei einer hochqualitativen Initialisierung der Partikelmenge zu Beginn des gesamten Filterprozesses ist durch wiederholte Ausführung des Prädiktions-schritts mit einem Abdriften vieler Partikel in Bereiche des Zustandsraums zu rechnen, die nicht entscheidend zur Beschreibung der gesuchten Wahrscheinlichkeitsdichte beitragen, da an diesen Stellen des Zustandsraums die lokale Wahrscheinlichkeit nahezu Null ist. Diese Unterrepräsentation von Partikeln in den relevanten Bereichen der Wahrscheinlichkeitsdichte wird als Degenera-

tionsproblem bezeichnet [51]. Um diesem Problem zu begegnen, kann ein sogenannter Resampling-Schritt durchgeführt werden. Resampling kann auf verschiedene Arten realisiert werden. Ein Ansatz ist das sogenannte multinomiale Resampling nach Gordon et al. [67], das durch Duplizieren von Partikeln aus der aktuellen Partikelmenge eine neue Partikelmenge erzeugt, die anstatt der alten Partikelmenge verwendet wird. Welche Partikel dupliziert werden, wird zufällig, jedoch gemäß der den Partikeln zugeordneten Gewichtungsfaktoren, ermittelt. Partikel mit hohen Gewichtungsfaktoren sind demnach mit größerer Wahrscheinlichkeit häufiger in der neuen Partikelmenge vertreten. Nachteilig an dieser Vorgehensweise ist, dass eine gleichgewichtete Partikelmenge sich unnötigerweise durch die Anwendung des multinomialen Resamplings verändern kann. Das in [7] und [192] beschriebene systematische Resampling stellt in diesem Zusammenhang eine deterministische Variante zum multinomialen Resampling dar. Die Grundlage für die Erklärung dieses Resampling-Schritts ist die kumulative Dichtefunktion der Wahrscheinlichkeiten, die den Partikeln zugeordnet sind. Diese Dichtefunktion erlaubt eine Abbildung von kumulierter Wahrscheinlichkeit auf den Index eines Partikels. Systematisches Resampling tastet die kumulative Dichtefunktion der Partikelgewichte gleichmäßig im Abstand  $\frac{1}{N}$  ab und ermittelt an den Abtastpunkten den zum kumulierten Wahrscheinlichkeitswert zugeordneten Partikelindex. Das durch diesen Index definierte Partikel wird analog zur Vorgehensweise beim multinomialen Resampling dupliziert, um so durch Duplikation aller Partikel an den entsprechenden Abtastpunkten eine neue Partikelmenge zu erzeugen. Ein weiteres Resampling-Verfahren ist das sogenannte *stratified resampling* [97]. Gerade im Zusammenhang mit multimodalen Verteilungen zeigt dieses Verfahren ein deutlich verbessertes Verhalten des Filters durch Aufteilung des Ereignisraums in Unterintervalle deren Wahrscheinlichkeiten bekannt sind bzw. geschätzt werden. In den Unterintervallen wiederum können dann jeweils die oben eingeführten Verfahren zum eigentlichen Resampling genutzt werden. Ausführliche Vergleiche der soeben beschriebenen und weiterer Resampling-Verfahren finden sich in [50] und [81].

Zu welchem Zeitpunkt ein Resampling-Schritt durchgeführt wird, ist nicht grundsätzlich festgelegt. Die Anzahl der effektiven Abtastpunkte

$$N_{eff} = \frac{1}{\sum_{i=1}^N (w_t^i)^2}, \quad 1 \leq N_{eff} \leq N \quad (4.33)$$

dient als robuster Indikator bezüglich der Frage, wann ein Resampling-Schritt nötig ist. Fällt die Anzahl der effektiven Abtastpunkte unter einen zuvor definierten Schwellwert, ist die Ausführung eines Resampling-Schritts angezeigt.

### 4.3.3 Grundprinzip der Partikelfilter-basierten Lokalisierung mit digitalen Kartendaten

Zunächst soll an dieser Stelle eine intuitiv verständliche Einführung in das Grundprinzip der Partikelfilter-basierten Lokalisierung mit digitalen Kartendaten anhand eines vereinfachten Beispiels gegeben werden.

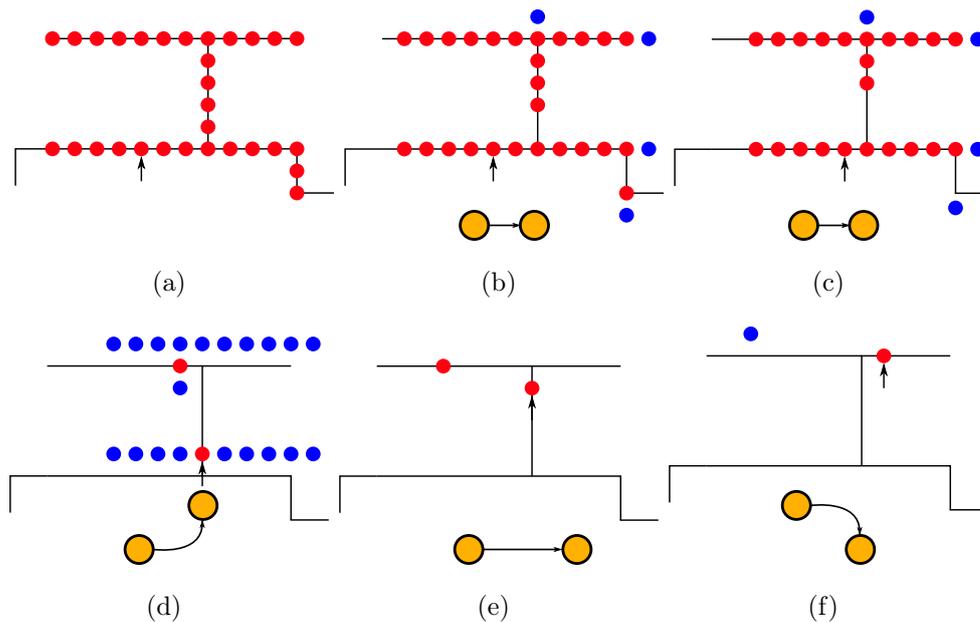


Abbildung 4.3: **Iterationsschritte des kartenbasierten Partikelfilter-Algorithmus zur Lokalisierung eines mobilen Roboters.** Schwarzer Pfeil: Tatsächliche Position des Roboters. Rot: Große Gewichtungen. Blau: Kleine Gewichtungen. Gelb: Bewegung des Roboters.

Abbildung 4.3 zeigt fünf aufeinander folgende Iterationsschritte des kartenbasierten Partikelfilter-Algorithmus. Der Algorithmus beginnt mit der äqui-

distanten Initialisierung der Partikel entlang der Kanten des Wegenetzes (a). Nach zwei Geradeaus-Bewegungen erhalten einige Partikel geringe Gewichungen, da sie sich von den Kanten des Wegenetzes entfernt haben ((b) und (c)). Da sich aber der Großteil der Partikel noch auf Kanten des Wegenetzes befindet, sind diese weiterhin hoch bewertet und daher ist keine eindeutige Lokalisierung des Roboters möglich. Dieser Zustand ändert sich fundamental nach dem ersten Abbiegevorgang des Roboters (d). Viele der zuvor noch hoch bewerteten Partikel befinden sich nun nicht mehr auf Kanten des Wegenetzes und es bleibt lediglich eine kleine Anzahl an Partikeln übrig, die noch hoch bewertet bleiben. Die anschließende längere Geradeaus-Bewegung ändert an diesem Zustand nichts Wesentliches (e). Der nächste Abbiegevorgang jedoch eliminiert auch die letzten falschen Partikel und eine eindeutige Lokalisierung des Fahrzeugs ist nun möglich (f). Der Filter ist somit konvergiert. Es ist wichtig festzustellen, dass ausschließlich relative Positionsmessungen genutzt wurden, um den Roboter erfolgreich im Wegenetz zu lokalisieren. Da die Motivation für die Entwicklung einer alternativen Methode zur Lokalisierung eines mobilen Roboters im erdfesten Koordinatensystem die Befreiung der Abhängigkeit von satellitenbasierten Lokalisierungsmethoden ist, ist diese Erkenntnis von zentraler Bedeutung.

#### 4.3.4 Entwurf des Partikelfilters

In Kapitel 4.3.2 wurden bereits die grundlegenden Mechanismen der rekursiven Bayes-Schätzung im Kontext der Lokalisierung eines mobilen Roboters eingeführt. Der nichttriviale Unterschied zu Algorithmen, die dieses Konzept konkret implementieren, besteht jedoch in der spezifischen Ausgestaltung der Punkte: Definition des Zustandsraums, Initialisierung der Partikelmenge, Entwurf von Bewegungs- und Beobachtungsmodell und Wahl der Resampling-Strategie. In den folgenden Unterkapiteln wird beschrieben, wie diese Aspekte für die in dieser Arbeit entwickelte neue Lokalisierungsmethode konkret umgesetzt worden sind.

## Zustandsraum

Ein Zustand  $x$  ist im Folgenden als dreidimensionaler Posenvektor  $(p_x \ p_y \ \phi)^T$  definiert, der die Pose des Roboters im erdfesten Koordinatensystem festlegt. Somit gilt:

$$x = (p_x \ p_y \ \phi)^T. \quad (4.34)$$

## Initialisierung

Die Wahl einer effektiven Methode zur Erzeugung der initialen Partikelmengens ist eine wichtige Entwurfsentscheidung, die die spätere Leistungsfähigkeit des Partikelfilters stark beeinflusst. Die klassische Methode für die Initialisierung der Partikelmengens eines Partikelfilters ist die Annahme einer initialen parametrischen Wahrscheinlichkeitsverteilung, gemäß der die initiale Partikelmengens zufällig durch Ziehung ermittelt wird. Falls eine ungefähre Startposition des Roboters bekannt ist, kann dies beispielsweise eine Normalverteilung sein, deren Mittelwert mit der approximierten Startposition des Roboters zusammenfällt. Ist die Startposition vollständig unbekannt, muss sogar von einer Gleichverteilung ausgegangen werden. Diese Methoden ignorieren jedoch vollständig die Informationen, die das digitale Wegenetz zur Verfügung stellt.

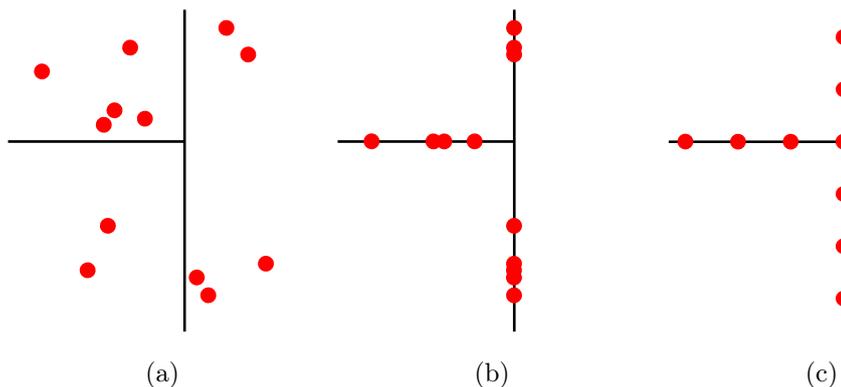


Abbildung 4.4: **Varianten bei der Initialisierung der Partikelmengens zu Beginn des Filterprozesses.** Zufällige Verteilung im gesamten Zustandsraum (a). Projektion auf die jeweils nächstgelegene Kante (b). Äquidistante Kantenabtastrung (c). [136]

Dies führt zur initialen Positionierung von Partikeln an Stellen im Zustandsraum, die mit hoher Wahrscheinlichkeit nicht die korrekte Fahrzeugpose repräsentieren, da sie sich nicht in der Nähe von Kanten des Wegenetzes befinden (siehe Abb. 4.4 (a)). Die orthogonale Projektion der Partikel auf die jeweils nächstgelegene Kante des Wegenetzes entlang der Normale der Kante nach der Initialisierung mit der zuvor beschriebenen Methode stellt bereits eine deutliche Verbesserung dar (siehe Abb. 4.4 (b)). Allerdings bietet auch diese Methode noch Optimierungspotenzial, da es durch die Projektion auf nächstgelegene Kanten zu lokalen Partikelhäufungen kommen kann, die das Verhalten des Filters negativ beeinflussen können. In dieser Arbeit wird daher die sogenannte äquidistante Kantenabtastung als Initialisierungsmethode eingeführt. Das Ziel dieser Methode ist es, alle initialen Partikel möglichst gleichmäßig über das gesamte genutzte Wegenetz zu verteilen. Der Zusammenhang zwischen der Anzahl der Partikel  $N$ , dem Abstand zwischen im Wegenetz adjazenten Partikeln  $\Delta s$  und der akkumulierten Länge

$$L_E = \sum_{i=1}^{|E^{[1]}|} l(e^i) \quad (4.35)$$

aller Kanten im lokalen Wegenettausschnitt  $M^{[1]}$  wird durch

$$N = \frac{L_E}{\Delta s} \quad (4.36)$$

beschrieben. Wird die Partikelanzahl  $N$  anhand der zur Verfügung stehenden Rechenressourcen bestimmt, kann diese als konstant vorausgesetzt werden. Durch Variation des Verhältnisses  $\frac{L_E}{\Delta s}$  kann die Initialisierungsphase an die Anforderungen der Lokalisierungsanwendung angepasst werden. Soll ein möglichst großer initialer Wegenettausschnitt abgedeckt werden, müssen im Gegenzug eine längere Konvergenzphase und sogar ein eventuelles Scheitern akzeptiert werden. Experimente im Rahmen dieser Arbeit haben gezeigt, dass für die Lokalisierung eines mobilen Roboters, der sich mit ca. 1 bis  $1,5 \frac{m}{s}$  bewegt, die Wahl des Abstands zwischen adjazenten Partikeln  $\Delta s$  aus dem Intervall  $[1m, 2m]$  gute Ergebnisse erzielt. Ab einem Abstand  $\Delta s > 5m$  erhöhte sich die Häufigkeit der Fehllokalisierungen nach erfolgreicher Konvergenz des

Filters deutlich.

Abschließend werden die initialen Zustandsvektoren aller Partikel mit mittelwertfreien, normalverteilten Rauschtermen versehen, um etwaige Fehler in den Daten des Wegenetzes zu modellieren.

## Bewegungsmodell

Das Bewegungsmodell beschreibt die Propagation des Systemzustands im Prädiktionsschritt des Filters abhängig von den Steuerdaten  $u_t$ . Zur Umsetzung des kartenbasierten Partikelfilter-Algorithmus wird das sogenannte Constant Turn Rate and Velocity (CTRV)-Bewegungsmodell genutzt [172]. Das CTRV-Modell bezieht im Gegensatz zu einem rein linearen Bewegungsmodell zusätzlich zur Geschwindigkeit  $v$  die Drehrate des Fahrzeugs um die z-Achse  $\omega$  in die Propagation des Fahrzeugzustands ein. Somit werden die Steuerdaten  $u_t$  durch den zweidimensionalen Vektor  $\begin{pmatrix} v_t & \omega_t \end{pmatrix}^T$  beschrieben und der propagierte Fahrzeugzustand  $x_{t+1}$  zum Zeitpunkt  $t + 1$  ist definiert durch

$$x_{t+1} = \begin{pmatrix} \frac{v_t}{\omega_t} \sin(\omega \Delta t + \phi_t) - \frac{v_t}{\omega_t} \sin(\phi_t) + p_{x_t} \\ -\frac{v_t}{\omega_t} \cos(\omega \Delta t + \phi_t) + \frac{v_t}{\omega_t} \cos(\phi_t) + p_{y_t} \\ \omega \Delta t + \phi_t \end{pmatrix}, \quad (4.37)$$

wobei  $\Delta t$  die Zeitdifferenz zwischen den Zeitpunkten  $t$  und  $t + 1$  repräsentiert. Die Steuerdaten werden dabei stets und ausschließlich aus den Messungen der Koppelnavigation des Bodenroboters abgeleitet. Da das zuvor beschriebene Bewegungsmodell von störungsfreien Steuerdaten ausgeht und somit die Modellierung eventueller Fehler in den Daten der Koppelnavigation versäumt, werden vor der Propagation des Systemzustands mit dem Bewegungsmodell die Steuerdaten mit mittelwertfreien, normalverteilten Rauschtermen versehen.

## Beobachtungsmodell

Der intuitivste und einfachste Weg, das Beobachtungsmodell eines Partikelfilters im Rahmen der kartenbasierten Lokalisierung eines mobilen Roboters zu implementieren, ist, jedes Partikel anhand des Abstands zur jeweils nächsten

Kante des Wegenetzes zu gewichten [71]. Je näher ein Partikel der entsprechenden Kante ist, desto besser wird es bewertet. Diese Vorgehensweise hat sich in ersten Experimenten jedoch nicht als ausreichend erwiesen. Komplexe lokale Wegenetzkonstellationen in Kombination mit unregelmäßigen Bewegungen des Roboters, wie sie während der autonomen Operation eines mobilen Roboters häufig auftreten können, verlangen nach einer komplexeren Methode der Partikelgewichtung. Daher wird hier eine optimierte Variante der Bewertung von Partikeln eingeführt, die nicht nur den aktuellen Zustand des betrachteten Partikels einbezieht, sondern zusätzlich eine endliche Menge von vergangenen lokalen Fahrzeugbewegungen.

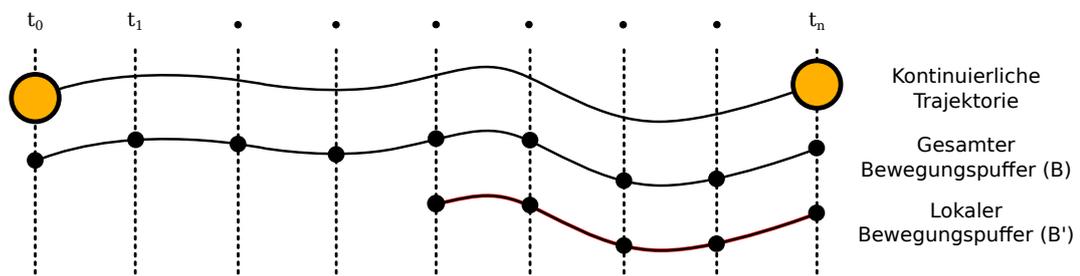


Abbildung 4.5: **Erzeugung des lokalen Bewegungspuffers durch Abtastung der kontinuierlichen Fahrzeugtrajektorie.**

Aus der kontinuierlichen, von der Koppelnavigation erzeugten Trajektorie des Fahrzeugs wird durch äquidistante Abtastung die geordnete Menge von Fahrzeugposen

$$B = \left\{ \rho_i = \begin{pmatrix} p_i & \phi_i \end{pmatrix}^T = \begin{pmatrix} p_{x_i} & p_{y_i} & \phi_i \end{pmatrix}^T \in \mathbb{R}^3 \mid i = 1, 2, \dots, n \right\} \quad (4.38)$$

erzeugt, die den sogenannten Bewegungspuffer des Roboters darstellt. Das Bezugskoordinatensystem des Bewegungspuffers ist das initiale Szenenkoordinatensystem  $f_k$  (siehe Kapitel 5.2.3), das durch die Pose des Roboters im erdfesten Koordinatensystem zu Beginn der autonomen Operation des Roboters definiert ist. Das  $n$ -te Element  $\rho_n$  der geordneten Menge stellt die aktuellste Messung dar. Aus dem Bewegungspuffer  $B$  werden die  $k$  aktuellsten Elemente selektiert und so transformiert, dass sie bezüglich des durch die aktuellste Pose

$\rho_n$  definierten Koordinatensystems dargestellt werden. So entsteht der lokale Bewegungspuffer

$$B' = \left\{ p'_i = R_{-\phi_n}(p_i - p_n) \in \mathbb{R}^2 \mid i = n - k, \dots, n ; \begin{pmatrix} p_i \\ \phi_i \end{pmatrix}, \begin{pmatrix} p_n \\ \phi_n \end{pmatrix} \in B \right\}, \quad (4.39)$$

wobei  $R_{-\phi_n}$  eine zweidimensionale Rotationsmatrix mit positiver mathematischer Drehrichtung darstellt.  $B'$  enthält nur noch zweidimensionale Elemente, da die Orientierung der Elemente des lokalen Bewegungspuffers für die folgenden Schritte nicht benötigt wird. Der lokale Bewegungspuffer  $B'$  wird genutzt, um einen Linie-zu-Linie-Abgleich mit dem Wegenetz zu realisieren. Dies geschieht für jedes Partikel separat. Daher wird  $B'$  abhängig vom aktuell betrachteten Partikel jeweils so transformiert, dass das Bezugskordinatensystem durch den Zustand des aktuell betrachteten Partikels definiert ist:

$$B'(x_i) = \left\{ p'_j(x_i) = (R_{\phi_i} p'_j) + p_i \in \mathbb{R}^2 \mid j = n - k, \dots, n ; p'_j \in B' \right\} \quad (4.40)$$

Auf Basis des partikelzustandsabhängigen Bewegungspuffers  $B'(x_i)$  wird die Gewichtung jedes Partikels implementiert. Wie schon zuvor angesprochen, geschieht dies durch einen Linie-zu-Linie-Abgleich mit dem Wegenetz. Abbildung 4.6 zeigt das Grundprinzip dieser Vorgehensweise an einem Beispiel.

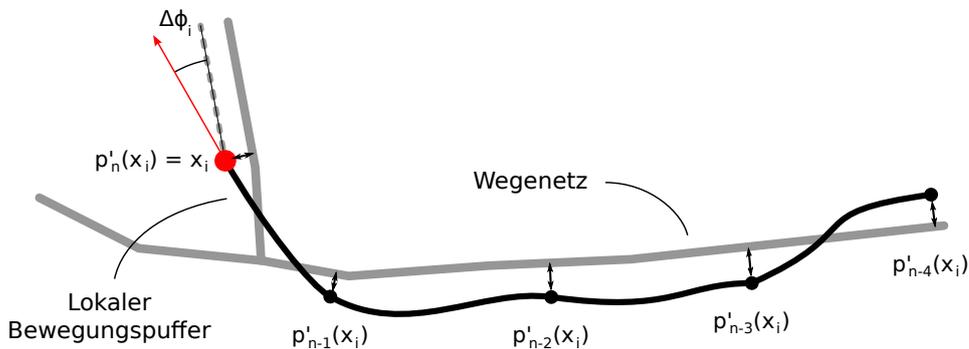


Abbildung 4.6: **Linie-zu-Linie-Abgleich zwischen lokalem Bewegungspuffer und Wegenetz als Sequenz von Punkt-zu-Linie-Abgleichen.**

Der Linie-zu-Linie-Abgleich wird als Sequenz von Punkt-zu-Linie-Abgleichen umgesetzt, deren akkumulierte Ergebnisse den Linie-zu-Linie-Abgleich implementieren. Die Punkte für den Abgleich entsprechen den Punkten  $p'_j(x_i)$  des lokalen Bewegungspuffers des zu bewertenden Partikelzustandsvektors  $x_i$ . Die damit zu vergleichende Linie wird durch die in der Nähe liegenden Kanten des Wegenetzes repräsentiert. Die einzelnen Punkt-zu-Linie-Abgleiche ermitteln die kürzeste Distanz zur nächsten Kante des Wegenetzes nach dem in Abb. 4.7 dargestellten Prinzip. Zunächst wird eine Orthogonalprojektion auf die nächste Kante entlang ihrer Normale versucht. Ist diese erfolgreich, entspricht der gesuchte Abstand dem Abstand zwischen projiziertem und betrachtetem Punkt des Bewegungspuffers. Ist diese nicht erfolgreich, wird stattdessen die Distanz zum nächsten Knoten des Wegenetzes als Ergebnis gewählt. Der so bestimmte kleinste Abstand des Punktes  $p'_j(x_i)$  zur nächster Wegenetzkante wird mit  $d_{min}(p'_j(x_i))$  bezeichnet.

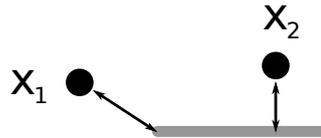


Abbildung 4.7: Schematische Darstellung der zwei möglichen Fälle beim Punkt-zu-Linie-Abgleich.

Die Gewichtung  $w_i$  eines Partikels  $s_i$  wird mit diesen Vorüberlegungen schließlich bestimmt durch:

$$w_i = \mathcal{N}_{\Delta\phi}(\Delta\phi_i) \sum_{j=1}^{|B'(x_i)|} \kappa(j) \mathcal{N}_{d_{min}}(d_{min}(p'_j(x_i))). \quad (4.41)$$

Neben den minimalen Kantenabständen  $d_{min}(p'_j(x_i))$  geht der Winkel  $\Delta\phi$  zwischen der Orientierungsachse des Partikels und der Wegenetzkante in die Gewichtung des betrachteten Partikels ein, wobei  $\mathcal{N}_{\Delta\phi}$  und  $\mathcal{N}_{d_{min}}$  mittelwertfreie Normalverteilungen darstellen. Zusätzlich wird die abklingende Gewichtungsfunktion  $\kappa$  eingeführt, die jedes Element des lokalen Bewegungspuffers  $B'(x_i)$  individuell gewichtet, um den Einfluss der einzelnen Elemente des Bewegungspuffers zu kontrollieren. Mit einer abklingenden Funktion  $\kappa$  werden somit die älteren Elemente des lokalen Bewegungspuffers geringer gewichtet, weil davon

ausgegangen werden kann, dass die neuesten Elemente die größte Relevanz für die Gewichtungsfunktion haben. Zusammengenommen implementiert die Gewichtungsfunktion für  $w_i$  eine glatt abklingende, reziproke Bewertungsfunktion.

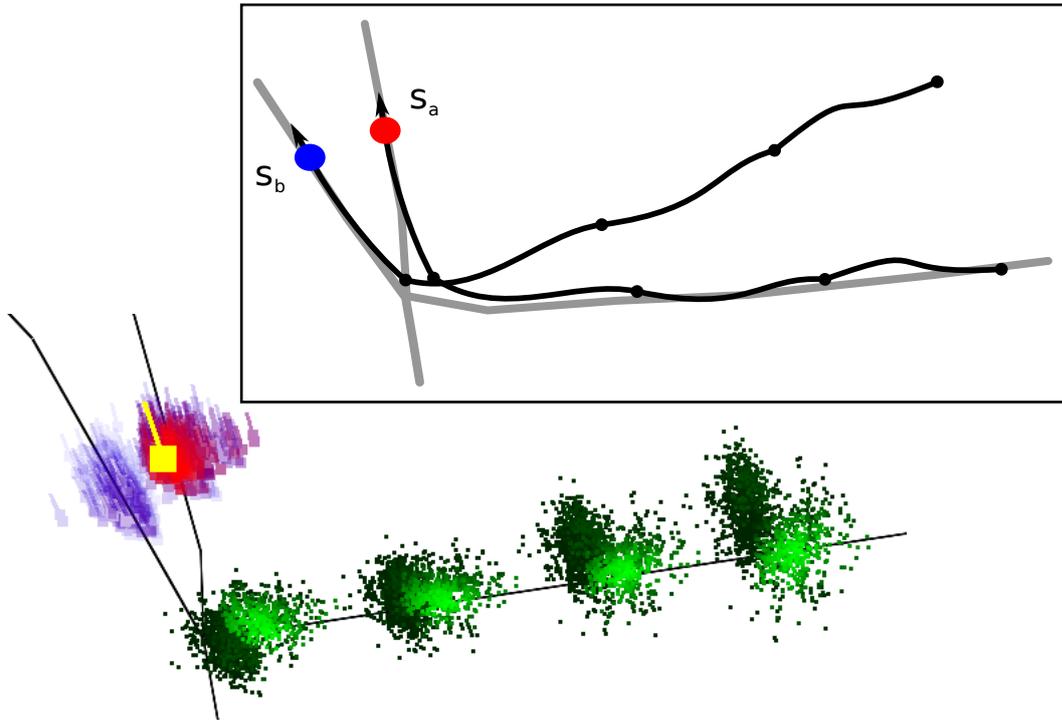


Abbildung 4.8: **Partikelgewichtung unter Berücksichtigung des Bewegungspuffers während eines beispielhaften Abbiegevorgangs des Roboters.** Partikel: Hohe Gewichtung (rot), niedrige Gewichtung (blau). Elemente des lokalen Bewegungspuffers: Hohe Gewichtung (helles Grün), niedrige Gewichtung (dunkles Grün). Aktuelle Schätzung der Fahrzeugpose (gelb).

Zur Verdeutlichung der vorgestellten Methode der Partikelgewichtung zeigt Abb. 4.8 eine beispielhafte Partikelverteilung nach einem Abbiegevorgang des Bodenroboters. Das Partikel  $s_a$  ist ein Beispiel für ein Partikel mit hoher Gewichtung, da der lokale Bewegungspuffer des Partikels gut zur lokalen Form des Wegennetzes passt. Im Gegensatz dazu wird das Partikel  $s_b$  niedrig gewichtet. Zwar liegt das Partikel selbst nahe an einer Wegenetzkante, die hinteren Elemente des Bewegungspuffers jedoch weisen große Kantenabstände

auf. Diese Beispielsituation unterstreicht die herausragende Fähigkeit dieser Gewichtungsmethode, potenziell uneindeutige Partikelfilterverteilungen gerade in schwierigen Abzweig- bzw. Kreuzungssituationen zu vermeiden. Dies stellt einen großen Vorteil gegenüber der Methode von Gustafsson et al. [71] dar.

### Resampling-Strategie

Die Wahl der Resampling-Strategie eines Partikelfilters stellt eine wichtige Entwurfsentscheidung dar. Stets eine ausreichende Anzahl an Partikeln an relevanten Punkten im Zustandsraum zu positionieren, ist essenziell für die erfolgreiche Durchführung des Filterprozesses. Zunächst wird für den hier beschriebenen Filter der klassische Ansatz des *stratified resampling* eingesetzt (siehe Kapitel 4.3.2), da es durch die vorteilhafte Behandlung von multimodalen Wahrscheinlichkeitsverteilungen, gerade im Szenario der kartenbasierten Lokalisierung, besonders gut geeignet ist, weil uneindeutige Partikelverteilungen über eine gewisse Zeit Bestand haben können, bis sie aufgelöst werden können. Als Resampling-Indikator dient die zuvor eingeführte Anzahl der effektiven Abtastpunkte  $N_{eff}$ .

Zusätzlich zum *stratified resampling* wurde jedoch eine neue, speziell für den Anwendungszweck der kartenbasierten Lokalisierung entwickelte Resampling-Methode implementiert. Im Gegensatz zu klassischen Resampling-Verfahren werden bei der neu entwickelten Methode die Partikel nicht durch Duplikation anderer Partikel aus der Partikelmenge reinitialisiert. Stattdessen werden die 15% am schlechtesten bewerteten Partikel der gesamten Partikelmenge nach der in Abb. 4.7 gezeigten Methode auf die jeweils nächste Kante des Wegenetzes projiziert.

Diese neue Variante des Resamplings wird nach jedem Iterationsschritt durchgeführt und erhöht die Robustheit und Genauigkeit des Schätzergebnisses während des Filterprozesses deutlich, gerade bei der Nutzung einer driftenden relativen Lokalisierungsmethode wie der hier benutzten Koppelnavigationsvariante. Stark driftende Koppelnavigationsdaten führen zu einem raschen Entfernen der Partikel von Wegenetzkanten (siehe Abb. 4.11). Dieses Entfernen kann in einigen Fällen so schnell passieren, dass das *stratified resampling* dies nicht kompensieren kann, was häufig zu einem Scheitern des Filters führt. Die neu

entwickelte Resampling-Methode sorgt in diesem Fall dafür, dass eine ausreichende Anzahl von Partikeln in der Nähe von Wegenetzkanten gehalten wird, um so weiterhin eine erfolgreiche Schätzung der Fahrzeugpose zu ermöglichen.

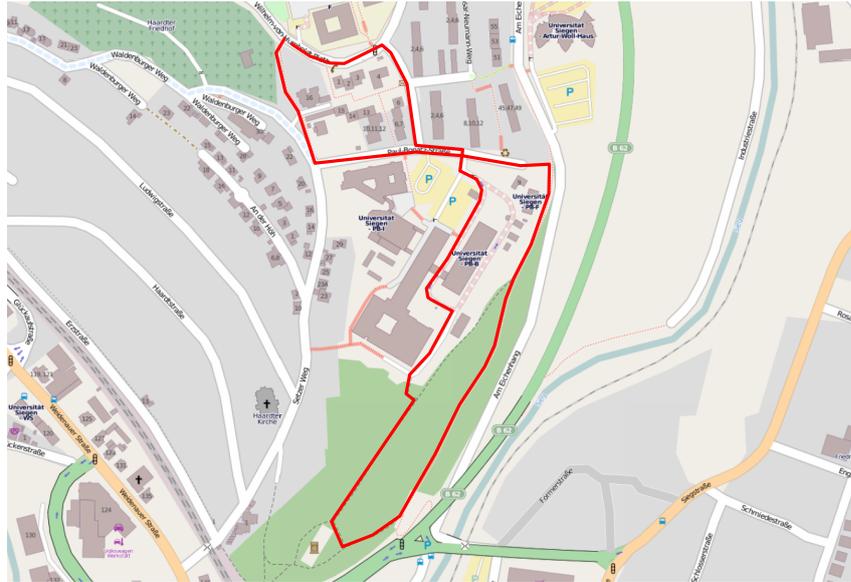
### 4.3.5 Ergebnisse

Das Verhalten des implementierten Partikelfilters wurde anhand mehrerer unterschiedlicher Testfahrten ausgewertet, um eine Überanpassung der Filterparameter zu verhindern. Im folgenden Kapitel werden stellvertretend eine beispielhafte Testfahrt von 1560 m Länge (Abb. 4.9 (a)) sowie einige ausgewählte Beispielszenen (Abb. 4.10, 4.11 und 4.12) analysiert. Der Vergleich mit der in [71] vorgestellten Methode dient jeweils als Referenz während dieser Analyse, um das verbesserte Systemverhalten der in dieser Arbeit entwickelten Lokalisierungsmethode zu demonstrieren. Als absolute Referenzmessungen dienen weiterhin Positionsmessungen, die von einem auf AMOR montierten DGPS-Empfänger generiert wurden. Die Partikelanzahl in den Partikelmengen beider Filterimplementierungen bestand während der Experimente jeweils aus 500 Partikeln. Die Wegenetzdaten für die beschriebenen Experimente entstammen der freien Geodatenbank OpenStreetMap<sup>6</sup>. Um eine Schätzung des lokalen Kartenfehlers zu erhalten, wurde zusätzlich der jeweilige Abstand der mit dem DGPS-Empfänger bestimmten "wahren" Position zur aktuell nächsten Wegenetzkante ermittelt. Da die kartenbasierte Lokalisierung immer nur so gut wie die Qualität der Kartendaten sein kann, sollte dieser Fehler bei der Evaluation stets gedanklich berücksichtigt werden.

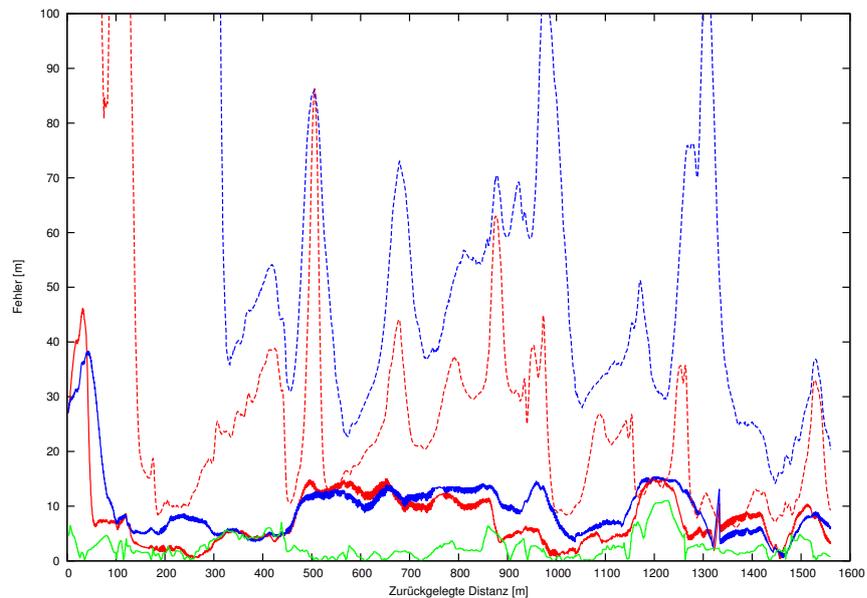
Die in Abbildung 4.9 dargestellten Ergebnisse der Auswertung der exemplarischen Testfahrt lassen sich wie folgt zusammenfassen: Die in dieser Arbeit entwickelte Methode hat im Vergleich stets eine geringere Standardabweichung der Partikelmenge. Der Positionsfehler ist in der überwiegenden Mehrzahl der Fälle geringer als bei der Vergleichsmethode. Weiterhin ist die Konvergenzgeschwindigkeit höher. Mit Berücksichtigung des lokalen Kartenfehlers liegt der absolute Lokalisierungsfehler grundsätzlich unter 15m; in den meisten Fällen sogar deutlich niedriger bei stets unter 5m. Die hohe, gleichbleibende Abweichung zwischen der 500m- und der 850m-Marke ist auf eine lange Geradeaus-

---

<sup>6</sup><http://www.openstreetmap.org>



(a)



(b)

Abbildung 4.9: **Evaluationsergebnis der Partikelfilter-basierten Lokalisierung mit digitalen Kartendaten.** Überblick der gefahrenen Strecke der Testfahrt (a) [136]. Lokalisierungsfehler während der Testfahrt (b). In dieser Arbeit implementierte Methode (rot). Methode nach [71] (blau). Durchgezogene Linien repräsentieren den absoluten Fehler gegenüber der DGPS-Referenzmessung. Gestrichelte Linien beschreiben die Standardabweichung aller Partikelpositionen aus der Partikelmenge. Schätzung des lokalen Fehlers des digitalen Wegenetzes (grün).

fahrt zurückzuführen, während derer beide Filteransätze aufgrund von fehlenden Wegenetzmerkmalen keine Verbesserung der Positionierungsgenauigkeit erreichen konnten. Bei der 850m-Marke jedoch ist der Roboter auf eine Y-Abzweigung getroffen, die mit der in dieser Arbeit implementierten Methode zu einer direkten, deutlichen Verbesserung der Positionsschätzung genutzt werden konnte. In den Abschnitten 300m-450m und 1150m-1250m gibt es ebenfalls einen vergleichsweise hohen absoluten Positionierungsfehler. Dieser ist jedoch in diesem Fall auf die Ungenauigkeit der lokalen Wegenetzdaten zurückzuführen, wie der Verlauf des geschätzten Kartenfehlers anzeigt.

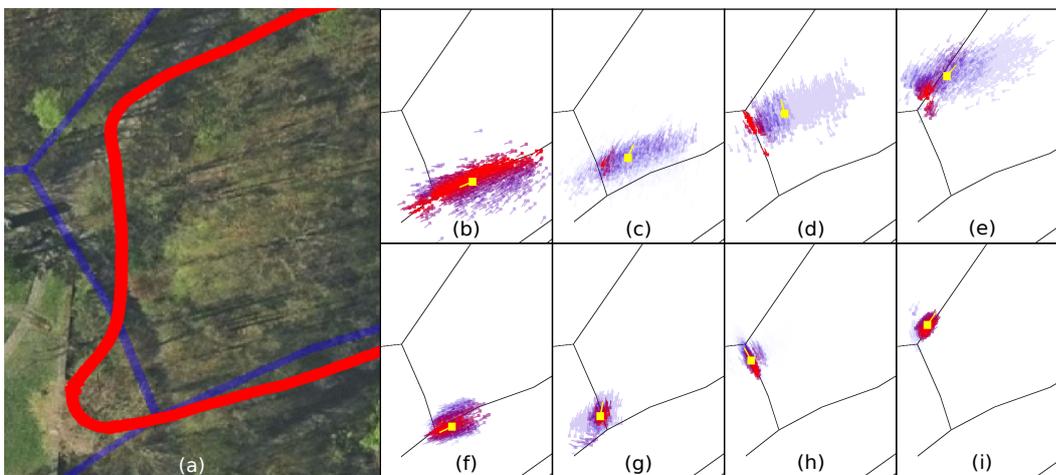


Abbildung 4.10: **Beispielszene mit einer Sequenz zweier scharfer Abbiegevorgänge.** Luftbild überlagert mit Wegenetzkanten (blau) und Positionsdaten der Koppelnavigation (rot) (a). Obere Reihe (b, c, d, e): Methode nach [71]. Untere Reihe (f, g, h, i): In dieser Arbeit entwickelte Methode. Aktuell geschätzte Pose des Roboters (gelb). Partikel: Hohe Gewichtung (rot), niedrige Gewichtung (blau). Copyright des Luftbilds: Google Inc.

Vor dem Hintergrund des vollständigen Verzichts auf absolute Lokalisierungsquellen, wie Daten von satellitenbasierten Positionssensoren, ist die in dieser Arbeit implementierte Lokalisierungsmethode somit eine praktikable Lösung zur Lokalisierung eines mobilen Roboters im unstrukturierten Außenbereichsszenario.

Es folgt nun die Analyse dreier häufig auftretender Situationen, die sehr gut die verbesserte Leistung gegenüber den verglichenen Methoden aus der Literatur

demonstrieren.

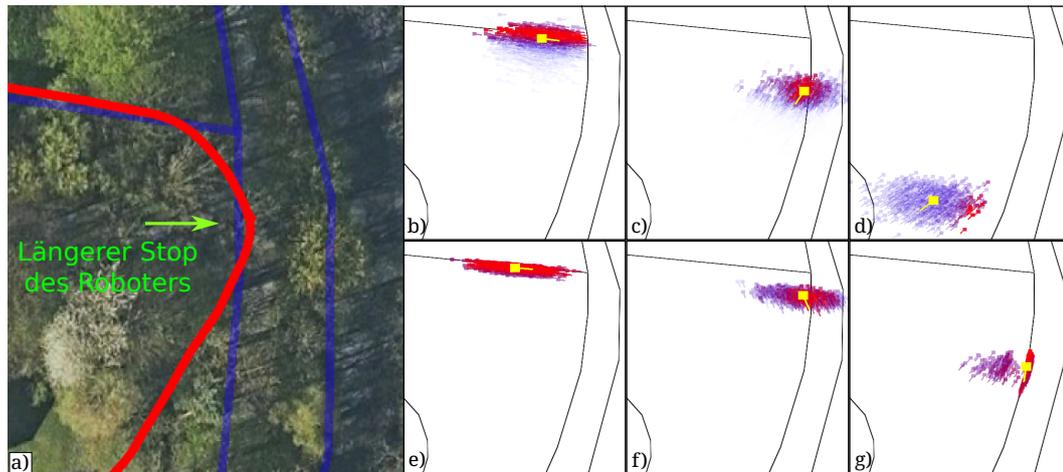


Abbildung 4.11: **Beispielszene mit stark gestörten Koppelnavigationsdaten.** Luftbild überlagert mit Wegenetzkanten (blau) und Positionsdaten der Koppelnavigation (rot) (a). Obere Reihe (b, c, d): Methode nach [71]. Untere Reihe (e, f, g): In dieser Arbeit entwickelte Methode. Aktuell geschätzte Pose des Roboters (gelb). Partikel: Hohe Gewichtung (rot), niedrige Gewichtung (blau). Copyright des Luftbilds: Google Inc.

Die erste Situation zeigt eine Sequenz aus zwei kurz hintereinander ausgeführten scharfen Abbiegevorgängen (siehe Abb. 4.10). Schon vor dem ersten Abbiegevorgang hat die in dieser Arbeit entwickelte Methode eine Partikelmenge mit deutlich geringerer Streuung erzeugt. Während der Abbiegevorgänge ist eine erheblich bessere Verfolgung der Wegenetzkanten zu beobachten, wobei die Streuung der Partikelwolke ebenfalls stets geringer bleibt.

In der zweiten Situation (siehe Abb. 4.11) wird der positive Effekt der zusätzlichen Resampling-Strategie deutlich. Durch eine längere Standphase des Roboters (siehe grüner Pfeil in Abb. 4.11) kam es zu lokal stark gestörten Daten der Koppelnavigation. Im folgenden Verlauf der Fahrt drifteten die Partikel daher stark von der richtigen Wegenetzkante weg. Durch Rückprojektion von schlecht bewerteten Partikeln mit der hier neu eingeführten Methodik wird es möglich, diesen stark störenden Effekt zu kompensieren und somit zu erreichen, dass genug Partikel in der Nähe von Wegenetzkanten verbleiben. Wird dieser zusätzliche Resampling-Schritt nicht durchgeführt, bewegen sich die Partikel

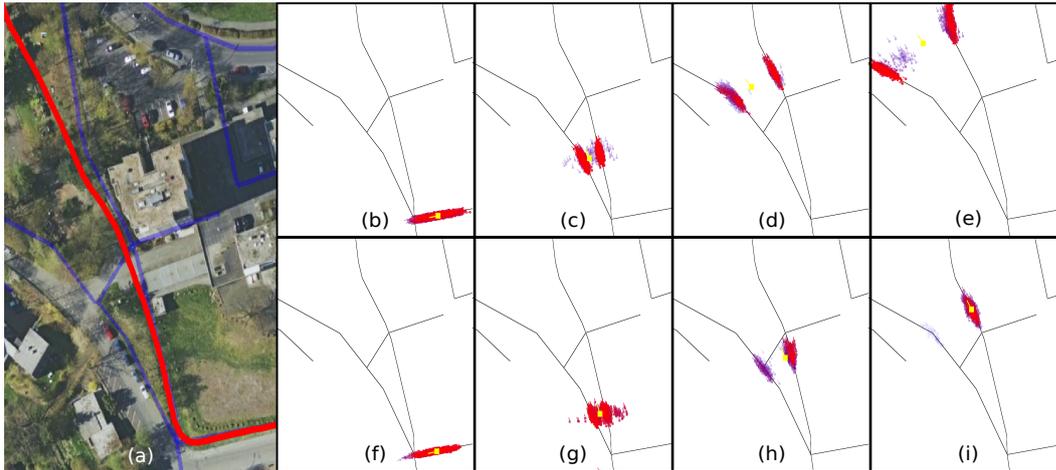


Abbildung 4.12: **Beispielszene einer Y-Abzweigung.** Luftbild überlagert mit Wegenetzkanten (blau) und Positionsdaten der Koppelnavigation (rot) (a). Obere Reihe (b, c, d, e): Methode nach [71]. Untere Reihe (f, g, h, i): In dieser Arbeit entwickelte Methode. Aktuell geschätzte Pose des Roboters (gelb). Partikel: Hohe Gewichtung (rot), niedrige Gewichtung (blau). Copyright des Luftbilds: Google Inc.

solange von der richtigen Wegenetzkante weg, bis eine andere Kante besser bewertet wird, was zu einer nachhaltigen Fehllokalisierung des Roboters führt. Die dritte Situation stellt die häufig auftretende Wegenetzkonfiguration einer Y-Abzweigung dar. Dies ist eine komplexe Situation im Rahmen der kartenbasierten Lokalisierung, gerade wenn ausschließlich relative Lokalisierungsdaten aus der Koppelnavigation zur Propagation der Partikelzustände genutzt werden. Durch die geringe Winkeldifferenz zwischen den zwei optionalen Wegenetzkanten ist eine Aufteilung der Partikelmenge in zwei Partikelhäufungen zu erwarten. Bei einer Gewichtung jeden Partikels ohne Einbeziehung des in dieser Arbeit eingeführten lokalen Bewegungspuffers kann die Uneindeutigkeit zwischen den zwei Partikelhäufungen nicht aufgelöst werden. Wird jedoch mit dem lokalen Bewegungspuffer ein Linie-zu-Linie-Abgleich umgesetzt, konzentrieren sich die Partikel auf eine der optionalen Wegenetzkanten und es wird ein eindeutiges Schätzergebnis erzeugt.

## 4.4 Lokalisierung durch Luft-Boden-Kooperation

Die Kooperation des Bodenroboters mit einem als externe Sensorplattform agierenden Flugroboter eröffnet eine weitere Möglichkeit der absoluten Lokalisierung des Bodenfahrzeugs, ohne auf einen GPS-Empfänger auf dem Bodenfahrzeug zurückgreifen zu müssen. Grundlage ist die Annahme, dass ein Flugroboter, der in einer gewissen Mindesthöhe operiert, nahezu immer auf ein zuverlässiges GPS-Signal zurückgreifen kann. Durch einen Vergleich der Bilder einer nach unten zeigenden Kamera auf der Drohne mit georeferenzierten Bildern aus einer Orthofoto-Datenbank mit der gleichzeitigen Detektion des Bodenroboters im Videobild des Flugroboters wird es möglich, eine absolute Lokalisierung des Bodenroboters durchzuführen [105].

Die Umsetzung dieser neuen Lokalisierungsmethode gliedert sich in zwei getrennte Unterbereiche, die in der Forschung seit Längerem bekannt sind und bearbeitet werden. Auf der einen Seite müssen aktuelle, von einem Flugroboter aufgenommene Luftbilder mit georegistrierten Luftbildern aus einer Geodatenbank verglichen bzw. gegeneinander registriert werden, um die Live-Luftbilder im erdfesten Koordinatensystem zu lokalisieren. Auf der anderen Seite muss der Bodenroboter im Live-Luftbild detektiert und lokalisiert werden.

Lin und Medioni [127] beschreiben eine merkmalsbasierte Methode zur Registrierung von Live-Luftbildern mit georegistrierten Orthofotos. Die Merkmale werden mit der SIFT-Methode nach Lowe [130] erzeugt, wobei Ausreißer unter den jeweils zugeordneten Merkmalen mithilfe des RANSAC-Verfahrens [59] eliminiert werden. In [178] kommt anstatt einer nach unten schauenden Luftbildkamera eine nach vorne schauende zum Einsatz. Dies bedingt die Entwicklung eines anderen Ansatzes zur Registrierung. Gebäudeumrisse im Luftbild spielen dabei eine zentrale Rolle. Zunächst werden diese im Luftbild detektiert und mithilfe von visuellen 3D-Rekonstruktionsmethoden dreidimensional modelliert. Danach kann die rekonstruierte dreidimensionale Form auf die geschätzte Bodenebene projiziert werden. Die projizierten Gebäudeumrisse werden abschließend manuell markierten Gebäudeumrissen im georegistrierten Orthofoto zugeordnet, um die Registrierung herzustellen, wobei ähnliche Methoden wie in [127] verwendet werden. Fan et al. [55] schlagen die Nutzung alternativer Merkmale als "Ankerpunkte" für die Registrierung von Luftbildern vor,

die mit nach unten zeigender Kamera aufgenommen wurden. Kantendetektion und Entropiebetrachtung werden eingesetzt, um darauf aufbauend durch einen Schablonenvergleich eine Registrierung des Live-Luftbilds zu erreichen. Rotation und Skalierung des Luftbilds werden nicht in das Suchverfahren einbezogen, sondern direkt von den Sensoren des Flugroboters bezogen, um eine effiziente Suche zu ermöglichen.

Der andere für den Gesamtkontext der kooperativen Lokalisierung relevante Themenbereich ist die Detektion von Objekten - speziell Bodenfahrzeugen - im Luftbild. In [216] wird in diesem Zusammenhang ein Verfahren präsentiert, dass Pkws, die sich auf befestigten Straßen bewegen, in Luftbildern detektiert. Basierend auf der Ausprägung der Außenkontur eines Fahrzeugs, dessen Windschutzscheibe und dessen Schlagschatten werden Fahrzeuge im Luftbild detektiert, wobei diese anhand eines generischen 3D-Fahrzeugmodells, eines Oberflächenreflexionsmodells, des Einfallswinkels der Sonne und der Kameraparameter validiert werden. Die Klassifikation in verschiedene Fahrzeugtypklassen wird mittels eines Bayesschen Netzes implementiert. Breckon et al. [29] benutzen mehrere kaskadierte Klassifikatoren, die durch überwachtetes Anlernen mit Haar-artigen Merkmalen [203] trainiert wurden, um multiple Fahrzeuge im Luftbild zu erkennen. In [65] wird von Gleason et al. eine Detektionsmethode für künstliche, also von Menschen gemachte Objekte vorgeschlagen, die sich der Annahme bedient, dass Objekte dieser Kategorie eine auffallend hohe Dichte an geraden Kanten und Ecken aufweisen. Durch Auffinden von Regionen, die eine hohe Dichte dieser Merkmale aufweisen, können die Positionen künstlicher Objekte ausfindig gemacht werden. Für die Merkmalsextraktion wird die Methode der Histogramme orientierter Gradienten (HoG) nach [41] eingesetzt und verschiedene Klassifikatortypen werden untersucht, um Fahrzeugobjekte von Nicht-Fahrzeugobjekten zu unterscheiden.

Ein weiterer interessanter Ansatz zur Lokalisierung eines Bodenfahrzeugs mithilfe von georegistrierten Orthofotos soll an dieser Stelle der Vollständigkeit halber ebenfalls genannt werden, obwohl dabei kein Flugroboter als Quelle von aktuellem Bildmaterial zum Einsatz kommt. Vielmehr werden die Bilder einer auf dem zu lokalisierenden Bodenfahrzeug montierten Frontkamera einer perspektivischen Projektion unterzogen, um das gleiche visuelle Erscheinungsbild eines Luftbilds zu erhalten. Danach wird das projizierte Bild ähnlich der

oben beschriebenen merkmalsbasierten Methoden mit einem georegistrierten Orthofoto registriert. Durch Projektion des Kamerabilds werden jedoch Verzerrungen induziert, was sich negativ auf die visuelle Qualität im Vergleich zu einem nativen Luftbild auswirkt. Beispiele für diesen Ansatz sind in [154] und [149] zu finden, wobei beide in diesen Arbeiten vorgestellten Verfahren für die Lokalisierung von Fahrzeugen auf befestigten Straßen mit Straßenmarkierungen ausgelegt sind, um eine ausreichende Anzahl an robusten Merkmalen im Bild zur Verfügung zu haben.

Die Kombination der oben genannten Teilprobleme zu einem integrierten System, das die schritthaltende Lokalisierung eines kooperierenden Bodenroboters in Echtzeit erlaubt, ist zum Zeitpunkt der Veröffentlichung dieser Methode in der verwandten Forschung noch nicht versucht worden und stellt damit einen der zentralen Innovationsaspekte dieser Arbeit dar. Im hier angestrebten Szenario der Boden-Luft-Kooperation sind im Vergleich zu anderen Arbeiten, die sich mit den Teilproblemen der Gesamtaufgabenstellung befassen, folgende Unterschiede festzustellen, welche die Neuentwicklung der im Folgenden vorgestellten Methode bedingen: Es wird von einer niedrigen Flughöhe (ca. 20 m) ausgegangen, was die Stabilität von Verfahren negativ beeinflusst, die auf etablierten Merkmalsextraktoren bzw. -deskriptoren basieren (bspw. SIFT [130] oder SURF [17]). Die Notwendigkeit der Erkennung verschiedener Fahrzeugtypen und -formen ist nicht gegeben, da apriori bekannt ist, welches Fahrzeugobjekt zu erwarten bzw. zu detektieren ist. Weiterhin ist stets der Bodenroboter als störender Faktor im Luftbild präsent und schließlich wird eine sehr geringe Laufzeit des Verfahrens erreicht, um eine möglichst hohe Aktualisierungsrate der Position des Bodenroboters zu ermöglichen.

Abb. 4.13 zeigt eine Übersicht des Ablaufs der kooperativen Lokalisierung. Der Flugroboter nimmt ein Luftbild auf und sendet dies mit zeitlich synchronisierten Poseninformationen zum Bodenroboter. Auf dem Bodenroboter wird das Luftbild mithilfe der von der Drohne kommunizierten Höhe und Orientierung transformiert. Die Transformation hat das Ziel, dass das Live-Luftbild auf Pixelebene mit einem georeferenzierten Orthofoto aus einem geographischen Informationssystem (GIS) verglichen werden kann. Das zu nutzende Orthofoto wird parallel zur Transformation des Luftbilds aus dem GIS geholt, wobei die Positionsdaten, die der GPS-Empfänger des Flugroboters produ-

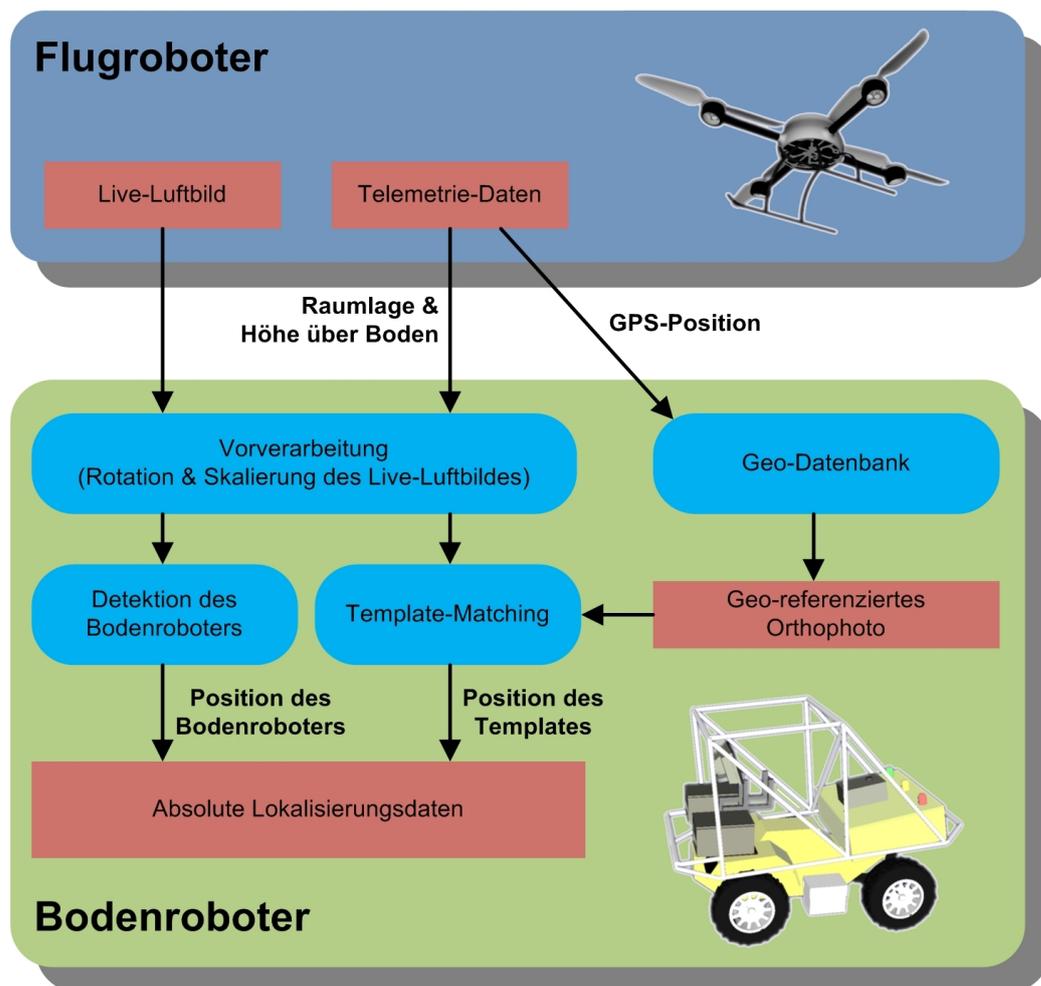


Abbildung 4.13: Ablauf der kooperativen Luft-Boden-Lokalisierung.

ziert hat, genutzt werden, um den passenden Kartenausschnitt im GIS effizient aufzufinden. Ein Schablonenvergleich-Verfahren wird genutzt, um die Korrespondenz zwischen Live-Luftbild vom Flugroboter und georeferenziertem Orthofoto herzustellen. Anschließend wird eine Detektion des Bodenfahrzeugs im Live-Luftbild durchgeführt, um die gewünschte Korrespondenz zwischen Bodenroboter und georeferenziertem Orthofoto herzustellen. Mit bekannter Position des Live-Luftbilds im erdfesten Koordinatensystem und bekannter Position des Bodenroboters im Luftbild-Koordinatensystem kann nun abschließend, unter Zuhilfenahme der intrinsischen Parameter der Luftbildkamera (siehe Kapitel 5.3.2) und der Telemetriedaten des Flugroboters, die Position des Bodenroboters hochpräzise im erdfesten Koordinatensystem festgestellt werden.

#### 4.4.1 Ortho-Registrierung des Luftbildes

In aktuellen geographischen Informationssystemen (GIS) wie z.B. Google Maps<sup>7</sup> oder den Web Map Services (WMS) der einzelnen Bundesländer in Deutschland ist eine große Anzahl an georeferenzierten Luftbildern verfügbar. Daher ist es im Anwendungskontext der Luftbild-Registrierung nötig, einen nur sehr begrenzten Kartenausschnitt in die Betrachtung einzubeziehen, da ansonsten der Rechenaufwand ein praktikables Maß bei Weitem übersteigen würde. Zu diesem Zweck wird die durch den GPS-Empfänger des Flugroboters ermittelte Position genutzt, um den richtigen Kartenausschnitt aus dem GIS zu erhalten. Dass die Genauigkeit dieser Positionsmessung messprinzipbedingt um mehrere Meter von der tatsächlichen Position abweichen kann, ist für den weiteren Ablauf der Registrierung unerheblich, da für die Wahl des richtigen Kartenausschnitts aufgrund seiner Ausmaße keine hoch präzise Position bekannt sein muss. Es ist an dieser Stelle zu betonen, dass dies das einzige Mal ist, dass auf die GPS-Technologie zurückgegriffen wird, zusätzlich ausschließlich der Empfänger auf dem Flugroboter verwendet wird und die Messung keinerlei direkten Einfluss auf die Genauigkeit der resultierenden Lokalisierung des Bodenroboters hat. Ist das richtige Referenz-Luftbild aus dem GIS gefunden, gilt es eine Korrespondenz zwischen aktuellem Live-Luftbild  $I_{\text{uav}}$  und Referenz-Luftbild  $I_{\text{ortho}}$  herzustellen.

---

<sup>7</sup><https://maps.google.de>

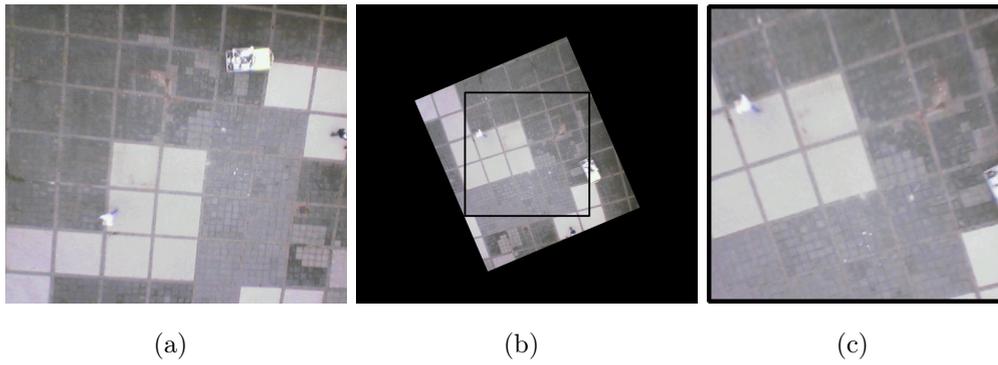


Abbildung 4.14: **Vorverarbeitungsschritte des Live-Luftbildes vor der Nutzung im Registrierungsprozess.** Unverarbeitetes Luftbild (a). Rotation zur Einnordung und Skalierung, abhängig von der Flughöhe des Flugroboters (b). Selektion eines quadratischen Bildausschnitts (c).

Zur Vereinfachung dieses Prozesses wird eine Vorverarbeitung des Live-Luftbildes durchgeführt (siehe Abb. 4.14). Das Ziel dieses Schritts ist, ein eingenordetes Live-Luftbild zu erzeugen, das die gleiche Bodenauflösung wie das Referenz-Orthofoto hat. So ist für eine erfolgreiche Registrierung beider Bilder nur noch die Translation in Bildkoordinaten zwischen den Bildern zu ermitteln. Die Einnordung wird durch eine Rotation des Bildes um den Bildmittelpunkt mit dem vom Inertialsystem ermittelten absoluten Gierwinkel  $\phi_z$  des Flugroboters realisiert. Der Skalierungsfaktor

$$s = \frac{2 h_c \tan \frac{\alpha}{2}}{r_{\text{ortho}}} \quad (4.42)$$

wird mit der Höhe  $h_c$  der Luftbildkamera über Boden, dem Öffnungswinkel der Kamera  $\alpha$  und der Auflösung  $r_{\text{ortho}}$  des georegistrierten Kartenbildmaterials ermittelt. Praktisch wird die Vorverarbeitung durch die Hintereinanderausführung von affinen Transformationen auf  $I_{\text{uav}}$  umgesetzt, die die untransformierten Pixelinformationen  $I_{\text{uav}}(u, v)$  in die transformierten Pixelinformationen  $I'_{\text{uav}}(u, v)$  überführen. Die kombinierte Transformation der Pixelkoordinaten wird durch

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = s \begin{pmatrix} \cos \phi_z & -\sin \phi_z \\ \sin \phi_z & \cos \phi_z \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \quad (4.43)$$

definiert, woraus

$$I'_{\text{uav}}(u, v) = I_{\text{uav}}(s(u \cos \phi_z - v \sin \phi_z), s(u \sin \phi_z + v \cos \phi_z)) \quad (4.44)$$

für die Transformation der Bildinformationen folgt. Das Ergebnis dieses Verarbeitungsschritts ist die Ausrichtung der Hochachse von  $I'_{\text{uav}}$  in nördlicher Richtung und die Übereinstimmung der Bodenauflösung  $I'_{\text{uav}}$  und  $I_{\text{ortho}}$ . Das Verfahren zur Herstellung der Korrespondenz zwischen beiden Bildern muss daher weder rotations- noch skalierungsinvariant sein. Die Korrespondenz zwischen aktueller Messung  $I'_{\text{uav}}$  und Referenz  $I_{\text{ortho}}$  wird daher durch eine Maximumssuche in einem normierten Kreuzkorrelationsbild hergestellt (siehe Abb. 4.15 (a)). Das normierte Kreuzkorrelationsbild  $I_{\text{nk}k}(x, y)$  ist definiert durch

$$I_{\text{nk}k}(u, v) = \frac{\sum_{u'} \sum_{v'} I'_{\text{uav}}(u', v') I_{\text{ortho}}(u + u', v + v')}{\sqrt{\sum_{u'} \sum_{v'} I'_{\text{uav}}(u', v')^2 \sum_{u'} \sum_{v'} I_{\text{ortho}}(u + u', v + v')^2}}. \quad (4.45)$$

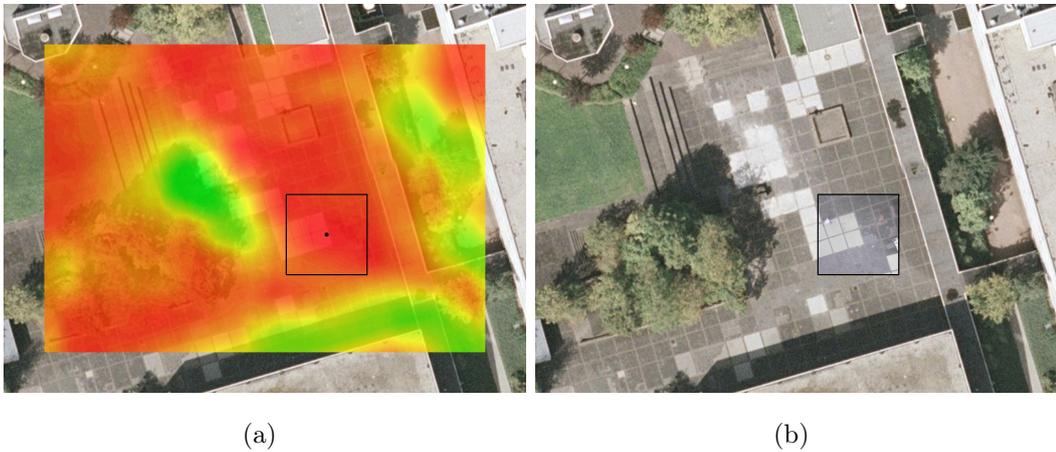


Abbildung 4.15: **Ergebnis des Schablonenvergleichs** (in Falschfarben codiert, grün = minimale Ähnlichkeit, rot = maximale Ähnlichkeit) (a). Das schwarze Rechteck markiert die Position, an der die maximale Ähnlichkeit detektiert wurde. Einfügen des Live-Luftbilds in das Orthofoto an der detektierten Position zeigt die gute Qualität der Registrierung (b). Copyright des Orthofotos: Google Inc.

Um ein effizientes Auffinden der Position von  $I'_{\text{uav}}$  im Referenzbild  $I_{\text{ortho}}$  durchführen zu können, wird zunächst je eine Bildpyramide für  $I'_{\text{uav}}$  und  $I_{\text{ortho}}$  aufgebaut. Die Anzahl der empfehlenswerten Stufen der Bildpyramiden sind dabei abhängig von der absoluten Größe von  $I_{\text{ortho}}$ . Die Korrespondenzsuche wird im Folgenden auf der höchsten Stufe  $k_n$  der Bildpyramide durchgeführt, um eine ungefähre Position des gesuchten Bildes  $I'_{\text{uav}}$  zu erhalten. Danach wird auf niedrigster Bildpyramidenstufe  $k_0$  in einer beschränkten Region-of-Interest eine Korrespondenzsuche nach Gleichung 4.45 mit anschließender Maximumsuche durchgeführt. So wird um die zuvor bestimmte ungefähre Position nach der exakten Position von  $I'_{\text{uav}}$  in  $I_{\text{ortho}}$  gesucht. Ein beispielhaftes Ergebnis dieses Vorgangs ist in Abb. 4.15 (b) zu sehen.

Die erzielbare Genauigkeit der eben beschriebenen Registrierungsmethode ist maßgeblich von der Bodenauflösung  $r_{\text{ortho}}$  von  $I_{\text{ortho}}$  abhängig. Ist diese geringer als die Bodenauflösung  $r_{\text{uav}}$  von  $I_{\text{uav}}$  wird die größere Auflösung von  $I_{\text{uav}}$  nicht in vollem Maße ausgenutzt und die maximal erreichbare Genauigkeit des Verfahrens reduziert. Dies ist der typische Fall beim Einsatz von kleinen Flugrobotern zur Erzeugung von aktuellen Live-Luftbildern, da durch geringe Flughöhen sehr hohe Bodenauflösungen pro Pixel erreicht werden können. In unseren Experimenten ist die physikalische Pixelauflösung der georeferenzierten Luftbilder  $I_{\text{ortho}}$  0.30 Meter pro Pixel. Die Auflösung der Live-Luftbilder mit der in den Experimenten eingesetzten Kamera liegt jedoch bei 0.052 Meter pro Pixel bei einer Flughöhe von 20 Metern. Daher wird per bilinearer Interpolation die Pixelauflösung des Bildes  $I_{\text{ortho}}$  auf 0.06 Meter erhöht, um das zuvor beschriebene Korrelationsverfahren auf Subpixel-Ebene durchführen zu können. Somit wird der größte Anteil der in  $I_{\text{uav}}$  enthaltenen Bildinformation genutzt und so eine maximale Genauigkeit für die Lokalisierung erreicht.

#### 4.4.2 Detektion des Bodenfahrzeugs im Live-Luftbild

Die in 4.4.1 beschriebene Korrespondenzsuche hat eine Geo-Registrierung des aktuell aufgenommenen Luftbilds des Flugroboters  $I_{\text{uav}}$  zum Ergebnis. Es kann also jedem Pixel in  $I_{\text{uav}}$  eine Position in erdfesten Koordinaten zugeordnet werden. Durch die Lokalisierung des Bodenroboters auf Bildebene wird es möglich, auch diesen bez. eines erdfesten Koordinatensystems zu registrieren. Zu diesem

Zweck wird unter Ausnutzung der prominenten Farbe von Teilen des Bodenroboters AMOR ein Farbsegmentierungsverfahren eingesetzt, um den Roboter in  $I_{\text{uav}}$  aufzufinden.

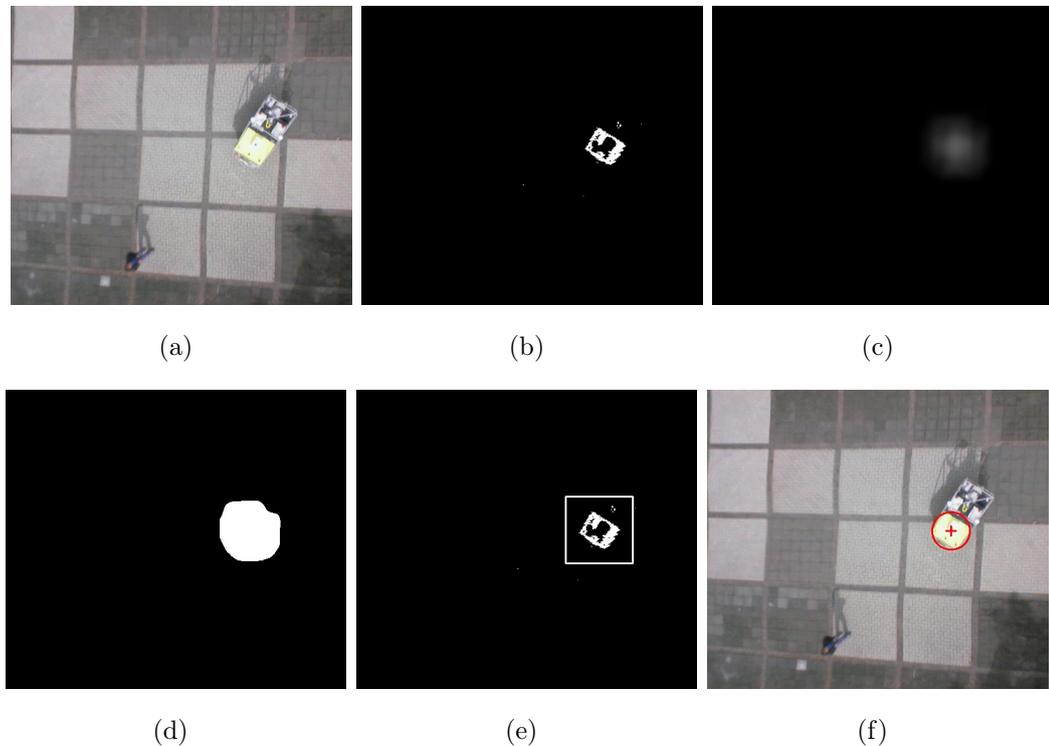


Abbildung 4.16: **Verarbeitungsschritte des Verfahrens zur Detektion des Bodenroboters im Live-Luftbild.** Unverarbeitetes Luftbild (a). Binarisierung durch Farbsegmentierung im HSV-Farbraum (b). Tiefpassfilterung mittels Rechteckfilter (c). Binarisierung durch Anwendung eines festen Schwellwerts (d). Mittelwertbildung in einer Region-of-Interest zur Ermittlung der Position des Bodenroboters im Luftbild (e). Finales Detektionsergebnis (f).

Die eingesetzte Methode setzt sich aus mehreren sequenziellen Verarbeitungsschritten zusammen, die in Abbildung 4.16 zusammenfassend zu sehen sind. Zunächst wird eine Binarisierung des Luftbildes durch pixelweise Analyse der Farbkomponenten im HSV-Farbraum durchgeführt. Anschließend wird eine Tiefpassfilterung des Bildes mit einem Rechteckfilter mit der Anwendung eines festen Schwellwerts kombiniert, um kleinflächige Fehldetektionen im Luftbild zu unterdrücken. Durch Mittelwertbildung der Pixelkoordinaten der Pixel, die im zuvor erzeugten Binärbild verbleiben, wird eine Region-of-Interest definiert.

Sie bildet abschließend die Grundlage, um die gesuchte Position des Bodenroboters im ungefilterten Binärbild ebenfalls durch Mittelwertbildung der Pixelkoordinaten der detektierten Pixel festzustellen.

Die beschriebene Bildverarbeitungsmethode ist ein Vorentwicklungsschritt des optimierten Verfahrens zur Detektion des Bodenroboters im Live-Luftbild, das in Kapitel 5.4.1 im Kontext der kooperativen Umgebungsmodellierung durch Sensorfusion detailliert erörtert wird. Um Wiederholungen zu vermeiden, wird daher auf Kapitel 5.4.1 bezüglich aller algorithmischen Details zum farbsegmentierungsbasierten Detektionsverfahren verwiesen.

### 4.4.3 Ergebnisse

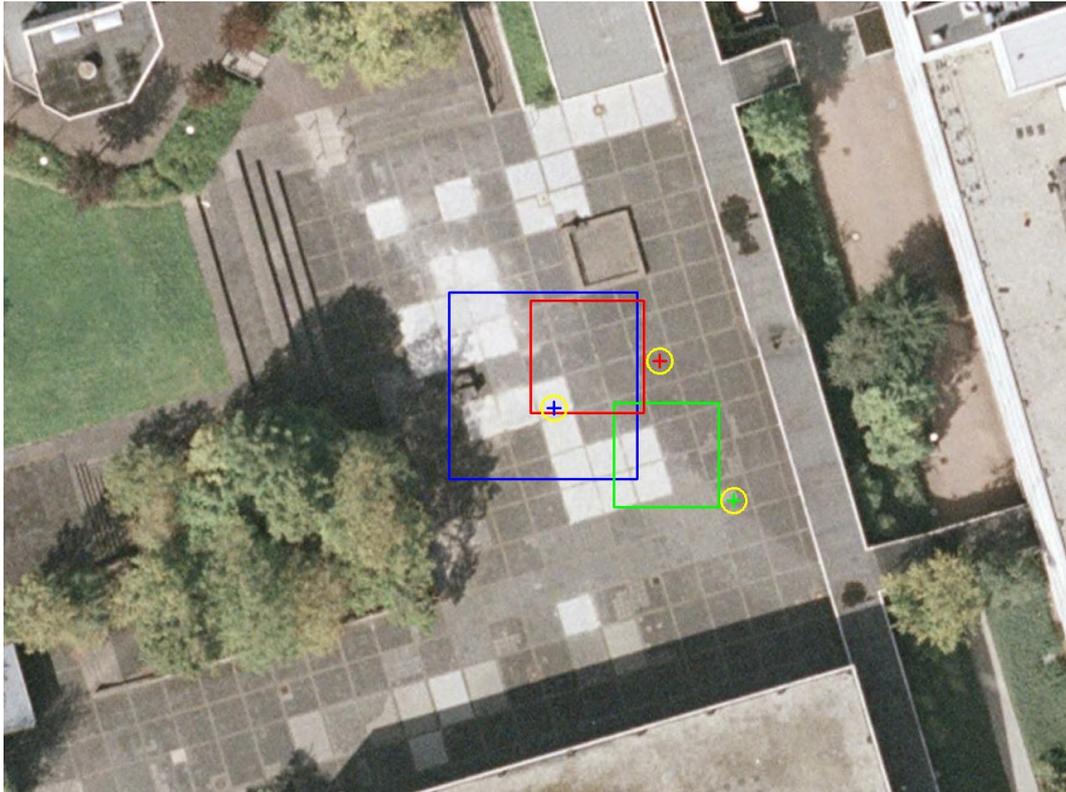
Abbildung 4.17 zeigt drei beispielhafte Ergebnisse der kooperativen Lokalisierung durch Georegistrierung eines Live-Luftbilds. Das Verfahren erzielt gute Lokalisierungsergebnisse sowohl bei verschiedenen Flughöhen des Flugroboters als auch bei nur teilweiser Sichtbarkeit des Bodenroboters im Live-Luftbild (Abb. 4.17 (b)), lokal unterschiedlichem Erscheinungsbild des Live-Luftbild verglichen mit dem Orthofoto (Abb. 4.17 (b)) und leichten Störungen des Live-Luftbilds (Abb. 4.17 (d)).

Die Genauigkeit der beschriebenen kooperativen Lokalisierungsmethode setzt sich aus unterschiedlichen Komponenten zusammen. Das eingesetzte subpixelgenaue Korrelationsverfahren erzeugt eine maximale Registrierungsgenauigkeit von ca. 0.06 Meter (siehe 4.4.1). Die Genauigkeit der Detektion von AMOR im Live-Luftbild liegt bei ca. 0.25 Meter. Dies führt zu einer Gesamtgenauigkeit der Lokalisierung von etwa 0.30 Meter, was vor allen Dingen auf die Einfachheit der eingesetzten Segmentierungsmethode für die Detektion des Bodenroboters zurückzuführen ist. An dieser Stelle besteht erhebliches Optimierungspotenzial. Nichtsdestotrotz ist die Genauigkeit der beschriebenen Methode bereits vergleichbar mit der Lokalisierungsgenauigkeit, die mit DGPS-Empfängern erreicht werden kann.

Abschließend bleibt noch festzuhalten, dass die Laufzeit des Algorithmus pro Luftbild  $I_{\text{uav}}$  den Einsatz in Echtzeit problemlos zulässt. Durchschnittlich dauert die Verarbeitung 76 ms<sup>8</sup> pro verarbeitetem Luftbild.

---

<sup>8</sup>gemessen mit Intel Core2Duo L7100@1.20GHz



(a)



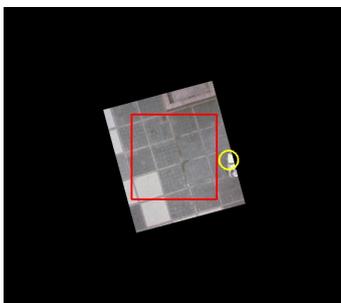
(b)



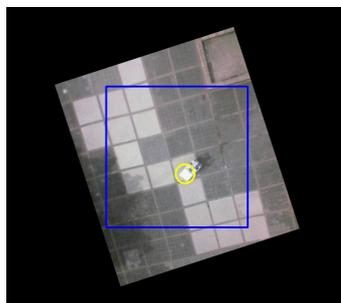
(c)



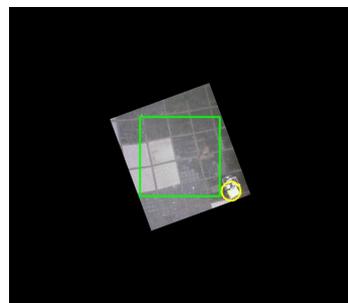
(d)



(e)



(f)



(g)

Abbildung 4.17: **Ergebnisse der kooperativen Lokalisierung des Bodenroboters AMOR** durch Registrierung von Live-Luftbildern (b, c und d) mit einem georegistrierten Orthofoto (a). Copyright des Orthofotos: Google Inc.



# Kapitel 5

## Umgebungsmodellierung durch Sensorfusion

Eine erste Definition des Begriffs der Sensorfusion wurde vom Joint Directors of Laboratories (JDL) Workshop erarbeitet [208]. Die Definition lautet sinngemäß: Sensorfusion ist ein mehrstufiger Prozess, der sich mit der Assoziation, Korrelation und Kombination von Daten und Informationen aus einzelnen oder multiplen Quellen befasst, um verbesserte Objektpositionen zu generieren und Schätzungen, sowie komplette und rechtzeitige Einschätzungen von Situationen und Gefahren zu erzeugen. Hall und Llinas formulierten später in [73] eine allgemeinere Definition, die sich damit auf eine deutlich größere Anzahl von Anwendungs- und Problembereichen beziehen ließ. Sinngemäß heißt es dort, dass Sensorfusion die Kombination von Daten aus mehreren Sensorquellen bedeutet, um so eine erhöhte Genauigkeit und Qualität von Inferenzen zu erreichen, die auf Basis nur eines einzelnen Sensors nicht hätten erreicht werden können.

Gerade, was die erhöhte Qualität von Inferenzen angeht, entspricht dieser Ansatz dem Grundgedanken, der den Verfahren zur Umgebungsmodellierung eines mobilen Roboters in dieser Arbeit zugrunde liegt. Das Ziel der entwickelten Methoden ist die Erzeugung eines möglichst informationsreichen Umweltmodells durch die Zusammenführung der Daten aus verschiedenen Sensorquellen, um so eine möglichst umfassende und fundierte Basis für nachfolgende Planungsentscheidungen des Roboters zu erzielen. Dies beinhaltet, dass durch

den erhöhten Informationsgehalt nicht nur bessere Entscheidungen zu erwarten sind, sondern eventuell sogar Probleme lösbar werden, die zuvor aufgrund mangelnder Informationen keiner befriedigenden Lösung zugeführt werden konnten.

Nach Durrant-Whyte sind in diesem Zusammenhang drei verschiedene Kategorien von Relationen der Sensordaten zueinander zu unterscheiden [53].

- **komplementär** Daten aus mehreren Sensorquellen repräsentieren unterschiedliche Teile einer Szene. Die Fusion der Daten führt zu einer kompletteren Beschreibung der überwachbaren Gesamtszene.
- **redundant** Daten aus mehreren Sensorquellen, die ähnliche Sensoreigenschaften besitzen, repräsentieren gleiche Teile einer Szene. Die Fusion der Daten führt zu einer erhöhten Präzision bzw. einem erhöhten Vertrauen gegenüber den gemessenen Daten. In [32] wird diese Kombination auch als **konkurrierend** bezeichnet.
- **kooperativ** Daten aus mehreren Sensorquellen, die unterschiedliche Sensoreigenschaften besitzen, repräsentieren gleiche Teile einer Szene. Die Fusion der Daten führt zu einer komplexeren, informationsreicheren Beschreibung der Szene.

Dasarathy wählt einen anderen Ansatz und klassifiziert Techniken zur Sensorfusion nach Typ der Ein- bzw. Ausgabedaten [42]. Unterschieden werden die Typen: Rohdaten, Merkmale und Entscheidungen, wobei der Abstraktionsgrad der Daten von Rohdaten über Merkmale bis hin zu den Entscheidungen stets steigt. Datentypen eines bestimmten Abstraktionsgrads können nach Dasarathy im Rahmen der Sensorfusion entweder in Datentypen des gleichen oder eines höheren Abstraktionsgrads überführt werden.

Alle drei von Durrant-Whyte identifizierten Relationskategorien finden sich in den Methoden zur Umgebungsmodellierung wieder, die im Rahmen dieser Arbeit entwickelt wurden, wobei der komplementären und kooperativen Nutzung von Sensoren ein stärkerer Fokus gewidmet wird als der redundanten Nutzung. Was die drei von Dasarathy beschriebenen Datentypen betrifft, kommt in den folgenden Ausführungen der Rohdaten-zu-Rohdaten-Fusion die größte Rolle zu, da es vor allen Dingen darum geht, Daten aus verschiedenen Quellen in einem einheitlichen Modell zusammenzuführen, indem Korrespondenzen

zwischen den Daten hergestellt werden. Während des Prozesses der Korrespondenzfindung werden jedoch teilweise Merkmale verwendet, um die Bestimmung der Korrespondenzen effizient gestalten zu können. Die Datentypen "Merkmaleüind Entscheidungen" nach Dasarathy kommen dann in den Komponenten gehäuft zum Einsatz, die auf Basis des erzeugten fusionierten Umweltmodells Funktionen wie Hindernisdetektion und Pfadplanung realisieren.

## 5.1 Lokaler Bezugspunkt: Die Szene



Abbildung 5.1: **Beispiel einer Szene**, dargestellt durch ein texturiertes Oberflächenmodell der lokalen Umgebung des Roboters, erzeugt mit den in dieser Arbeit entwickelten Methoden.

Um die Verständlichkeit der folgenden Ausführungen zu gewährleisten, soll an dieser Stelle der in dieser Arbeit zentrale Begriff der **Szene** eingeführt werden. Eine Szene sei definiert als die lokale Umgebung des Roboters in Form eines geometrischen Umgebungsmodells, das mit zusätzlichen Informationen angereichert wird. Das geometrische Umgebungsmodell, also das noch nicht mit weiteren Informationen angereicherte Modell, ist zunächst im Kontext dieser Arbeit stets in Form einer Punktwolke (siehe 5.2.2), einem zugehörigen Refe-

renzkoordinatensystem (siehe 5.2.5) und einem festen Aufzeichnungszeitpunkt beschrieben. Die räumliche Ausdehnung einer Szene wird so gewählt, dass sie eine der Geschwindigkeit des Roboters angemessene Planungsvorausschau bietet.

Die auf dem geometrischen Modell aufbauende, angereicherte Form der Umgebungsdarstellung ist stark durch lang bewährte Methoden aus der Computergraphik inspiriert. Konkret heißt dies, dass eine angereicherte Szene einem texturierten Oberflächenmodell der Umgebung des Roboters entspricht. Abb. 5.1 zeigt eine beispielhafte Szene in diesem Zusammenhang. Im Gegensatz zu diversen anderen Beispielen aus der Literatur ist dies eine Mischung aus geometrischer und visueller Darstellung der Umgebung. Des Weiteren wird durch die Nutzung eines Oberflächenmodells eine Beziehung zwischen benachbarten Sensormesspunkten hergestellt. Auch dies ist eher unüblich, da normalerweise ausschließlich Punktwolken aus den Sensormesspunkten erzeugt werden, ohne dabei Aussagen über die Nachbarschaftsverhältnisse der Punkte zu treffen. Der hier vorgestellte Ansatz bringt den Vorteil einer Hinzunahme visueller Informationen erst richtig zur Geltung. Denn die meist deutlich geringere Ortsauflösung der geometrischen Daten würde eine Unterausnutzung der visuellen Daten bedeuten, falls die übliche punktweise Texturierung der aufgezeichneten Punktwolke durchgeführt würde.

Während die geometrischen Informationen der Szene in dieser Arbeit ausschließlich von auf dem Bodenroboter angebrachten Sensoren erzeugt werden, werden die visuellen Informationen sowohl von Sensoren auf dem Bodenroboter als auch von Sensoren auf dem kooperierenden Flugroboter generiert.

Da die Erzeugung von lokalen Szenen schritthaltend während der Bewegung der Roboter passiert, entsteht eine Topologie von Szenen aus dem kontinuierlichen Sensordatenstrom. Diese Topologie von Szenen bildet das Ergebnis der Umweltmodellierung durch Sensorfusion während der autonomen Exploration des Roboterteams.

In den nächsten Kapiteln dieser Arbeit wird die Thematik der Umweltmodellierung durch Sensorfusion daher in die folgenden Bestandteile zerlegt: Betrachtung der geometrischen Beschreibung der Umwelt (siehe Kap. 5.2), Betrachtung der visuellen Beschreibung der Umwelt (siehe Kap. 5.3), Herstellung der Korrespondenz zwischen diesen Beschreibungsformen. Dabei wird der speziel-

le Fall, dass die zu fusionierenden Sensoren auf mehreren Fahrzeugen verbaut sind, wegen der speziellen Schwierigkeiten besonders betrachtet wird.

## 5.2 Geometrische Beschreibung der Umgebung

### 5.2.1 Sensorik

Zur geometrischen Erfassung der Umgebung stehen verschiedene Sensortypen zur Verfügung. In dieser Arbeit wird für die geometrische Umgebungsmodellierung der wohl verbreitetste und zuverlässigste Sensortyp in der Außenbereichsrobotik eingesetzt: der Laserscanner. Auch die Nutzung eines stereoskopischen Kamerasystems mit entsprechenden Algorithmen zur Erzeugung geometrischer Informationen aus stereoskopischen Bildern wurde in diesem Zusammenhang geprüft. Dieser Ansatz wurde jedoch aufgrund des damit verbundenen Aufwandes zur Detektion und Eliminierung von Fehlmessungen verworfen, um eine möglichst robuste Umgebungsrekonstruktion zu ermöglichen. Abb. 5.2 vergleicht das Ergebnis der Umgebungsmodellierung mit einem Laserscanner auf der einen Seite und einer stereoskopischen Kamera auf der anderen Seite.



(a) Laserscanner.

(b) Stereokamera.

Abbildung 5.2: **Punktvolke mit Farbinformationen einer lokalen Szene.** Vergleich zwischen Daten aus einem Laserscanner (a) und einer stereoskopischen Kamera (b).

Ein Laserscanner tastet die Umgebung mit Lichtstrahlen ab, indem von einer Laserdiode erzeugte Lichtimpulse ausgesendet werden. Die Reflexionen der

ausgesendeten Strahlen werden im Scanner registriert. Daraufhin wird eine Laufzeitmessung zwischen dem Aussenden des Strahls und dem Eintreffen des reflektierten Lichts durchgeführt. Die Laufzeit des Strahls lässt einen direkten Rückschluss auf die Entfernung des Sensors zu einem abgetasteten Objekt zu, da die Bewegungsgeschwindigkeit des ausgesendeten Lichts bekannt ist.

Ein einziger in einer festen Richtung ausgesandter Strahl erlaubt allerdings keine flächige Abtastung einer Szene. Es können verschiedene Strategien angewendet werden, um dieses Problem zu lösen. Der Strahl kann entweder so von einem im Strahlengang positionierten Spiegel abgelenkt werden, dass bei einer Rotation des Spiegels verschiedene Abstrahlwinkel erzeugt werden und somit eine sternförmige Sequenz von Strahlen zur Abtastung der Umgebung erzeugt wird.

Die Abtastung der Umwelt mit sternförmig ausgesendeten Strahlen entspricht einem gesampelten Ebenenschnitt mit der Umgebung. Um die geforderte flächige Abtastung der Umgebung tatsächlich zu erreichen, kann eine Rotation der Scan-Ebene durchgeführt werden. Wulf und Wagner [210] unterscheiden dabei verschiedene Arten der Rotation. Abbildung 5.3 zeigt die verschiedenen Varianten und gibt Aufschluss darüber, wie sich die entstehenden Punktwolken hinsichtlich ihrer lokalen Punktedichte bei einer konstanten Rotationsgeschwindigkeit unterscheiden. In allen vier Varianten ist die Punktedichte in der Nähe der jeweiligen Rotationsachse am größten.

Zusätzlich können weitere Laserstrahlen parallel erzeugt werden, um nicht nur einen Abtastpunkt je Zeitschritt, sondern gleich mehrere parallel zu erzeugen. Beispiele für diesen Ansatz sind die rotierenden Laserscanner des Herstellers Velodyne<sup>1</sup> und die gleichzeitig in mehreren Scan-Ebenen messenden Systeme von Ibeo<sup>2</sup> und Sick<sup>3</sup>.

Von einem Laserscanner erzeugte Messungen stehen messprinzipbedingt zunächst in (räumlichen) Polarkoordinaten zur Verfügung (siehe Abb. 5.4). Ein gemessener Abtastpunkt  $P^\times$  wird demnach zunächst durch das Tripel  $(r, \theta, \varphi)$  beschrieben. Dabei bezeichnet  $r$  die gemessene Distanz zum abgetasteten Objekt,  $\theta$  den Polarwinkel zwischen dem Ortsvektor  $P$  des Abtastpunktes  $P^\times$  und

<sup>1</sup><http://velodynelidar.com/lidar/hdlproducts/hdl64e.aspx>

<sup>2</sup><http://www.ibeo-as.com/laser-scanner-portfolio.html>

<sup>3</sup>[http://www.sick.com/de/de-de/home/products/product\\_portfolio/laser\\_measurement\\_systems/Seiten/outdoor\\_laser\\_measurement\\_technology.aspx](http://www.sick.com/de/de-de/home/products/product_portfolio/laser_measurement_systems/Seiten/outdoor_laser_measurement_technology.aspx)

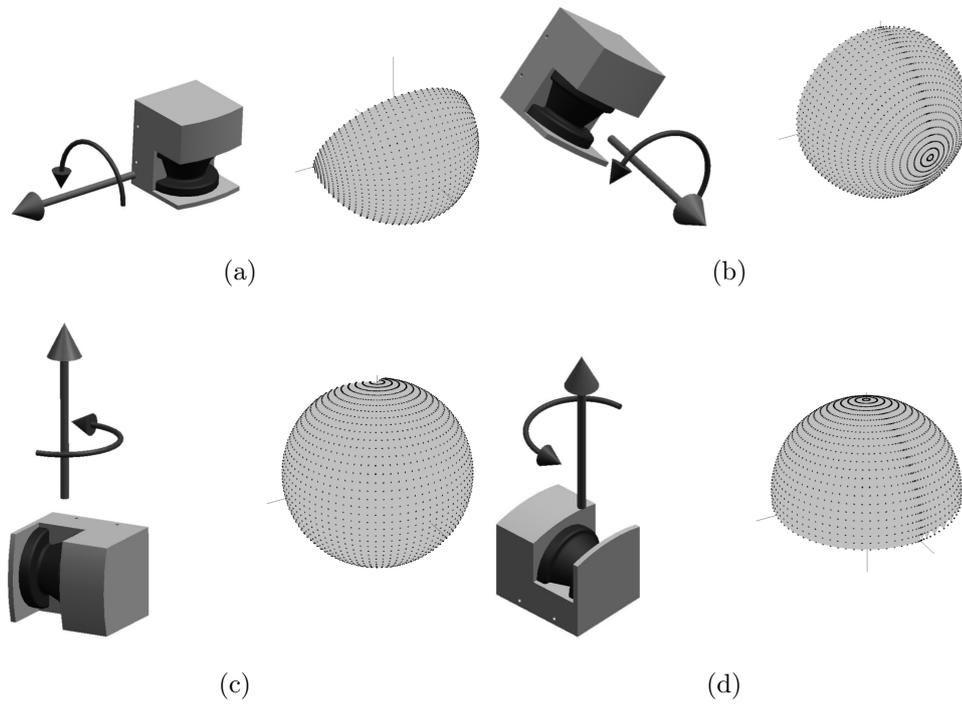


Abbildung 5.3: **Rotationsvarianten planarer Laserscanner zur Erzeugung von Punktwolken** (nicken (a), rollen (b) und gieren (c, d)). Abbildungen aus [210].

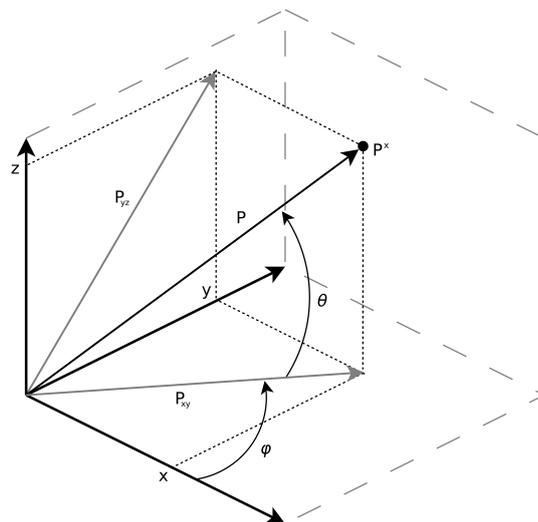


Abbildung 5.4: **Zusammenhang zwischen räumlichen Polarkoordinaten und kartesischen Koordinaten.**

der senkrechten Projektion  $P_{xy}$  von  $P$  auf die XY-Ebene und  $\varphi$  den Azimutwinkel zwischen positiver x-Achse und  $P_{xy}$ . Die oft intuitivere Darstellung als kartesisches Koordinatentripel  $(x, y, z)$  kann mit

$$x = r \cos \varphi \quad (5.1)$$

$$y = r \sin \varphi \cos \theta \quad (5.2)$$

$$z = r \sin \varphi \sin \theta \quad (5.3)$$

berechnet werden. Im Fall eines rotierten, planaren Laserscanners bei dem die Scan-Ebene genickt wird, entspricht der Polarwinkel  $\theta$  der von der Rotationseinrichtung gelieferten Verkippung des Sensors und der Azimutwinkel  $\varphi$  dem vom Scanner selbst gelieferten Ablenkungswinkel des aktuellen Strahls. Diese einfache Transformation gilt jedoch nur für den Fall, dass die Rotationsachse deckungsgleich mit der x-Achse des nativen Sensorkoordinatensystems verläuft. Ist dies nicht der Fall, müssen die gemessenen Abtastpunkte so transformiert werden, dass sie auf ein fahrzeugfestes Koordinatensystem bezogen werden, da das native Sensorkoordinatensystem während der aktiven Rotation durch die Rotationseinrichtung ständig gegenüber dem fahrzeugfesten Koordinatensystem verschoben wird. Im Fall des genickten Laserscanners, der auf dem Roboter AMOR montiert ist, werden die zuvor definierten Ausdrücke zu

$$x = r \cos \varphi + x_0 \quad (5.4)$$

$$y = r \sin \varphi \cos \theta + y_0 \cos \theta - z_0 \sin \theta \quad (5.5)$$

$$z = r \sin \varphi \sin \theta + y_0 \sin \theta - z_0 \cos \theta, \quad (5.6)$$

erweitert, wobei  $(x_0 \ y_0 \ z_0)^T$  dem Verschiebungsvektor von fahrzeugfestem zu nativem Sensorkoordinatensystem entspricht.

## 5.2.2 Geometrische Messungen als Punktwolken

Eine Punktwolke  $C \subset \mathbb{R}^3$  ist definiert als echte Teilmenge von  $\mathbb{R}^3$  mit einer endlichen Anzahl an Elementen  $P_i$  mit  $i \in \mathbb{N}$ . Eine Punktwolke enthält beispielsweise alle Abtastpunkte in Vektorform, die während eines Abtastvorgangs der

Umwelt mit einem Sensor entstanden sind. Die Punktwolke als Darstellungsform von Sensormessungen stellt die generische Form der Beschreibung von geometrischen (Roh-)Messungen im Rahmen dieser Arbeit dar und kann für verschiedene Sensortypen, die geometrische Messungen erzeugen, gleichermaßen genutzt werden. Abb. 5.5 zeigt die Visualisierung einer Punktwolke, die von einem Laserscanner aufgenommen wurde, farbkodiert entsprechend der Höhe über der Bodenebene.

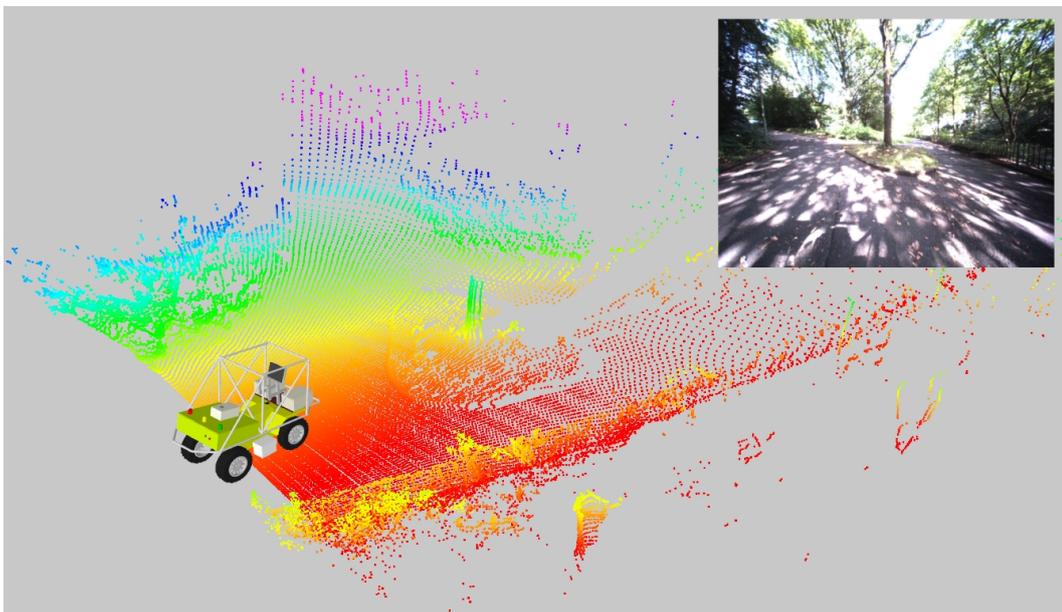


Abbildung 5.5: **Beispiel einer Punktwolke einer Außenbereichsszene.** Punkte sind entsprechend ihrer Höhe über der Bodenebene in Falschfarben gefärbt.

Grundsätzlich entspricht diese Darstellung zunächst einer Punktwolke ohne Ordnung, die keinerlei Informationen über die Struktur der in ihr enthaltenen Punkte beinhaltet; wie etwa Nachbarschaftsbeziehungen zwischen den Punkten. Diese aufgrund ihrer Einfachheit äußerst gängige Darstellungsform kann jedoch bei Bedarf alternativ durch eine Matrixdarstellung erweitert werden,

die implizit die Nachbarschaften der Punkte kodiert:

$$M_C = \begin{pmatrix} P_{11} & \dots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{m1} & \dots & P_{mn} \end{pmatrix} \mid n, m \in \mathbb{N} \quad (5.7)$$

Benachbarte Matrixelemente stellen in dieser Darstellungsform Abtastpunkte dar, die aus benachbarten Polar- respektive Azimutwinkeln entstanden sind. Diese zusätzliche Information kann später beispielsweise zur beschleunigten Generierung von Oberflächen aus gescannten Punktwolken genutzt werden (siehe 5.2.7). Gerade im Fall von sequenziell aufgenommenen Scanzeilen eines rotierten, planaren Laserscanners kann eine solche Matrixdarstellung sehr einfach implementiert werden, indem Scanzeilen als Zeilenvektoren

$$l = (P_1 \ P_2 \ \dots \ P_n) \mid n \in \mathbb{N} \quad (5.8)$$

in die Matrix

$$M_C = \begin{pmatrix} l_1 \\ \vdots \\ l_m \end{pmatrix} \mid m \in \mathbb{N} \quad (5.9)$$

übernommen werden.

### 5.2.3 Hierarchische Definition von Koordinatensystemen

An dieser Stelle folgt ein Überblick über die beteiligten Bezugskoordinatensysteme, die bei der Rekonstruktion der Umwelt des Roboters in Form lokaler Szenen eine Rolle spielen, und deren Relationen zueinander. Die verwendeten Koordinatensysteme werden als Hierarchie von Koordinatensystemen beschrieben. Das heißt, ausgehend von einem globalen, festen Weltbezugskoordinatensystem werden sukzessive verschiedene Koordinatensysteme abgeleitet, um beispielsweise die räumliche Beziehung von Fahrzeug zu lokaler Szene oder

Sensor zu Fahrzeug zu beschreiben. Die hierarchische Struktur der verwendeten Koordinatensysteme lässt sich in Form eines gerichteten Baums

$$G = (\mathbb{F}, \mathbb{T}) \quad (5.10)$$

darstellen, wobei die Knoten  $\mathbb{F}$  des Baums  $G$  Koordinatensysteme darstellen und die Kanten  $\mathbb{T}$  Transformationen, welche die Relation der Koordinatensysteme zueinander definieren. Transformationen zwischen im Baum direkt mit einer Kante verbundenen Koordinatensystemen werden durch relative Posen beschrieben. Der Vorteil dieser Darstellungsweise verglichen mit einer Definition aller relevanten Koordinatensysteme bezüglich eines einzigen, globalen Koordinatensystems, entsteht aus zwei Gründen: Erstens ändert sich die Lage vieler involvierter Koordinatensysteme zueinander entweder gar nicht oder nur selten. Zweitens ist die Bestimmung der relativen Lagen der Koordinatensysteme in hierarchischer Form deutlich einfacher und intuitiver. Die Transformation  ${}^b_aT \in \mathbb{T}$ , definiert durch das Tripel

$${}^b_aT = (f_a, f_b, {}^b_a\check{\rho}), \quad (5.11)$$

enthält das Eltern-Koordinatensystem  $f_a \in \mathbb{F}$ , das Kind-Koordinatensystem  $f_b \in \mathbb{F}$  und den geometrischen Zusammenhang zwischen beiden in Form der Pose  ${}^b_a\check{\rho}$ . Die Pose  ${}^b_a\check{\rho}$  ist nach Definition 4.19 aus Translations- und Rotationskomponente zusammengesetzt. Für die folgenden Ausführungen werden die Translationskomponente als Translationsvektor  ${}^b_a p \in \mathbb{R}^3$  und die Rotationskomponente als Einheitsquaternion  ${}^b_a q \in \mathbb{H}$  aus dem Posen-Tupel  ${}^b_a\check{\rho}$  verwendet. Auf Basis dieser Definition ist die Transformation eines bez. des Koordinatensystems  $f_a$  definierten Punktes  $P^\times$  bzw. seines Ortsvektors  $P$  in das Koordinatensystem  $f_b$  definiert durch

$$P' = P - {}^b_a p \quad (5.12)$$

$$\check{P}'' = {}^b_a \bar{q} \check{P}' {}^b_a q, \text{ mit } \check{P}' = \begin{pmatrix} 0 \\ P' \end{pmatrix} \mid \check{P}' \in \mathbb{H}. \quad (5.13)$$

Diese Transformation wird im Folgenden als Vorwärtstransformation bezeichnet. Die Invertierung dieser Transformation, im Folgenden als Rückwärtstransformation bezeichnet, entspricht einer Transformation des bez. des Koordinatensystems  $f_b$  definierten Vektors  $Q$  in das Koordinatensystem  $f_a$  und ist definiert durch

$$\check{Q}' = {}^b_a q \check{Q} {}^b_a \bar{q}, \text{ mit } \check{Q} = \begin{pmatrix} 0 \\ Q \end{pmatrix} \mid \check{Q} \in \mathbb{H} \quad (5.14)$$

$$Q'' = Q' + {}^b_a p. \quad (5.15)$$

Durch eine sequenzielle Ausführung mehrerer Transformationen kann somit jedes Koordinatensystem  $f_a \in \mathbb{F}$  in ein beliebiges anderes Koordinatensystem  $f_b \in \mathbb{F}$  überführt werden, solange beide Koordinatensysteme Teil des gleichen Baumes  $G$  sind. Das bedeutet implizit, dass mit dieser Methode auch alle Punkte, die bezogen auf ein Koordinatensystem  $f_a \in \mathbb{F}$  definiert sind, bezüglich jedes anderen Koordinatensystems aus  $G$  dargestellt werden können. Dies ist im Folgenden besonders hilfreich, wenn es darum geht, Daten aus verschiedenen Sensorquellen auf einem sich bewegenden Roboter in ein konsistentes Modell zu überführen, das bezüglich eines festen, lokalen Koordinatensystems definiert ist. Ergänzend sei an dieser Stelle erwähnt, dass alle Koordinatensysteme, die Teil des Baums  $G$  sind, kartesische Koordinatensysteme sind.

Folgende Koordinatensysteme sind für den Roboter AMOR im Rahmen dieser Arbeit definiert:

- $f_w$  Festes Weltbezugskordinatensystem. In einem erdfesten Koordinatensystem definiert (siehe Kapitel 4).
- $f_{n_0}, f_{n_1}, \dots$  Lokale Szenenkoordinatensysteme.  $f_{n_0}$  ist definiert durch die Pose des Roboters beim Systemstart, bezogen auf  $f_w$ .
- $f_k$  Initiales lokales Szenenkoordinatensystem. Entspricht dem Szenenkoordinatensystem  $f_{n_0}$  und stellt das Bezugskordinatensystem der Koppelnavigation dar.

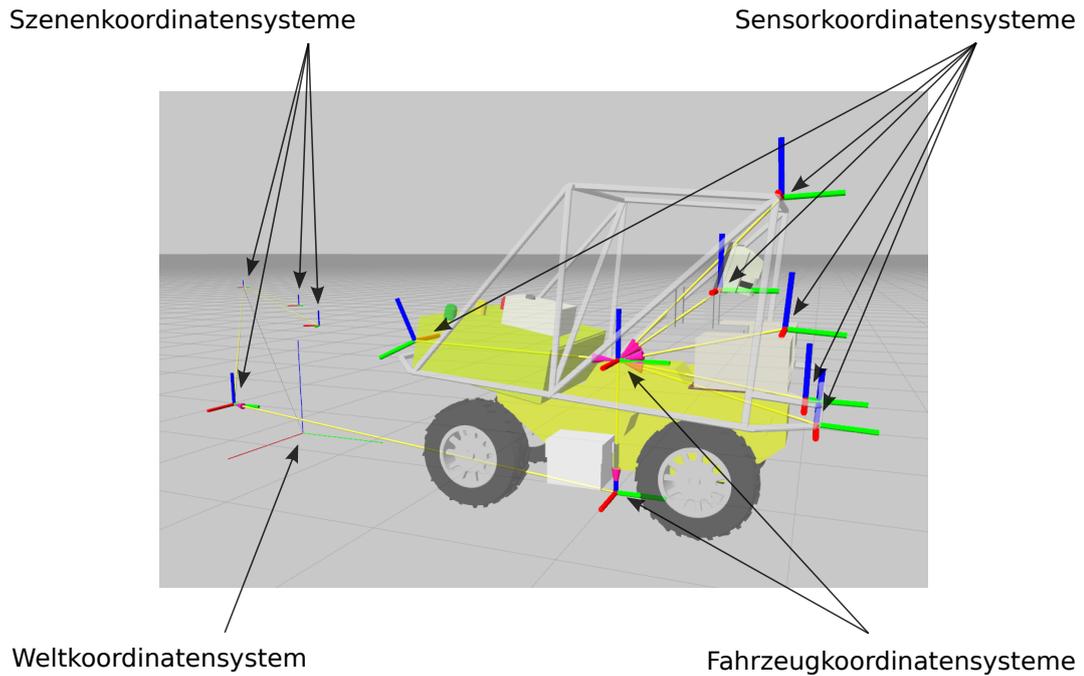


Abbildung 5.6: **Hierarchie der Koordinatensysteme für die Umgebungsrekonstruktion auf dem Roboter AMOR.**

- $f_n$  Aktuelles lokales Szenenkoordinatensystem. Aktueller lokaler Bezugspunkt aller seit Szenenerzeugung entstandenen Sensormessungen. Entspricht dem neuesten Koordinatensystem aus der Menge der Szenenkoordinatensysteme  $f_{n_0}, f_{n_1}, \dots$
- $f_v$  Fahrzeugkoordinatensystem, dessen Ursprung im Rotationszentrum des Fahrzeugs liegt.
- $f_{v'}$  Projiziertes Fahrzeugkoordinatensystem. Projektion von  $f_v$  entlang der z-Achse auf die approximierte Bodenebene, die durch die vier Berührungspunkte der Fahrzeugräder mit dem Untergrund näherungsweise beschrieben wird.
- $f_{s_0}, f_{s_1}, \dots$  Sensorkoordinatensysteme. Definieren Posen von Sensoren, bezogen auf das Fahrzeugkoordinatensystem  $f_v$ .

Der hierarchische Zusammenhang zwischen den genannten Koordinatensystemen ist in Abbildung 5.7 dargestellt. Es gilt zu beachten, dass die Kante, die  $f_{v'}$  und das aktuelle Szenenkoordinatensystem  $f_n$  verbindet, nach Erzeu-

gung einer neuen Szene eliminiert wird und durch eine neue Kante ersetzt wird, die dann  $f_{v'}$  und das neue aktuelle Szenenkoordinatensystem  $f_n$  verbindet. Weiterhin wird zusätzlich das Koordinatensystem  $f_s$  verwendet, das ein Hilfskoordinatensystem darstellt. Es wird als unspezifischer Platzhalter für ein beliebiges spezifisches Sensorkoordinatensystem eingesetzt, solange keine Unterscheidung unterschiedlicher Sensorkoordinatensysteme nötig ist, um die folgenden Ausführungen zu vereinfachen.

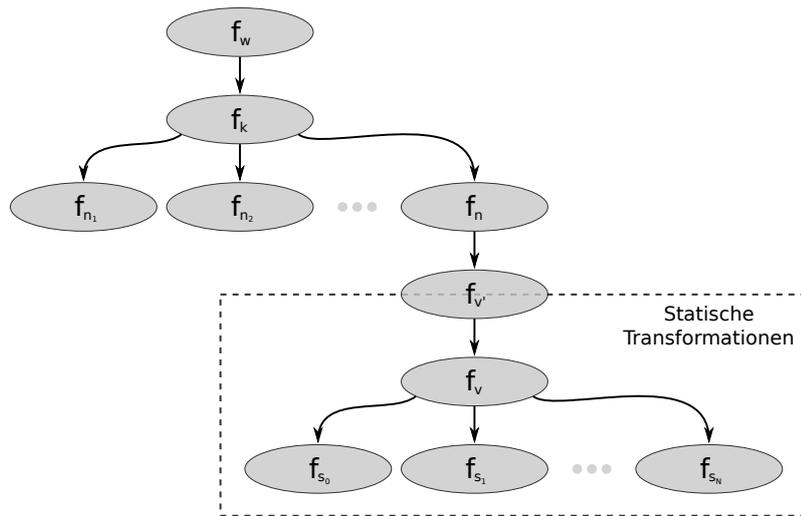


Abbildung 5.7: **Hierarchie der Koordinatensysteme in Baumdarstellung.** Statische Transformationen sind definiert als nach einer initialen Kalibrierung konstante Transformationen, die im Fall des hier abgebildeten Baums die Transformationen zwischen Fahrzeug- und Sensorkoordinatensystemen festlegen.

#### 5.2.4 Statische Transformationen im fahrzeugfesten Koordinatensystem

Die Vorüberlegungen zu den Abhängigkeiten der relevanten Koordinatensysteme werden nun eingesetzt, um Sensormessungen, die zunächst bezüglich eines lokalen Sensorkoordinatensystems  $f_s$  vorliegen, in das aktuelle Szenenkoordinatensystem  $f_n$  zu projizieren. Dabei werden zunächst die statischen, also nicht über die Zeit veränderlichen Transformationen durchgeführt, um die Messung statt bezüglich des zugehörigen Sensorkoordinatensystems  $f_s$  bezüglich des

Fahrzeugkoordinatensystem  $f_v$  darzustellen, das für alle Sensoren einheitlich ist. Damit muss keine Unterscheidung von geometrischen Daten aus unterschiedlichen Sensorquellen mehr gemacht werden und bei stehendem Fahrzeug entsteht bereits ein konsistentes, fusioniertes Modell mit Daten aus verschiedenen Sensorquellen.

### Relative Lage der Sensorkoordinatensysteme

Bei bekannter relativer Lage von  $f_s$  zu  $f_v$  kann ein Messpunktvektor  $P$  mit den Transformationen 5.14 und 5.15 direkt von  $f_s$  in  $f_v$  projiziert werden. Die Hauptkomplexität dieser Aufgabe besteht daher nicht in der tatsächlichen Durchführung der eigentlichen Transformation der Messungen, sondern vielmehr in der Bestimmung der relativen Lage der Koordinatensysteme zueinander. Um diese Aufgabe zu lösen, muss eine Methode zur Bestimmung der Lage auf Basis der Vermessung einer Referenzszene entwickelt werden, da eine händische Messung mittels Zollstock o.ä. nicht die gewünschte Genauigkeit erzielen würde. Die entwickelte Methode nutzt die hohe Messgenauigkeit der Laserscanner direkt, um eine präzise Registrierung der Sensoren per Messung experimentell zu bestimmen.

Zunächst wird die Verkippung der Sensoren bez. einer imaginären idealen Bodenebene bestimmt, um die relativen Nick- und Rollwinkel der Sensoren zu bestimmen. Zu diesem Zweck wird eine möglichst plane Fläche vermessen, auf welcher der Roboter, der die Vermessung durchführt, ebenfalls steht. Zur Bestimmung der Verkippung wird ein effizientes Verfahren zur Ebenenschätzung aus einer Menge an Messvektoren  $P_i = (x_i \ y_i \ z_i)^T \in \mathbb{R}^3$  eingesetzt. Zunächst wird eine Singulärwertzerlegung der Kovarianzmatrix  $A \in \mathbb{R}^{3 \times 3}$  zur Menge aller Punkte  $P_i$  durchgeführt. Die Kovarianzmatrix wird konstruiert durch

$$A = \sum_{i=1}^N \begin{pmatrix} (x_i - x_\mu)^2 & (x_i - x_\mu)(y_i - y_\mu) & (x_i - x_\mu)(z_i - z_\mu) \\ (x_i - x_\mu)(y_i - y_\mu) & (y_i - y_\mu)^2 & (y_i - y_\mu)(z_i - z_\mu) \\ (x_i - x_\mu)(z_i - z_\mu) & (y_i - y_\mu)(z_i - z_\mu) & (z_i - z_\mu)^2 \end{pmatrix} \quad (5.16)$$

wobei  $P_\mu = (x_\mu \ y_\mu \ z_\mu)^T$  den Schwerpunkt der Punktwolke und  $N$  die Anzahl der Punkte in der Punktwolke definiert. Die Singulärwertzerlegung zerlegt die

Matrix  $A$ , sodass gilt:

$$A = USV^T, \quad (5.17)$$

wobei  $U, S, V \in \mathbb{R}^{3 \times 3}$  gilt.  $U$  und  $V$  sind orthogonale Matrizen und  $S$  ist eine Diagonalmatrix, bestehend aus den Singulärwerten  $\delta_1 \geq \delta_2 \geq \delta_3$ . Die Matrix

$$U = \begin{pmatrix} u_1 & u_2 & u_3 \end{pmatrix} \quad (5.18)$$

besteht dabei aus den Spaltenvektoren  $u_1$ ,  $u_2$  und  $u_3$ , die orthogonal zueinander sind. Der Spaltenvektor  $u_3$  entspricht, für den Fall, dass die Punkte  $P_i$  ungefähr in einer Ebene liegen, der Ebenennormale  $n = (n_x \ n_y \ n_z) \in \mathbb{R}^3$  der geschätzten Ausgleichsebene durch die Punktwolke. Mit der Ebenennormale können die Verkippung bzw. der Nickwinkel  $\phi_x$  und der Rollwinkel  $\phi_y$  gegenüber der idealen Bodenebene bestimmt werden:

$$n' = \begin{cases} n, & \text{wenn } n_z \geq 0 \\ -n, & \text{sonst} \end{cases} \quad (5.19)$$

$$v_p = \begin{pmatrix} e_y n' \\ e_z n' \end{pmatrix} = \begin{pmatrix} n_y \\ n_z \end{pmatrix} \Rightarrow \phi'_p = \arccos e \frac{v_p}{\|v_p\|} \quad (5.20)$$

$$v_r = \begin{pmatrix} e_x n' \\ e_z n' \end{pmatrix} = \begin{pmatrix} n_x \\ n_z \end{pmatrix} \Rightarrow \phi'_r = \arccos e \frac{v_r}{\|v_r\|} \quad (5.21)$$

$$\phi_x = \begin{cases} \phi_x, & \text{wenn } n_y \geq 0 \\ -\phi_x, & \text{sonst} \end{cases} \quad (5.22)$$

$$\phi_y = \begin{cases} \phi_y, & \text{wenn } n_x \leq 0 \\ -\phi_y, & \text{sonst} \end{cases} \quad (5.23)$$

wobei  $e_x = (1 \ 0 \ 0)^T$ ,  $e_y = (0 \ 1 \ 0)^T$ ,  $e_z = (0 \ 0 \ 1)^T$  und  $e = (0 \ 1)^T$ .

Die Bestimmung des Gierwinkels  $\phi_z$  ist nicht analog mit diesem Verfahren durchführbar, da die ideale Bodenebene für die Bestimmung von  $\phi_z$  nicht als

Referenzebene verwendbar ist. Die Bestimmung des relativen Gierwinkels eines Sensorkoordinatensystems bezüglich des fahrzeugfesten Koordinatensystems kann nur durch sehr aufwendige Messverfahren erreicht werden, wie sie beispielsweise in der Automobilindustrie zur Ausrichtung der Spur eines Fahrzeugs verwendet werden.



Abbildung 5.8: **Manuelle Bestimmung des Nick- (a) und Rollwinkels (b) des rückwärtigen Laserscanners.**

Aufgrund der geringen Fertigungstoleranzen der mechanischen Werkstatt, welche die Trägerkonstruktion hergestellt hat, wurde daher von einer experimentellen Bestimmung des Gierwinkels der montierten Laserscanner abgesehen und für die weiteren Berechnungen die Annahme  $\phi_z = 0$  zugrunde gelegt. Weiterhin ist zu beachten, dass die vorgestellte Methode nur für aktiv nickende Sensoren durchgeführt werden kann, da diese eine flächige Abtastung bei stehendem Roboter ermöglichen. Bei dem nach hinten messenden Laserscanner wurde daher die Winkel  $\phi_x$  und  $\phi_y$  mit einem mobilen Beschleunigungssensor manuell bestimmt (siehe Abb. 5.8).

Da für die Darstellung im Baum  $T$  eine quaternionbasierte Pose verlangt wird, wird die relative Orientierung der Sensoren  ${}^s q$  bezüglich des fahrzeugfesten Koordinatensystems  $f_v$  nach Gleichung 4.14, basierend auf den zuvor bestimmten Lagewinkeln, bestimmt durch

$${}^s_v q = \begin{pmatrix} \cos \frac{\phi_x}{2} \cos \frac{\phi_y}{2} \\ \sin \frac{\phi_x}{2} \cos \frac{\phi_y}{2} \\ \cos \frac{\phi_x}{2} \sin \frac{\phi_y}{2} \\ -\sin \frac{\phi_x}{2} \sin \frac{\phi_y}{2} \end{pmatrix} \quad (5.24)$$

### Experimentelle Bestimmung der Position des fahrzeugfesten Koordinatensystems

Die Bestimmung der Position des fahrzeugfesten Koordinatensystems  $f_v$  relativ zu den Sensorkoordinatensystemen  $f_{s_i}$  wird ebenfalls experimentell durchgeführt. Der Ursprung von  $f_v$  liegt im Rotationszentrum des Roboters, der nicht zwangsläufig mit dem geometrischen Mittelpunkt des umschreibenden Quaders des Roboters übereinstimmt.

Der Federweg des Fahrwerks des Roboters AMOR erlaubt manuelles Nicken und Rollen des stehenden Roboters. Dies wird genutzt, um mit den Laserscannern des Roboters eine horizontale und eine vertikale Kalibrierungsebene zu vermessen, während der Roboter manuell rotiert wird.

Im Folgenden wird die Bestimmung des Rotationszentrums als Optimierungsproblem aufgefasst. Der Abstand und damit der Fehler zwischen aufgezeichneten Messpunkten und den Kalibrierungsebenen wird dabei abhängig von der zu bestimmenden relativen Position des fahrzeugfesten Koordinatensystems  $f_v$  minimiert. Zur Vereinfachung des Problems wird die Projektion der Messpunkte auf die  $yz$ -Ebene betrachtet. Das Optimierungsproblem für die vertikale Kalibrierungsebene wird beschrieben durch

$$\min_{(x_r, y_r) \in \mathbb{R}^2} \sigma = \sqrt{\frac{\sum_{i=1}^{|C|} \left( y'_i - \frac{\sum_{i=1}^{|C|} y'_i}{|C|} \right)^2}{|C|}} \quad (5.25)$$

$$\begin{pmatrix} x'_i \\ y'_i \\ z'_i \\ 0 \end{pmatrix} = \bar{q}_r \left( \begin{pmatrix} x_i \\ y_i \\ z_i \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ y_r \\ z_r \\ 0 \end{pmatrix} \right) q_r. \quad (5.26)$$

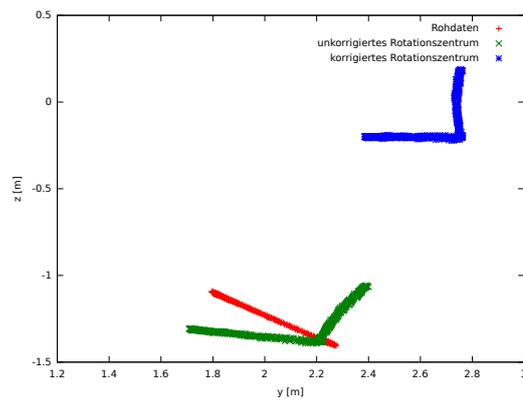
$$(5.27)$$

Dabei ist die Punktwolke  $C = \{P_i = (x_i \ y_i \ z_i)^T\}$  aus den Messpunkten  $P_i$  zusammengesetzt, die während der Vermessung der Kalibrierungsebene aufgenommen wurden.  $p_r = (0 \ y_r \ z_r)^T$  beschreibt die zu bestimmende Position des Sensors relativ zum Rotationszentrum des Roboters und  $q_r$  definiert die Orientierung des Roboters relativ zur initialen Orientierung des Roboters bei Messungsbeginn. Das Ergebnis dieser Methode ist in den Abbildungen 5.9 und 5.10 visualisiert. Damit ist neben der in Kapitel 5.2.4 bestimmten relativen Orientierung  ${}^s_v q$  zwischen  $f_v$  und  $f_s$  auch die Verschiebung  ${}^s_v p := p_r$  bestimmt.

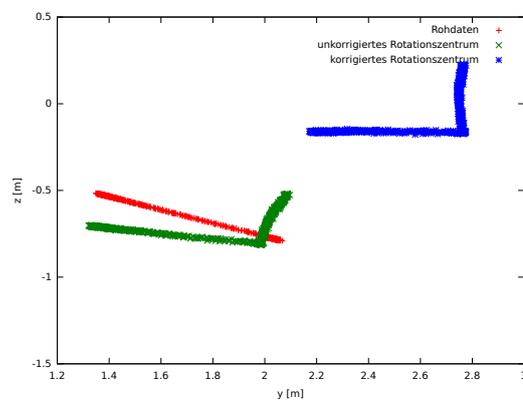
### 5.2.5 Dynamische Transformationen im szenenfesten Koordinatensystem

Bisher wurde der Fall eines unbewegten Roboters betrachtet. Bewegt sich nun der Roboter in der Welt, müssen die Translation und die Rotation des Fahrzeugkoordinatensystems  $f_v$ , bezogen auf das aktuellen Szenenkoordinatensystem  $f_n$ , mit in die Konstruktion einer konsistenten Punktwolke einbezogen werden. Am Rande sei hier erwähnt, dass durch eine reine Translation des Sensorkoordinatensystems durch Bewegung des Roboters auch ohne aktive Rotation der Scan-Ebene bei Nutzung eines planaren Laserscanners eine flächige Abtastung einer Szene realisiert werden kann. Der Roboter Stanley beispielsweise hat mit diesem Ansatz sehr erfolgreich den DARPA Wettbewerb Grand Challenge 2005 bestritten [193]. Der Nachteil dieses Ansatzes ist jedoch, dass zum erneuten Erfassen einer Umweltszene, um beispielsweise sich bewegende von statischen Hindernissen zu unterscheiden oder etwaige Fehlmessungen zu kompensieren, ein Zurücksetzen oder Wenden des Roboters nötig wäre, was in vielen Szenarien nicht praktikabel ist. In dieser Arbeit kommt daher eine Kombination aus aktiv rotierten, planaren Laserscannern und Fahrzeugbewegung zum Einsatz.

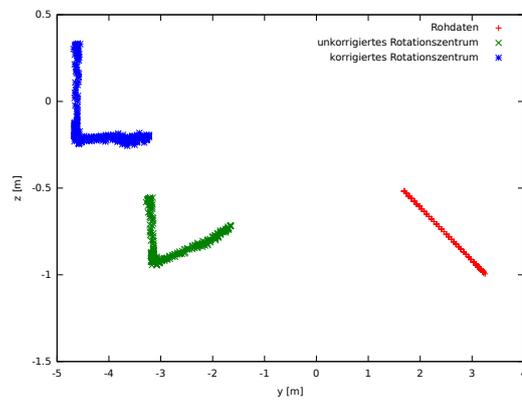
Um die Integration von Sensormessungen auf einem bewegten Roboter zu erreichen, werden alle geometrischen Sensormessungen mit Posen annotiert, die schritthaltend von der Koppelnavigation des Roboters erzeugt werden. Dafür wird die driftende Variante der Koppelnavigation verwendet (siehe Kapitel 4.1). Das Bezugskordinatensystem der Koppelnavigation ist das initiale Szenenkoordinatensystem  $f_k$  mit der zugehörigen relativen Pose  ${}^v_k \check{\rho}$  für die



(a)



(b)



(c)

Abbildung 5.9: **Ergebnisse der experimentellen Bestimmung der Position des Rotationszentrums des Roboters.** Rohdaten ohne Inertialsystem (rot). Einbeziehung des Inertialsystems (grün). Korrigiertes Rotationszentrum (blau). Front-Laserscanner (a und b). Rück-Laserscanner (c).

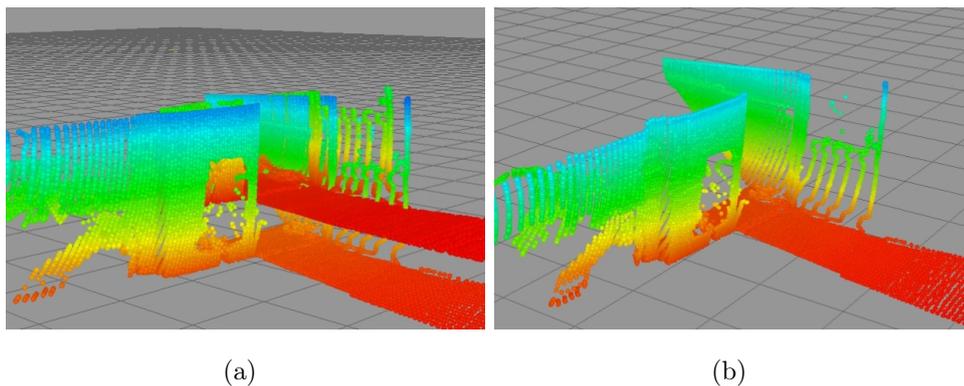


Abbildung 5.10: **Punktwolken zweier Laserscanner vor (a) und nach (b) der Bestimmung der relativen Lage von Sensorkoordinatensystemen und Fahrzeugkoordinatensystem.**

Beschreibung der relativen Fahrzeuglage in  $f_k$ . Die Komponenten  ${}^v_k p$  und  ${}^v_k q$  bestimmen sich nach den in Kapitel 4.1 beschriebenen Gleichungen 4.18 und 4.22.

Jede Scanzeile der Laserscanner wird mit der aktuellen Pose  ${}^v_k \check{\rho}$  annotiert. Um den Fehler im Vergleich zu einer Annotation je Scanpunkt zu ermitteln, wurden die maximalen Winkelgeschwindigkeiten analysiert, die während einer repräsentativen Testfahrt beim Bodenroboter AMOR auftreten. Die Messungen der Laserscanner werden mit einer Frequenz von 75 Hz durchgeführt. Ausgehend von der Annahme, dass bei einem Öffnungswinkel des Scanners von  $180^\circ$  nur die Hälfte des Messzeitintervalls für die tatsächliche Messung verwendet wird, ergibt sich eine Dauer von  $6,6$  ms für den eigentlichen Messvorgang. Damit ergeben sich für die Testfahrt folgende maximale Winkelfehler für einen Messvorgang:

Nickwinkel	Rollwinkel	Gierwinkel
$0,179^\circ$	$0,199^\circ$	$0,339^\circ$

Tabelle 5.1: Maximale Winkelfehler bei Annotation pro Scanzeile.

Weiterhin gilt bei einer maximalen Bewegungsgeschwindigkeit des Roboters von  $1,5 \frac{m}{s}$ , dass der zurückgelegte Weg während der Erzeugung einer Scanzeile bei maximal  $0,01$  m liegt. Zusammengenommen folgt hieraus, dass die Annotation je Scanzeile eine ausreichende Genauigkeit für die Aufgabe der Umgebungsmodellierung für den Bodenroboter AMOR erreicht.

Ziel ist es nun, alle Sensormessungen anstatt bezüglich des zugehörigen Sensorkoordinatensystems  $f_s$  bezüglich des aktuellen Szenenkoordinatensystems  $f_n$  darzustellen. Zunächst wird dafür die von der Koppelnavigation generierte Pose  ${}^v_k\check{\rho}$  in das aktuelle Szenenkoordinatensystem  $f_n$  projiziert, um die Pose des Roboters bezüglich der aktuellen Szene zu erhalten. Die Translations- bzw. Rotationskomponenten  ${}^v_n\check{\rho}$  und  ${}^v_nq$  der projizierten Pose  ${}^v_n\check{\rho}$  werden wie folgt bestimmt:

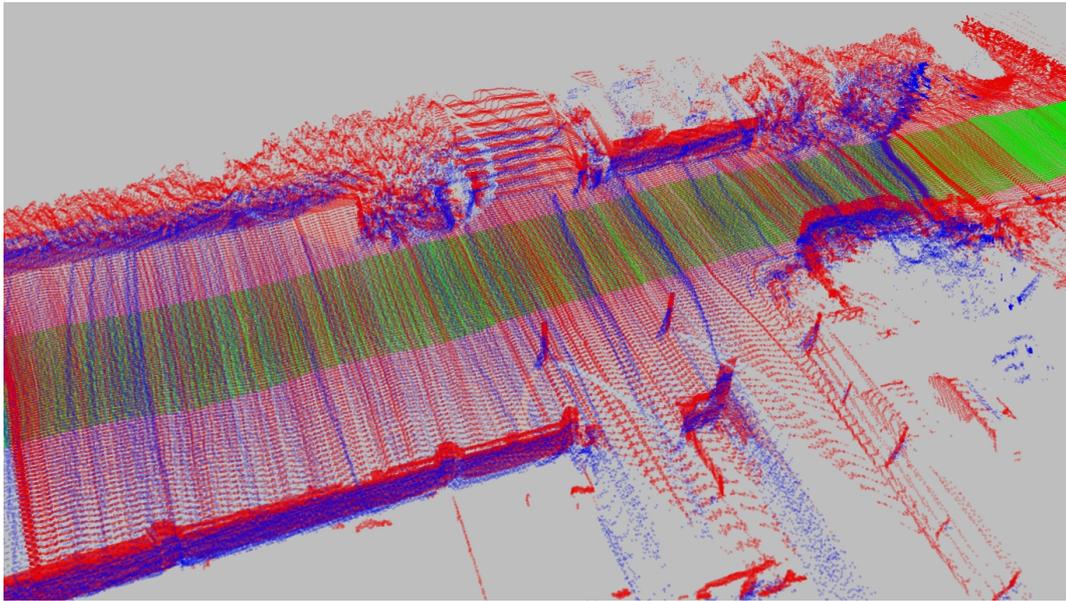
$${}^v_n\check{\rho} = {}^n_k\bar{q} ({}^v_k\check{\rho} - {}^n_k\check{\rho}) {}^n_kq \quad (5.28)$$

$${}^v_nq = {}^n_k\bar{q} {}^v_kq \quad (5.29)$$

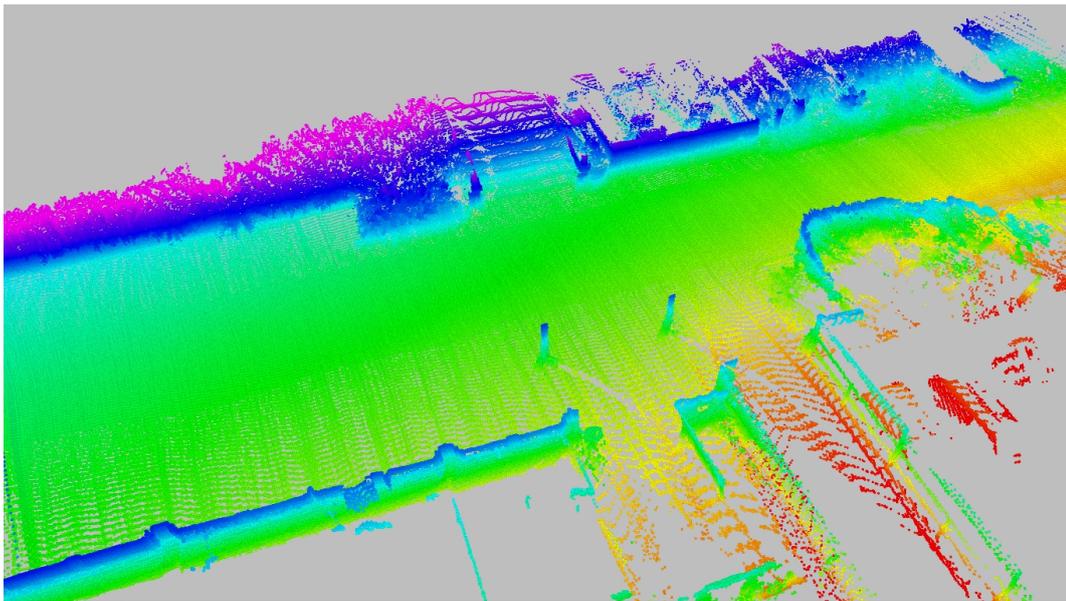
Damit ist die Pose des Roboters bezüglich  $f_n$  bekannt. Mit dieser Pose werden nun alle Messpunktvektoren  $P_i$ , die Element der aufgezeichneten Scanzeile  $l$  sind, nach  $f_n$  projiziert, wobei dafür zunächst noch eine Projektion nach  $f_v$  nötig ist, da die Messungen initial bezüglich  $f_s$  definiert sind (siehe Kap. 5.2.4). Dies entspricht zwei Rückwärtstransformationen, die durch folgenden Ausdruck beschrieben werden:

$$\check{P}'_i = ({}^v_nq (({}^s_vq \check{P}_i {}^s_v\bar{q}) + {}^s_v\check{\rho}) {}^v_n\bar{q}) + {}^v_n\check{\rho} \quad (5.30)$$

Diese generische Transformationsmethode kann für alle Laserscanner gleichermaßen angewandt werden. Damit ist das Ziel erreicht, alle Sensormessungen der Laserscanner in das aktuelle Szenenkoordinatensystem zu überführen und damit ein konsistentes, lokales geometrisches Umgebungsmodell in Form einer Punktwolke für den Roboter zu erzeugen. Abb. 5.11 zeigt ein beispielhaftes Ergebnis der in diesem Kapitel beschriebenen Methoden zur Szenenrekonstruktion auf einem **bewegten Roboter mit drei gleichzeitig eingesetzten Laserscannern**.



(a)



(b)

Abbildung 5.11: **Punktwolke als Ergebnis der Umgebungsmodellierung durch Fusion multipler Laserscanner.** Punktfärbung, abhängig von Scannerzugehörigkeit (a). Punktfärbung in Pseudofarbdarstellung, abhängig von der z-Koordinate des jeweiligen Punkts (b).

## 5.2.6 Variable Szenenkonstruktion aus dem kontinuierlichen Sensordatenstrom

Um variabel auf die Anforderungen der verarbeitenden Module, die das geometrische Umgebungsmodell nutzen, reagieren zu können, wurde eine Methode entwickelt, die es erstens erlaubt, den Zeitpunkt der Modellerzeugung von unterschiedlichen Bedingungen abhängig zu machen, und es zweitens ermöglicht zu beeinflussen, welche Messungen der Szene hinzugefügt werden. Beispielsweise kann gefordert sein, neue Szenen nur dann zu erzeugen, wenn der Roboter sich bewegt, um unnötige Neuerzeugungen und Überlappungen zu vermeiden. Abbildung 5.12 zeigt den Ablauf der variablen Szenenkonstruktion im Überblick.

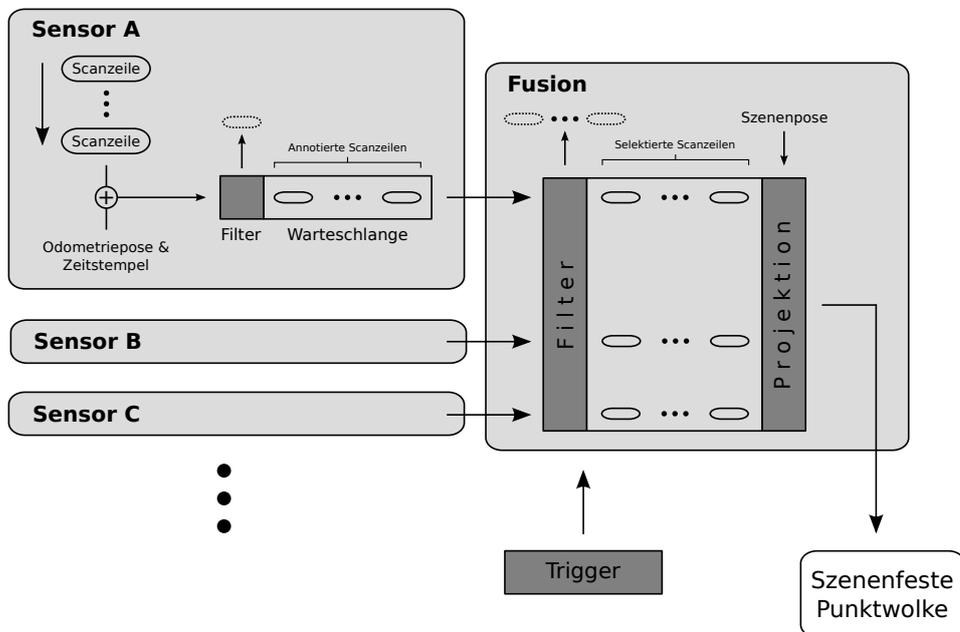


Abbildung 5.12: Ablauf der variablen Szenenkonstruktion aus dem kontinuierlichen Sensordatenstrom.

Jede Scanzeile, die von einem der beteiligten Sensoren erzeugt wird, wird zunächst, wie im vorherigen Kapitel beschrieben, mit der aktuellen Pose des Roboters annotiert. Zusätzlich wird der Aufnahmezeitpunkt mit der Scanzeile abgelegt. Danach wird die Scanzeile einer FIFO<sup>4</sup>-Warteschlange hinzugefügt, wobei vorher durch eine Filterkomponente entschieden wird, ob die Scanzeile

<sup>4</sup>First In First Out

für das Hinzufügen freigegeben oder stattdessen verworfen wird. Dies wird entweder von der vergangenen Zeit oder der zurückgelegten Distanz des Roboters seit der letzten hinzugefügten Messung abhängig gemacht. So wird beispielsweise das Hinzufügen neuer Messungen bei stehendem Roboter kontrolliert oder die Dichte der Messungen verringert, falls eine gröbere Abtastung der Umgebung gewünscht ist. Abhängig von einer Trigger-Komponente wird die eigentliche Konstruktion der szenefesten Punktwolke initiiert, wobei dies wie die Vorselektion der Scanzeilen auch von Zeit oder Distanz abhängig gemacht wird. So kann beispielsweise bei erhöhter Bewegungsgeschwindigkeit des Roboters eine gesteigerte Aktualisierungsrate des Umgebungsmodells realisiert werden.

Ist die Konstruktion einer neuen szenefesten Punktwolke getriggert worden, wird der Inhalt der Warteschlangen aller Sensoren zunächst durch einen weiteren Selektionsfilter verarbeitet. Dabei wird wiederum abhängig von Aufnahmezeitpunkt oder zurückgelegter Distanz entschieden, welche der Messungen aus den Warteschlangen in die szenefeste Punktwolke integriert werden. Das entspricht einem zeitlichen bzw. - bei konstanter Geschwindigkeit - räumlichen Schwellwert für die Integration vergangener Messungen in die zu erzeugende Punktwolke. Die Punkte aller selektierten Scanzeilen werden schließlich mithilfe der zugehörigen Posenannotationen in das durch die feste Szenenpose definierte Koordinatensystem projiziert, wobei die Methoden aus den Kapiteln 5.2.1 und 5.2.4 angewendet werden. Die Szenenpose ist dabei üblicherweise die Pose, die der aktuellen Fahrzeugpose am ähnlichsten ist. Sie könnte jedoch je nach Anwendungsfall auch vollkommen frei gewählt werden.

### 5.2.7 Oberflächenrekonstruktion aus Punktwolken

Das Problem der Oberflächenrekonstruktion aus Punktwolken ist ein intensiv behandeltes Thema in der Forschung. Ursprünglich wurde die Problematik hauptsächlich im Kontext der Computergrafik betrachtet, aber mit der fortschreitenden Verbreitung von Entfernungssensoren ist dieser Bereich auch in den Fokus der Robotikforschung gelangt. Dabei sind verschiedene Ansätze zu unterscheiden. Zunächst gibt es den grundlegenden Unterschied, ob entweder eine geordnete oder eine ungeordnete Punktwolke als Eingangsdatentyp zur

Verfügung steht (vgl. dazu auch Kapitel 5.2.2). Die Oberflächenrekonstruktion aus geordneten Punktwolken ist im Vergleich zur ungeordneten Variante deutlich weniger komplex. Allerdings kann dieser Ansatz bei der Nutzung von Zeilen-Laserscannern nur unter gewissen Umständen eingesetzt werden und ist damit auf gewisse Anwendungsszenarien beschränkt. Die Methoden zur Oberflächenrekonstruktion aus ungeordneten Punktwolken hingegen sind eher generischer Natur und lassen sich auf beliebige Punktwolken anwenden. In den folgenden Abschnitten werden die Details beider Varianten erörtert.

### Triangulation geordneter Punktwolken

Wie in Kapitel 5.2.2 bereits beschrieben wurde, zeichnen sich geordnete Punktwolken durch die Struktur der enthaltenen Punkte aus, die implizit das Nachbarschaftsverhältnis der Punkte zueinander definiert. Mit dieser Information kann eine Methode zur Oberflächenrekonstruktion aus der Punktwolke realisiert werden, die benachbarte Punkte zu Dreiecken verbindet und damit auf direktem Wege ein Dreiecksnetz erzeugt, das die von der Punktwolke beschriebene Oberfläche annähert [104]. Wie Abb. 5.13 zeigt, entstehen aus je vier benachbarten Punkten zwei Dreiecke.

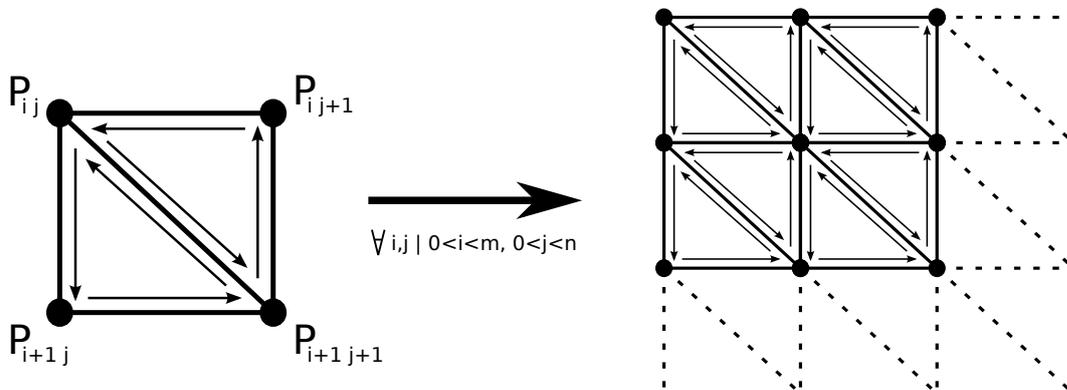


Abbildung 5.13: **Triangulationsschema für geordnete Punktwolke.**

Die Punkte  $P_{ij}$  mit  $0 < i < m \mid m \in \mathbb{N}$  und  $0 < j < n \mid n \in \mathbb{N}$  entstammen dabei der Matrixdarstellung  $M_C$  der geordneten Punktwolke  $C$  (vgl. Kap. 5.2.2), in der die sequenziell aufgenommenen Scanzeilen als untereinander angeordnete

te Zeilenvektoren enthalten sind.  $m$  definiert die Anzahl der aufgenommenen Scanzeilen und  $n$  die Anzahl der Elemente je Scanzeile.

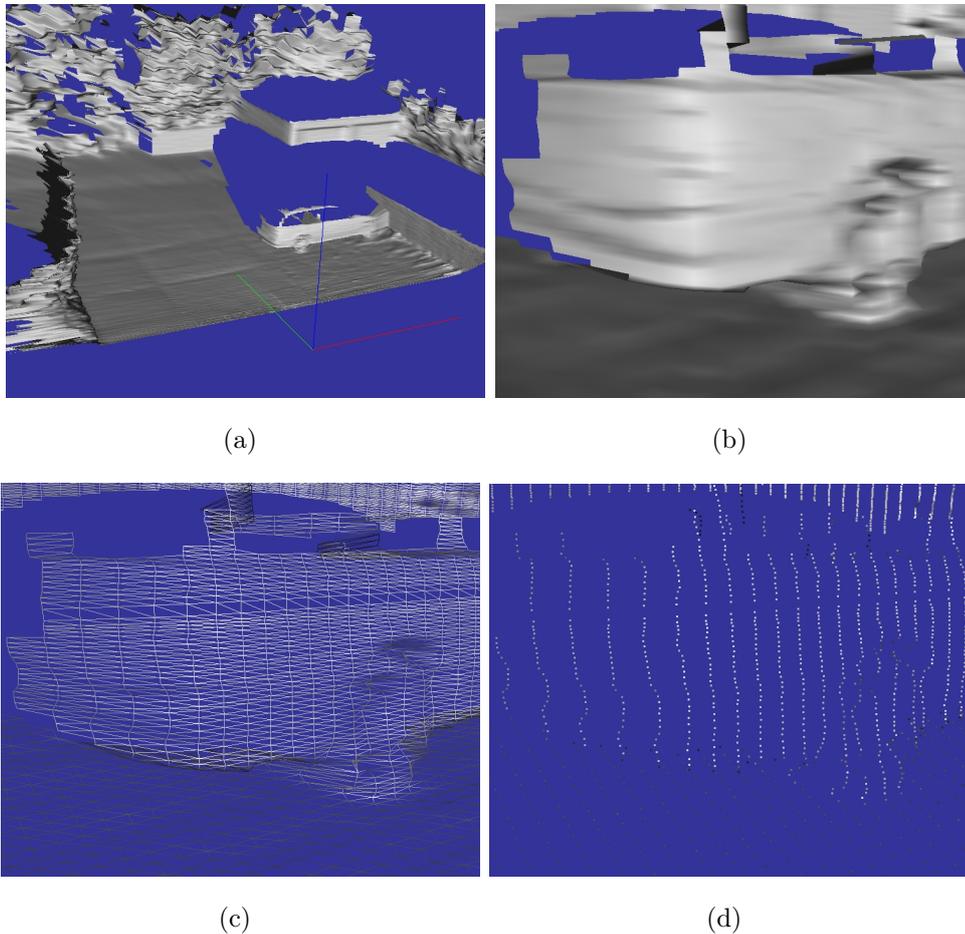


Abbildung 5.14: **Beispiel der Triangulation einer geordneten Punktwolke.** Aus einer geordneten Punktwolke generiertes Dreiecksnetz (a). Detailansichten der triangulierten Oberfläche: Beleuchtetes Dreiecksnetz (b), Gitternetz (c) und Punktwolke (d).

Diese Methode funktioniert allerdings nur zuverlässig, solange sich die korrekte Reihenfolge der Scanzeilen direkt und zuverlässig aus dem Aufzeichnungsprozess ergibt. Sei es durch eine definierte, aktive Rotationsbewegung des Sensors oder durch eine kontinuierliche Bewegung des Roboters, ohne zu große Rotationsbewegungen der Plattform, auf welcher der Sensor montiert ist. Zusätzlich ist bei der Verwendung multipler Sensoren, die einen überlappenden Öffnungswinkel haben, ebenfalls oft keine eindeutige Reihenfolge der aufgezeichneten Scanzeilen festzustellen.

## Triangulation ungeordneter Punktwolken

Die Algorithmen zur Triangulation von ungeordneten Punktwolken können nach [137] in verschiedene Kategorien unterteilt werden: Räumliche Unterteilung, Oberflächenverformung und inkrementelles Oberflächenwachstum. Viele Algorithmen können jedoch nicht ganz klar ausschließlich einer Kategorie zugeordnet werden, da sie eine Mischung dieser Ansätze nutzen, wie in den folgenden Beispielen dargestellt wird.

Algorithmen zur räumlichen Unterteilung dienen im dreidimensionalen Raum der Aufteilung eines umschreibenden Quaders einer Punktwolke in disjunkte Zellen. Dabei werden die Elemente der Punktwolke den Zellen zugeordnet, in denen sie enthalten sind. Die daraus entstehenden Datenstrukturen können genutzt werden, um eine Oberflächenrekonstruktion der Punktwolke durchzuführen. Dabei werden oberflächenbasierte und volumenbasierte Ansätze unterschieden.

Oberflächenbasierte Algorithmen zerlegen zunächst den Raum in Zellen und suchen danach die Zellen, die von der zu rekonstruierenden Oberfläche geschnitten werden, um aus diesen Zellen die Oberfläche zu erzeugen. Dabei können verschiedene Strategien für die Suche der Oberflächenzellen eingesetzt werden.

Hoppe et al. [82] nutzen eine vorzeichenbehaftete Abstandsfunktion für die Auswahl der Oberflächenzellen, die im Fall dieser Methode einem gleichmäßigen dreidimensionalen Gitter entstammen. Die Funktion bestimmt den minimalen Abstand jedes Punktes zur geschlossenen Oberfläche des zu rekonstruierenden Objekts. Positive Abstände beschreiben Punkte, die außerhalb des Objekts liegen. Negative Abstände beschreiben Punkte, die innerhalb des Objekts liegen. Folglich sind die Zellen als Oberflächenzellen zu wählen, die Punkte beinhalten, die Abstände mit gegensätzlichen Vorzeichen haben. Die Konstruktion der Oberfläche wird mit dem *marching cubes*-Algorithmus erreicht [129].

Auch Bajaj et al. [12] verwenden eine vorzeichenbehaftete Abstandsfunktion, die mithilfe einer approximierten Oberfläche definiert ist, die auf der Basis von  $\alpha$ -Shapes aus der Punktwolke erzeugt wird. Die Abstandsfunktion bestimmt im Folgenden den Ablauf des inkrementellen räumlichen Aufteilungs-

prozesses in disjunkte Tetraeder. Genau wie in [82] werden Oberflächenzellen durch gegensätzliche Vorzeichen der Abstandsfunktionen identifiziert. Für jeden Oberflächen-Tetraeder wird eine lokale Näherung der den Tetraeder schneidenden Oberfläche auf Basis von Bernstein-Bézier-Flächen durchgeführt. Ist der Fehler zwischen lokaler Approximation und den lokalen Objektpunkten zu groß, wird der Tetraeder weiter unterteilt. Dies wird solange fortgeführt, bis eine ausreichend genaue Näherung der Oberfläche erreicht ist. Die resultierende Oberfläche wird schließlich mit Bernstein-Bézier-Teilflächen beschrieben.

Edelsbrunner und Mücke [54] wählen eine Delaunay-Zerlegung [157] in Tetraeder als räumliche Aufteilungsmethode. Aus der entstehenden Datenstruktur werden sukzessive Tetraeder, Dreiecke und Kanten entfernt. Dabei werden alle Tetraeder, Dreiecke und Kanten, deren minimale umschließende Kugeln nicht in eine sogenannte  $\alpha$ -Kugel passen, eliminiert. Anschließend werden Dreiecke zur Rekonstruktion der Oberfläche aus der reduzierten Datenstruktur extrahiert, die der Regel genügen, dass kein anderer Punkt der Punktwolke sich in einer der beiden Kugeln mit Radius  $\alpha$  befindet, die sich aus den drei Eckpunkten des Dreiecks konstruieren lassen.

Volumenbasierte Algorithmen zerlegen zunächst genauso wie die oberflächenbasierten Algorithmen den Raum in disjunkte Zellen. Danach jedoch werden die Zellen entfernt, die nicht zu dem Volumen gehört, das von der Oberfläche eingegrenzt wird, die durch die Punktwolke beschrieben wird. Danach wird die Oberfläche aus den gewählten Zellen rekonstruiert.

Boissonnat führt in [23] ebenfalls zu Beginn eine Zerlegung des Raums mittels Delaunay-Triangulation durch. Abhängig von der Anzahl der Seiten, Kanten und Punkte der durch die Triangulation erzeugten Tetraeder werden gewisse Tetraeder eliminiert. Alle Tetraeder, die entweder zwei Seiten, fünf Kanten und vier Punkte oder eine Seite, drei Kanten und drei Punkte haben, werden entfernt. Diese Eliminationsregel bedingt jedoch, dass nur Objekte bzw. Oberflächen ohne Löcher rekonstruiert werden können. Zur sukzessiven Auswahl des nächsten zu eliminierenden Tetraeders wird eine Bewertungsfunktion definiert, welche die Tetraeder sortiert. Der Tetraeder mit der jeweils höchsten Bewertung wird als Erstes entfernt. Die Bewertungsfunktion beschreibt die maximale Distanz der Flächen eines betrachteten Tetraeders zu dessen Umkugel. Die Rekonstruktion ist beendet, wenn alle Punkte auf der zu erzeugenden

Oberfläche liegen oder wenn die Entfernung des Tetraeders mit der höchsten Bewertung die Summe der Bewertungen der Tetraeder nicht mehr erhöht, die direkt an den Rekonstruktionspolyeder angrenzen.

Veltkamp [201, 202] führen den Begriff des  $\gamma$ -Indikators ein. Der Indikator entspricht einer Bewertungsfunktion, die einen Wert für jeweils drei Randpunkte des die Oberfläche annähernden Polyeders bestimmt. Der absolute Wert der Funktion bestimmt sich durch  $1 - \frac{r}{R}$ .  $r$  ist dabei der Radius des Kreises, der durch die drei Randpunkte verläuft, und  $R$  ist der Radius der Umkugel des Tetraeders, dessen eine Randfläche aus den drei Randpunkte gebildet wird. Das Vorzeichen des  $\gamma$ -Indikators ist negativ, falls der Mittelpunkt der Umkugel auf der Innenseite des Polyeders liegt, und positiv, falls er außerhalb liegt. Der  $\gamma$ -Indikator ist skalierungsinvariant und damit nicht abhängig von der Größe des betrachteten Tetraeders bzw. Dreiecks. Mit der Definition des  $\gamma$ -Indikators ist die Vorgehensweise analog zu der in [23]. Nach einer Delaunay-Triangulation wird sukzessive derjenige Tetraeder entfernt, der den größten  $\gamma$ -Indikatorwert besitzt. Der Unterschied zu [23] besteht in der Tatsache, dass unterschiedliche lokale Punktedichten behandelt werden können, wobei trotzdem nur Objekte ohne Löcher rekonstruiert werden können.

Oberflächenverformungsbasierte Algorithmen gehen von einer initialen Oberfläche aus, die solange angepasst und verformt wird, bis sie eine adäquate Annäherung der zu approximierenden Punktwolke darstellt. Diese Vorgehensweise ist gerade dann anzuraten, wenn bereits eine grobe Näherung der zu rekonstruierenden Oberfläche bekannt ist.

Terzopoulos und Metaxas stellen in [190, 189] eine physikalische Herangehensweise auf der Basis verformbarer Superquadriken vor, welche die Vorteile der globalen Oberflächenbeschreibung eines Superellipsoids mit den zusätzlichen lokalen Freiheitsgraden eines Polynomzugs verbindet. Ein Ausgangsmodell wird an die zu beschreibende Punktwolke angepasst, indem die Eingangsdaten als physikalische Kräfte beschrieben und die Parameter des Modells anhand eines klassischen Bewegungsmodells durch eine Simulation bestimmt werden.

Einen zu [189] konträren Ansatz wählen Szeliski und Tonnesen in [187]. Es wird kein geschlossen beschriebenes Oberflächenmodell angepasst, sondern die Oberfläche wird vielmehr durch eine endliche Anzahl an orientierten Partikeln

beschrieben. Die Partikel werden durch Anziehung und Abstoßung während des Rekonstruktionsprozesses beeinflusst, wie dies beispielsweise aus physikalischen Simulationen von Flüssigkeiten bekannt ist. Die Partikelbewegung wird durch verschiedene Potenzialfunktionen geleitet, die dafür sorgen, dass bevorzugt planare und kugelförmige lokale Anordnungen von Partikeln entstehen. Weiterhin wird die Anzahl der Partikel angepasst, falls eine Erweiterung oder Reduktion der Oberfläche nötig sein sollte.

Algorithmen, die ein inkrementelles Oberflächenwachstum zur Oberflächenrekonstruktion umsetzen, konstruieren die Oberfläche ausgehend von einem initialen Punkt durch sequenzielles Hinzufügen von Punkten oder Dreiecken, bis alle Punkte der Punktwolke zur Oberfläche hinzugefügt wurden.

Der Algorithmus von Mencl und Müller [138] zur inkrementellen Oberflächenrekonstruktion setzt sich aus sieben Schritten zusammen. Zunächst wird ein minimaler Spannbaum aus der Eingangspunktwolke aufgebaut, der alle Punkte der Wolke enthält. Danach werden die Blätter des Baums, abhängig von verschiedenen geometrischen Bedingungen, auf Verknüpfbarkeit untersucht. So wird ein erweiterter Graph erzeugt, der sogenannte Oberflächenbeschreibungsgraph zweiter Ordnung. Im erweiterten Graphen werden im nächsten Schritt Merkmale gesucht, welche die Rekonstruktion erleichtern sollen. Merkmale können entweder Pfade oder Ringe sein. Mithilfe der detektierten Merkmale werden Kanten aus dem Graphen entfernt, die kontraproduktiv für die spätere Rekonstruktion sind. Beispielsweise werden Kanten entfernt, die zwei Ringe verbinden, falls die zu entfernende Kante nahezu senkrecht zu den Normalen beider Ringe ist und die Ringe weit genug voneinander entfernt sind. Danach werden ähnliche Merkmale vereint und Kanten mit ähnlichen Eigenschaften verbunden. Im letzten Schritt wird der in den vorherigen Schritten entstandene Oberflächenbeschreibungsgraph inkrementell mit Dreiecken gefüllt, um die Oberflächen in Form eines Dreiecksnetzes zu rekonstruieren.

Abschließend wird nun die inkrementelle Rekonstruktionsmethode von Marton et al. [135] vorgestellt, die im Rahmen dieser Arbeit zur Oberflächenrekonstruktion von ungeordneten Punktwolken eingesetzt wird. Der Algorithmus gehört zur Kategorie der Greedy-Algorithmen und kommt ohne die Konstruktion einer Hilfsdatenstruktur aus (im Gegensatz zu [138]). Damit ist die Methode sehr rechenzeiteffizient.

Die Grundlage für diese Methode bildet die Arbeit von Gopi und Krishnan [66]. Der Beitrag von Marton et al. besteht in der Reduktion des Rechenaufwands und der Optimierung auf 2,5D-Messdaten variabler lokaler Punktedichten. Somit ist das Verfahren ideal geeignet für die Echtzeitrekonstruktion von Oberflächen aus Laserscanner-erzeugten Messdaten in dieser Arbeit.

Die Triangulation passiert inkrementell. Dabei wiederholen sich die folgenden Schritte während des Rekonstruktionsprozesses: Für den aktuell betrachteten Punkt  $P$  wird die  $k$ -Nachbarschaft bestimmt, indem in einer Kugel mit Radius  $r = c d_0$  die  $k$  nächsten Nachbarn von  $P$  gesucht werden. Da  $d_0$  dabei den Abstand von  $P$  zu seinem nächsten Nachbarn definiert, passt sich der Algorithmus automatisch an unterschiedliche lokale Punktdichten an.  $c$  ist eine frei wählbare Konstante. Die Punkte in der  $k$ -Nachbarschaft werden danach entlang der geschätzten lokalen Oberflächennormale auf eine Ebene projiziert. Die Ebene ist eine Ausgleichsebene, die aus allen Punkten der  $k$ -Nachbarschaft ermittelt wurde. Im dritten Schritt werden die projizierten Punkte abhängig von einem Sichtbarkeits- und einem Winkelkriterium mit dem aktuellen Punkt  $P$  verbunden. Durch das inkrementelle Hinzufügen von Kanten zwischen Messpunkten entstehen so Dreiecke, welche die gesuchte Oberfläche approximieren.

Implementierungsseitig wird die Suche der  $k$  nächsten Nachbarn durch die Verwendung eines kd-Baums beschleunigt [8]. Die Methode zur Bestimmung der lokalen Ausgleichsebene, basierend auf der Singulärwertzerlegung der Kovarianzmatrix der Punkte in der lokalen Nachbarschaft des betrachteten Punktes, ist sehr ähnlich zum Ansatz in Kapitel 5.2.4 dieser Arbeit. Der Unterschied liegt lediglich in der Gewichtung der Summanden, aus denen sich die Kovarianzmatrix zusammensetzt, mit dem Faktor  $\xi_i = e^{-\frac{d_i^2}{\mu}}$ , wobei  $d_i$  den Abstand des Punktes  $P$  zu seinem Nachbarpunkt  $P_i$  und  $\mu$  den durchschnittlichen Abstand von  $P$  zu allen Punkten in der lokalen  $k$ -Nachbarschaft von  $P$  definiert. Dieser Ansatz liefert im Gegensatz zu anderen Ansätzen (wie beispielsweise in [66]) auch bei verrauschten Daten eine robuste Schätzung der Projektionsebene. Zur Verdeutlichung der Vorgehensweise beim inkrementellen Hinzufügen von Kanten zeigt Abb. 5.15 exemplarisch die lokale Nachbarschaft eines Punktes  $R$  vor und nach dem Hinzufügen neuer Kanten.

Für die Triangulation werden nur Punkte betrachtet, die in einer Kugel  $S_R$  um den aktuellen Referenzpunkt  $R$  liegen. Die Punkte, die innerhalb von  $S_R$

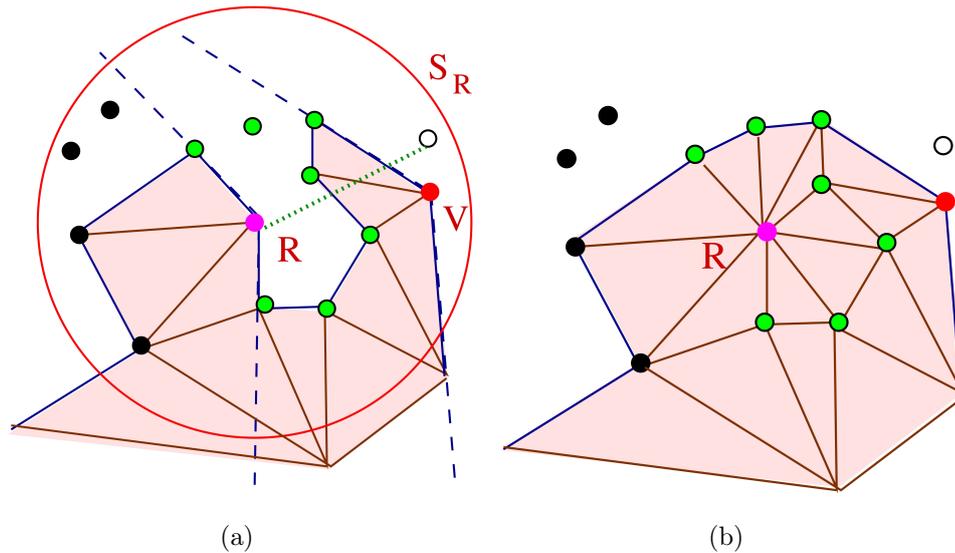


Abbildung 5.15: **Greedy-Triangulationsmethode** nach [66] und [135]. Abbildungen aus [66].

liegen, werden als Kandidaten bezeichnet und sind in der Punktmenge  $C_R$  zusammengefasst. Die Punkte in  $C_R$  werden abhängig vom Abstrahlwinkel der Kanten, die  $R$  und den jeweils betrachteten Punkt aus  $C_R$  verbinden, sortiert. Die Sortierung der Punkte um  $R$  wird genutzt, um Dreiecke zu erzeugen, die keine gegenseitigen Überlappungen haben. Die grünen Punkte in Abbildung 5.15 sind die Punkte, die in sortierter Reihenfolge durch Verknüpfung mit  $R$  Dreiecke kreieren. Dies kann jedoch nicht ausnahmslos mit allen Punkten in  $C_R$  durchgeführt werden. Punkte, die hinter den Grenzkanten (blau gestrichelte Linien) von  $R$  liegen, dürfen nicht mit  $R$  verknüpft werden (schwarze Punkte). Das Gleiche gilt für Punkte, die von  $R$  aus betrachtet durch bereits existierende Kanten verdeckt werden (weißer Punkt). Schließlich dürfen auch keine Punkte mit  $R$  verknüpft werden, hinter deren Grenzkanten  $R$  selber liegt (roter Punkt  $V$ ). Abbildung 5.15 (b) zeigt das Ergebnis der Verknüpfung aller validen Punkte aus  $C_R$ .

Das Sichtbarkeitskriterium, das im vorherigen Abschnitt beschrieben wurde, ist ausreichend, um eine gültige Triangulation der Punktmenge zu erreichen, führt allerdings in einigen Fällen zur Erzeugung sehr schmaler Dreiecke, die den visuellen Eindruck der rekonstruierten Oberfläche trüben können. Daher kann optional dazu noch ein Minimalwinkelkriterium benutzt werden, um durch Entfernung von einzelnen Punkten in  $C_R$  die Erzeugung von schmalen Dreie-

cken zu unterbinden.

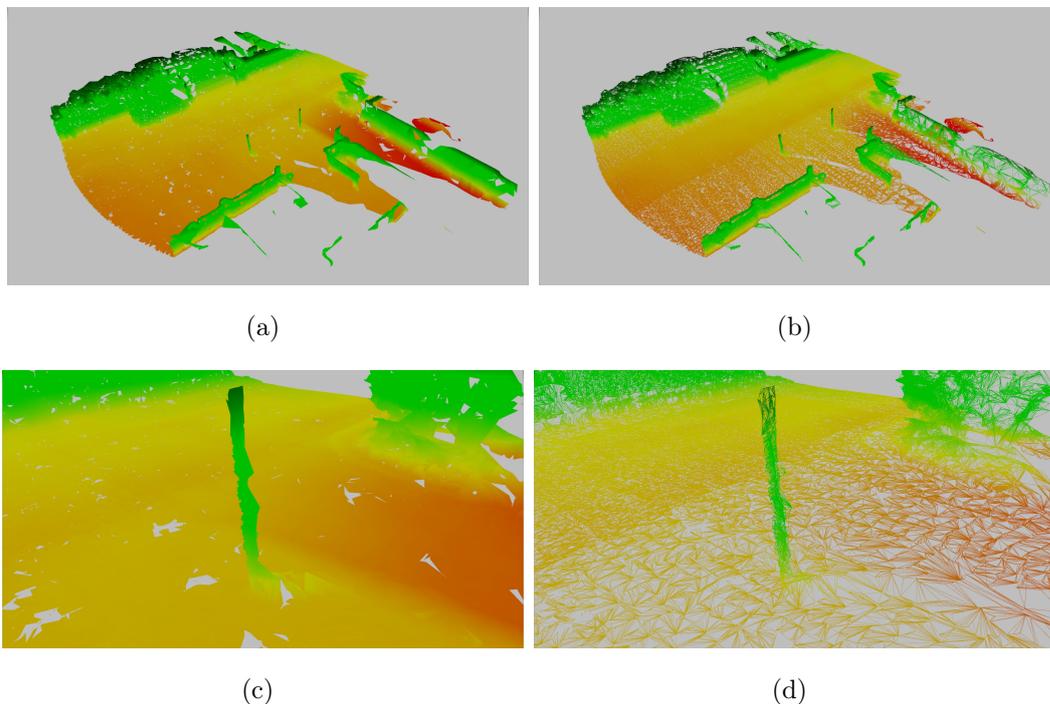


Abbildung 5.16: **Ergebnis der Triangulation einer ungeordneten Punktwolke**, die durch Fusion der Messungen mehrerer Laserscanner entstanden ist. Gesamte Szene als Dreiecksnetz (a) und als Gitternetz (b). Detailansicht der Gesamtszene als Dreiecksnetz (c) und als Gitternetz (d).

Abbildung 5.16 zeigt abschließend die Ergebnisse der Triangulation fusionierter Punktwolken aus mehreren unabhängig messenden Laserscannern anhand einer Beispielszene, die während einer Testfahrt mit dem Roboter AMOR entstanden ist.

### 5.3 Visuelle Beschreibung der Umgebung

Zur Erweiterung bzw. Anreicherung des bisher erzeugten rein geometrischen Modells werden Daten visueller Sensoren verwendet. So entsteht erstens eine wesentlich intuitivere und menschlichen Benutzern und Entwicklern zugänglichere Darstellungsform und zweitens werden Probleme lösbar, die auf Basis einer rein geometrischen Beschreibung entweder schwerer oder gar nicht

lösbar wären. Ein Beispiel in diesem Zusammenhang ist die Unterscheidung von ebenen Teilflächen, die unterschiedlichen Untergrundtypen (Straße, Waldweg, Morast, Wasserpfützen etc.) entsprechen. Dies ist bei einem weitgehend flachen geometrischen Modell nicht ohne Weiteres möglich. Die Hinzunahme registrierter visueller Informationen ermöglicht jedoch aufgrund eines jeweils unterschiedlichen visuellen Erscheinungsbildes eine Unterscheidung in diesem Kontext, was in vielen Situationen zu einem deutlich effizienteren Planungsverhalten des Roboters führen kann (siehe Kapitel 6).

### 5.3.1 Sensorik

Ebenso wie zur geometrischen Erfassung der Umgebung stehen auch zur visuellen Erfassung verschiedene Sensortypen zur Verfügung. In dieser Arbeit wurden auf Seiten der visuellen Sensorik ausschließlich Farbbildkameras verwendet. Allerdings wurden die folgenden drei unterschiedlichen Einsatzarten von Farbbildkameras evaluiert: Eine Farbbildkamera mit entozentrischem Objektiv (Abb. 5.17 (a)), eine Farbbildkamera mit katadioptrischem Objektiv (Abb. 5.17 (b)) und eine Farbbildkamera, die auf einem Flugroboter montiert ist (Abb. 5.17 (c)).

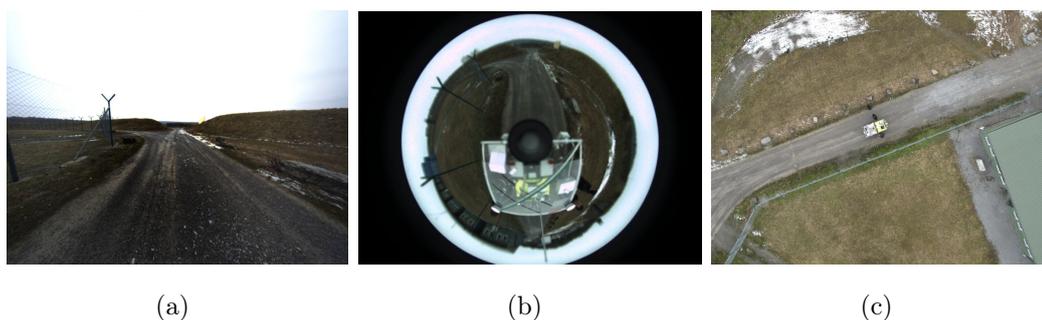


Abbildung 5.17: **Beispiele unterschiedlicher visueller Beschreibungen der gleichen Szene:** Entozentrisches Objektiv (a). Katadioptrisches Objektiv (b). Luftbildkamera (c).

### 5.3.2 Intrinsische Kalibrierung

Die Hauptaufgabe der visuellen Sensorik im Rahmen dieser Arbeit ist die Anreicherung des zuvor erzeugten geometrischen Modells mit visuellen Informationen. Daher ist die präzise Kalibrierung der eingesetzten Kameras von essenzieller Bedeutung. Sie teilt sich in die Bestimmung der intrinsischen und extrinsischen Parameter auf.

Die intrinsischen Parameter definieren die interne Geometrie von Kamera und Objektiv. Die extrinsischen Parameter beschreiben die Lage der Kamera bezüglich des Weltkoordinatensystems. Zusammengenommen parametrisieren intrinsische und extrinsische Parameter die Abbildungsvorschrift von Objekt- auf Bildpunkte, welche die Grundlage für die Fusion von geometrischen und visuellen Sensorinformationen bildet.

Zunächst wird in diesem Abschnitt die intrinsische Kalibrierung der visuellen Sensorik besprochen. Als Teilschritt der intrinsischen Kalibrierung wird dabei im Rahmen der vorgestellten Verfahren eine extrinsische Kalibrierung durchgeführt. Die extrinsische Kalibrierung ist jedoch in diesem Zusammenhang ausschließlich als reiner Vorverarbeitungsschritt zu sehen und nicht mit der eigentlich gesuchten extrinsischen Kalibrierung zu verwechseln, welche die räumliche Beziehung zwischen geometrischem und visuellem Sensor herstellt. Da unterschiedliche Objektivtypen unterschiedlicher Kalibrierungsmethoden bedürfen, werden im Folgenden Methoden für die Kalibrierung von Kameras mit entozentrischem und katadioptrischem Objektiv unterschieden.

#### Entozentrische Objektive

Zu den intrinsischen Parametern zählen zunächst die Brennweite  $f = ( f_x \ f_y )$  und der Bildhauptpunkt  $c = ( c_x \ c_y )$  der Kamera. Daneben sind die radialen und tangentialen Verzerrungsparameter  $\kappa = ( \kappa_1 \ \kappa_2 \ \kappa_3 \ \kappa_4 \ \kappa_5 )$  zu bestimmen. Sie parametrisieren nichtlineare Korrekturfunktionen, die systematische Fehler korrigieren, die aus fehlerhafter Linsen- oder Kamerageometrie entstehen, wobei der radiale Anteil der Verzerrung meist deutlich stärker ausgeprägt ist [198]. Die extrinsischen Parameter setzen sich aus einer  $3 \times 3$ -Rotationsmatrix  $R$  und einem dreidimensionalen Translationsvektor  $t$  zusammen.

Das folgende Kalibrierungsverfahren basiert hauptsächlich auf der Arbeit von Zhang [215] und wird erweitert durch das von Heikkila und Silvén [78] beschriebene Kameramodell, das im Gegensatz zu der Arbeit von Zhang neben der radialen auch die tangentielle Verzerrung berücksichtigt.

Ausgangspunkt für die experimentelle Bestimmung der Kalibrierungsparameter ist das Lochkameramodell. Es beschreibt eine einfache Form der optischen Abbildung, bei der die abzubildende Umwelt nur durch eine Lochblende von der Bildebene getrennt wird. Damit entspricht die Abbildung einer punktsymmetrischen Spiegelung der Szene am Durchstoßpunkt der zentralen optischen Achse mit der Lochblende ebene. Mit diesem Modell wird die Abbildung eines Punkts  $P = (x \ y \ z) \in \mathbb{R}^3$  im 3D-Weltkoordinatensystem auf den Bildpunkt  $P' = (u \ v)^T \in \mathbb{R}^2$  im 2D-Bildkoordinatensystem durch

$$s \begin{pmatrix} P' \\ 1 \end{pmatrix} = K \begin{pmatrix} R & t \end{pmatrix} \begin{pmatrix} P \\ 1 \end{pmatrix} \quad \text{mit } K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (5.31)$$

beschrieben. Für die experimentelle Bestimmung der Kalibrierungsparameter bei bekannter Abbildungsvorschrift 5.31 von Weltpunkten auf Bildpunkte ist es nötig, korrespondierende Welt-Bild-Punktpaare zu ermitteln. Dies wird durch die Nutzung eines Kalibrierobjekts mit bekannten Ausmaßen erreicht, beispielsweise einem planaren Objekt, auf das ein Schachbrettmuster aufgedruckt ist. Die inneren Ecken des Schachbretts können automatisch mit Methoden der digitalen Bildverarbeitung detektiert werden. Um eine möglichst robuste Lösung zu erreichen, ist zu empfehlen, mehrere Aufnahmen vom Kalibrierobjekt zu machen.

Das Weltkoordinatensystem wird so gewählt, dass für alle detektierten Punkte auf dem planaren Kalibrierobjekt  $z = 0$  gilt. Dies vereinfacht die Abbildung zu

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K \begin{pmatrix} r_1 & r_2 & t \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (5.32)$$

wobei  $r_i$  den  $i$ -ten Spaltenvektor der Matrix  $R$  bezeichnet. Der Zusammenhang

zwischen Bild- und Kalibrierobjektpunkt wird demnach durch die Homographie

$$H = \begin{pmatrix} h_1 & h_2 & h_3 \end{pmatrix} = K \begin{pmatrix} r_1 & r_2 & t \end{pmatrix} \quad (5.33)$$

beschrieben. Da  $r_1$  und  $r_2$  orthonormal zueinander sind, gilt:

$$h_1^T (K^{-1})^T K^{-1} h_2 = 0 \quad (5.34)$$

$$h_1^T (K^{-1})^T K^{-1} h_1 = h_2^T (K^{-1})^T K^{-1} h_2. \quad (5.35)$$

Sei Matrix  $B$  definiert als

$$B = \begin{pmatrix} B_{11} & B_{21} & B_{31} \\ B_{12} & B_{22} & B_{32} \\ B_{13} & B_{23} & B_{33} \end{pmatrix} = (K^{-1})^T K^{-1} = \begin{pmatrix} \frac{1}{f_x^2} & 0 & -\frac{c_x}{f_x^2} \\ 0 & \frac{1}{f_y^2} & -\frac{c_y}{f_y^2} \\ -\frac{c_x}{f_x^2} & -\frac{c_y}{f_y^2} & \frac{c_x^2}{f_x^2} + \frac{c_y^2}{f_y^2} + 1 \end{pmatrix}. \quad (5.36)$$

Mit der Vektordarstellung  $b$  der symmetrischen Matrix  $B$  und dem  $i$ -ten Spaltenvektor  $h_i$  der Homographiematrix  $H$  gilt folgender Zusammenhang

$$h_i^T B h_j = v_{ij}^T b \quad \text{mit} \quad v_{ij} = \begin{pmatrix} h_{i1} h_{j1} \\ h_{i1} h_{j2} + h_{i2} h_{j1} \\ h_{i2} h_{j2} \\ h_{i3} h_{j1} + h_{i1} h_{j3} \\ h_{i3} h_{j2} + h_{i2} h_{j3} \\ h_{i3} h_{j3} \end{pmatrix}. \quad (5.37)$$

Die Bedingungen 5.34 und 5.35 können mit Ausdruck 5.37 als

$$v_k' = \begin{pmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{pmatrix} = 0 \quad (5.38)$$

dargestellt werden. Mit  $n$  Aufnahmen des Kalibrierobjekts ergibt sich ein Gleichungssystem

$$Vb = \begin{pmatrix} v'_1 \\ \vdots \\ v'_n \end{pmatrix} b = 0, \quad (5.39)$$

wobei  $V$  eine  $2n \times 6$ -Matrix ist. Die Lösung des Gleichungssystems entspricht dem Eigenvektor von  $V^T V$ , des zugehörigen kleinsten Eigenwerts. Nach [215] kann aus der Lösung  $b$  direkt die Kameramatrix  $K$  bestimmt werden. Ausgehend von  $B = \lambda (K^{-1})^T K^{-1}$  mit beliebigem Skalierungsfaktor  $\lambda$  können die Elemente von  $K$  durch:

$$f_x = \sqrt{\frac{\lambda}{B_{11}}} \quad (5.40)$$

$$f_y = \sqrt{\frac{\lambda B_{11}}{B_{11} B_{22} - B_{12}^2}} \quad (5.41)$$

$$c_x = -\frac{B_{13} f_x^2}{\lambda} \quad (5.42)$$

$$c_y = \frac{B_{12} B_{13} - B_{11} B_{23}}{B_{11} B_{22} - B_{12}^2} \quad (5.43)$$

$$\lambda = \frac{B_{33} - (B_{13}^2 + c_y (B_{12} B_{13} - B_{11} B_{23}))}{B_{11}} \quad (5.44)$$

bestimmt werden. Mit bekannten intrinsischen Parametern werden die extrinsischen Parameter  $R$  und  $t$  bestimmt durch:

$$r_1 = \lambda A^{-1} h_1 \quad (5.45)$$

$$r_2 = \lambda A^{-1} h_2 \quad (5.46)$$

$$r_3 = r_1 \times r_2 \quad (5.47)$$

$$t = \lambda A^{-1} h_3 \quad (5.48)$$

Die so gewonnenen intrinsischen und extrinsischen Parameter dienen als Initiallösung für das folgende nichtlineare Optimierungsproblem zur zusätzlichen

Bestimmung der radialen und tangentialen Linsenverzerrungsparameter. Radiale und tangentiale Linsenverzeichnung kann nach [33] und [78] durch folgende Ausdrücke

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = (1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_5 r^6) \begin{pmatrix} u \\ v \end{pmatrix} \quad (5.49)$$

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} 2\kappa_3 uv + \kappa_4(r^2 + 2u^2) \\ \kappa_3(r^2 + 2v^2) + 2\kappa_4 uv \end{pmatrix} + \begin{pmatrix} u \\ v \end{pmatrix} \quad (5.50)$$

modelliert werden, wobei  $r^2 = u^2 + v^2$  gilt. Zur Vereinfachung der folgenden Ausführungen wird die zuvor ausführlich erörterte Abbildung von Welt- auf Bildpunkte als Funktion

$$h_e : \mathbb{R}^3 \rightarrow \mathbb{R}^2, \quad h_e(K, \kappa, R, t, P) = \begin{pmatrix} u \\ v \end{pmatrix} \quad (5.51)$$

dargestellt. Die zuvor bereits im Kamerabild detektierten Punkte auf dem Kalibrierobjekt dienen als Referenzpunkte  $R_{ij}$  im folgenden nicht linearen Optimierungsproblem zur Bestimmung der gesuchten Parameter:

$$\min_{(K, \kappa, R_j, t_j)} \sum_j \sum_i \|R_{ij} - h_e(K, \kappa, R_j, t_j, P_i)\|^2. \quad (5.52)$$

Das Ziel ist die Minimierung des Fehlers zwischen den im Bild gemessenen Referenzpunkten  $R_{ij}$  und den mit  $h_e$  abgebildeten Kalibrierobjektpunkten  $P_i$ . Zur Lösung dieses Optimierungsproblems kann beispielsweise der Levenberg-Marquardt-Algorithmus [125] [134] [144] verwendet werden, um die Kameraparameter näherungsweise zu bestimmen. Damit sind sowohl intrinsische als auch extrinsische Parameter einschließlich der Verzerrungsparameter bestimmt. Dabei ist zu beachten, dass dieses Optimierungsproblem die Minimierung eines geometrischen Maßes bedeutet. Im Unterschied dazu wurde die zuvor beschriebene Bestimmung der Initiallösung durch Minimierung eines rein algebraischen Maßes gewonnen.

Falls keine Schätzung der Verzerrungsparameter gewünscht ist, bedeutet die Nachschaltung des linearen Optimierungsverfahrens ohne Berücksichtigung der

Linsenverzerrung daher trotzdem eine Verbesserung der Schätzung der verbleibenden Kameraparameter.

### Katadioptrische Objektive

Werden zusätzlich zu den bspw. in einem entozentrischen Objektiv genutzten Linsen Spiegel als optische Elemente in einem Objektiv verwendet, wird dieses Objektiv als katadioptrisch bezeichnet [20]. In dieser Arbeit wird ein katadioptrisches System verwendet, das eine perspektivische Kamera mit einem hyperbolischen Spiegel kombiniert. So wird eine vollständige Rundumsicht um den Sensor erreicht, obwohl nur eine einzelne Kamera eingesetzt wird. Die Abbildung der Umwelt auf den Bildsensor der Kamera über einen hyperbolisch geformten Spiegel verursacht eine starke Verzerrung des Bilds. Da das im vorherigen Kapitel vorgestellte Kalibrierungsverfahren nicht für die Kalibrierung katadioptrischer Objektive geeignet ist, wird in diesem Kapitel eine alternative Methode für diesen Zweck vorgestellt, die der Arbeit von Scaramuzza et al. [166] entspricht.

Bei der in [166] dargestellten Methode wird die Benutzung eines Sensors mit einem einzigen effektiven Blickpunkt, also ein sogenanntes zentrales katadioptrisches System [13], als Annahme zugrunde gelegt. Weiterhin werden zwei Bezugsebenen definiert, die in der Beschreibung des Kameramodells eine zentrale Rolle spielen: die Bildebene der Kamera, die mit dem Bildsensor der Kamera zusammenfällt (siehe Abb. 5.18 (b)), und die Sensorebene des katadioptrischen Kamerasystems (siehe Abb. 5.18 (c)). Zu einer Menge von Weltpunkten  $P_i = (x \ y \ z)^T \in \mathbb{R}^3$  sind  $P'_i = (u' \ v')^T \in \mathbb{R}^2$  die zugehörigen Punkte auf der Bildebene und  $P''_i = (u'' \ v'')^T \in \mathbb{R}^2$  die zugehörigen Punkte auf der Sensorebene.

Der Zusammenhang zwischen den beiden Referenzebenen ist nach [140] durch die affine Abbildung

$$P'' = AP' + t_c \quad \text{mit} \quad A \in \mathbb{R}^{2 \times 2}, t_c \in \mathbb{R}^2 \quad (5.53)$$

beschrieben. Die Abbildungsfunktion  $g$  definiert den Zusammenhang zwischen einem Punkt  $P''$  auf der Sensorebene und dem Vektor  $U$ , der ausgehend vom

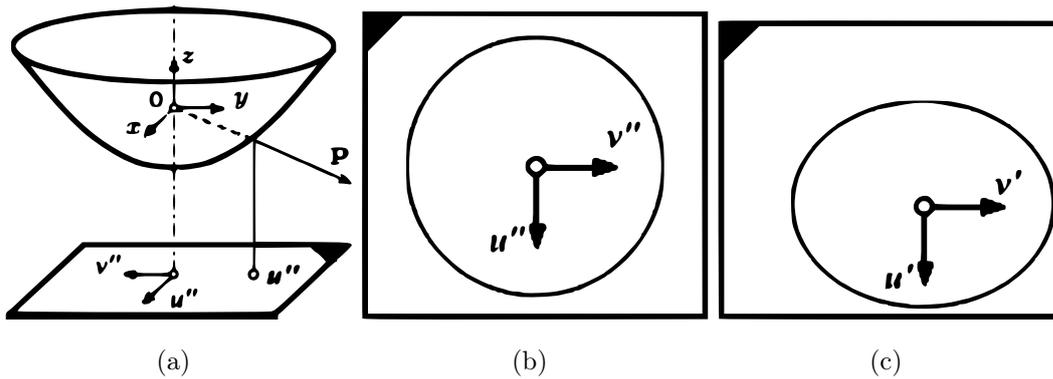


Abbildung 5.18: **Koordinatentransformationen in einem katadioptrischen System mit hyperbolischem Spiegel.** Abb. aus [166].

zentralen Blickpunkt des Objektivs  $Q$  im Weltpunkt  $P$  endet. Mit Gleichung 5.53 zusammen gilt

$$\lambda U = \lambda g(P'') = \lambda g(AP' + t_c) = M \begin{pmatrix} P \\ 1 \end{pmatrix} \quad \text{mit } \alpha > 0. \quad (5.54)$$

$M \in \mathbb{R}^{3 \times 4}$  ist eine perspektivische Projektionsmatrix. Das Ziel der Kalibrierung ist nun, ähnlich wie beim entozentrischen Objektiv, die Bestimmung der unbekannt Bestandteile des Ausdrucks 5.54 unter Nutzung bekannter Welt-Bild-Punktkorrespondenzen. Die Abbildungsfunktion wird modelliert durch

$$g(u'', v'') = \begin{pmatrix} u'' \\ v'' \\ f(u'', v'') \end{pmatrix} \quad (5.55)$$

$$f(u'', v'') = a_0 + a_1 r'' + a_2 r''^2 + \dots + a_N r''^N, \quad (5.56)$$

wobei  $r'' = \sqrt{u''^2 + v''^2}$  gilt und die Koeffizienten  $a_i, i = 0, \dots, N$  und der Grad des Polynoms  $N$  die zu bestimmenden unbekannt Parameter darstellen. Die Funktion  $f$  definiert die Form der eingesetzten optischen Elemente des katadioptrischen Systems. In [166] wird die Nutzung eines Polynoms vorgeschlagen, da so eine Vielzahl von katadioptrischen Systemen beschrieben werden kann. Für hyperbolische und parabolische Spiegel kann  $a_1 = 0$  angenommen werden,

da für diese Spiegelformen  $\frac{df}{dr'} \Big|_{r''=0} = 0$  gilt. Weitere Beispiele unterschiedlicher Funktionsformen im Zusammenhang mit katadioptrischen Systemen finden sich in [111], [141] und [185].

Vereinfachend wird für die folgende Kalibrierung angenommen, dass  $A$  der Einheitsmatrix entspricht und  $t_c = 0$  gilt. Die tatsächlichen Werte für  $A$  und  $t_c$  werden erst in einem nachgeschalteten, nichtlinearen Optimierungsschritt ermittelt. Bei bekannten Bild-Welt-Punktkorrespondenzen mit je  $j$  Weltpunkten, die auf einem planaren Kalibrierungsobjekt liegen, von dem  $i$  Aufnahmen angefertigt wurden, gilt analog zu Kapitel 5.3.2 mit  $z = 0$

$$\lambda_{ij} U_{ij} = \lambda_{ij} \begin{pmatrix} u' \\ v' \\ a_0 + a_1 r' + a_2 r'^2 + \dots + a_N r'^N \end{pmatrix} = \begin{pmatrix} r_1^i & r_2^i & t^i \end{pmatrix} \begin{pmatrix} x_{ij} \\ y_{ij} \\ 1 \end{pmatrix}. \quad (5.57)$$

Die Abhängigkeit vom Skalierungsfaktor  $\lambda_{ij}$  wird durch vektorielle Multiplikation der Gleichung 5.57 mit  $U_{ij}$  eliminiert und es entstehen die folgenden drei Ausdrücke für eine Aufnahme des Kalibrierobjekts:

$$v'_j(r_{31}x_j + r_{32}y_j + t_3) - f(r'_j)(r_{21}x_j + r_{22}y_j + t_2) = 0 \quad (5.58)$$

$$f(r'_j)(r_{11}x_j + r_{12}y_j + t_1) - u'_j(r_{31}x_j + r_{32}y_j + t_3) = 0 \quad (5.59)$$

$$u'_j(r_{21}x_j + r_{22}y_j + t_2) - v'_j(r_{11}x_j + r_{12}y_j + t_1) = 0. \quad (5.60)$$

Gleichung 5.60 ist dabei unabhängig von der nichtlinearen Funktion  $f$ . Bei Nutzung von  $k$  Punktkorrespondenzen und der Gleichung 5.60 entsteht demnach ein lineares Gleichungssystem mit  $k$  Gleichungen, für das mit den aus dem vorherigen Kapitel bekannten Methoden der linearen Algebra ein Lösungsvektor näherungsweise festgestellt werden kann. Bis auf die dritte Komponente des Translationsvektors  $t_3$  werden mit dieser Methode alle relevanten extrinsischen Parameter bestimmt.

Mit dem Einsetzen der gefundenen extrinsischen Parameter in die Gleichungen 5.58 und 5.59 können nun auf analoge Art und Weise auch die intrinsischen Parameter  $a_0, a_2, \dots, a_N$  und die fehlenden Komponenten der Translationsvektoren  $t_3^i$  näherungsweise ermittelt werden. Schließlich fehlt noch die Bestimmung

des optimalen Grads  $N$  des Polynoms  $f$ . Dies wird iterativ durch schrittweise Inkrementierung des Grads  $N$  mit gleichzeitiger Betrachtung des akkumulierten Reprojektionsfehlers über alle Punktkorrespondenzen erreicht. Ist der Fehler minimal, ist der optimale Wert für  $N$  gefunden.

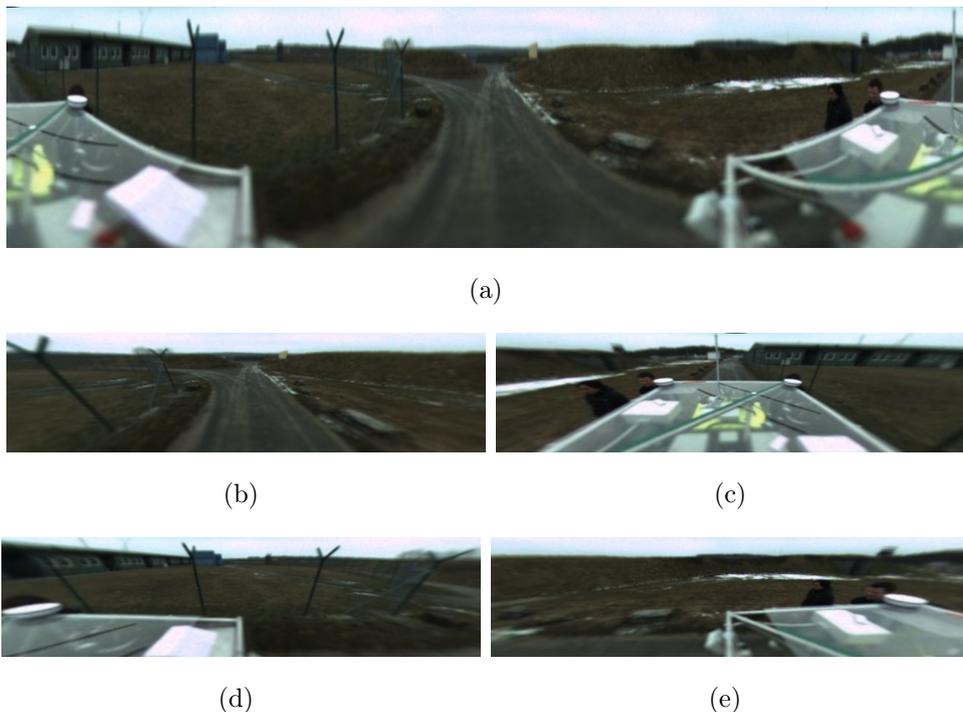


Abbildung 5.19: **Projektionsvarianten von Bilddaten einer omnidirektionalen Kamera.** Projektion auf Zylinder (a). Projektion auf perpendikulare Ebenen (b) bis (e).

Scaramuzza et al. [166] schlagen vor, zwei weitere Optimierungsschritte zur Verfeinerung der Parameter anzuschließen, einen linearen und einen nichtlinearen Schritt. Die lineare Optimierung stellt im Grunde eine Wiederholung der oben dargestellten Schritte dar. Allerdings mit dem Unterschied, dass die zuvor bestimmten Koeffizienten  $a_0, a_2, \dots, a_N$  des Polynoms  $f$  nun bei der Bestimmung der extrinsischen Parameter bekannt sind und damit alle drei Gleichungen 5.58, 5.59 und 5.60 benutzt werden können. Der nichtlineare Optimierungsschritt entspricht der gleichen Vorgehensweise wie zuvor in Kapitel 5.3.2 (siehe Ausdruck 5.51 und 5.52). Analog dazu wird versucht, eine Nähe-

rungslösung für das folgende Optimierungsproblem zu finden:

$$h_k : \mathbb{R}^3 \rightarrow \mathbb{R}^2, \quad h_k(A, t_c, a_0, a_2, \dots, a_N, R, t) = \begin{pmatrix} u' \\ v' \end{pmatrix}, \quad (5.61)$$

$$\min_{(A, t_c, a_0, a_2, \dots, a_N, R_i, t_i)} \sum_j \sum_i \|R_{ij} - h_k(A, t_c, a_0, a_2, \dots, a_N, R_i, t_i)\|^2. \quad (5.62)$$

Die initialen Werte entstammen dabei den vorherigen Schritten. Zur Lösung des Optimierungsproblems wird ebenfalls wieder der Levenberg-Marquardt-Algorithmus herangezogen.

Abschließend zeigt Abb. 5.19 zwei Beispiele für die Anwendung der durch die oben beschriebene Methode gewonnenen intrinsischen Parameter. Die Abbildung visualisiert die Projektion der originalen Bilddaten auf einen Zylinder bzw. auf vier perpendikuläre Ebenen, um eine für den Menschen deutlich intuitivere Darstellung der Bilddaten zu ermöglichen.

### 5.3.3 Extrinsische Kalibrierung

In Kapitel 5.3.2 wird beschrieben, wie für einen visuellen Sensor mit einem entozentrischen oder katadioptrischen Objektiv eine extrinsische Kalibrierung ermittelt werden kann. Dabei wird allerdings das Weltbezugskoordinatensystem so gewählt, dass dessen Ursprung in einem planaren Kalibrierobjekt liegt und die positive z-Achse des Koordinatensystems der Normalen des planaren Kalibrierobjekts entspricht. Damit ist nur die relative räumliche Beziehung zwischen visuellem Sensor und Kalibrierobjekt bekannt. Soll die relative räumliche Beziehung zwischen den Sensorkoordinatensystemen eines visuellen und eines geometrischen Sensors hergestellt werden, ist es daher nötig, die Referenzpunkte auf dem planaren Kalibrierobjekt mit dem geometrischen Sensor zu vermessen. Dies ist aufgrund der planaren Form und der damit fehlenden detektierbaren geometrischen Merkmale nicht ohne Weiteres möglich. Es ist deshalb angeraten, ein alternatives Verfahren einzusetzen, um Punkte zu finden, die sowohl im geometrischen als auch im visuellen Modell präzise und eindeutig aufzufinden und zuzuordnen sind. Zu diesem Zweck kann entweder ein

künstliches Kalibrierobjekt mit passenden geometrischen Eigenschaften oder eine natürliche Szene mit Objekten, die hervorstechende geometrische Merkmale aufweisen, vermessen werden.

### **Ermittlung von Punktkorrespondenzen**

Bei der manuellen Markierung von 2D/3D-Punktkorrespondenzen spielt die Darstellungsform der geometrischen Daten eine entscheidende Rolle für einen erfolgreichen Markierungsprozess. Scaramuzza et al. [167] schlagen beispielsweise eine Umwandlung der geometrischen Daten von der ursprünglichen Punktwolkenform zunächst in ein Tiefenbild und darauf aufbauend in ein sogenanntes "Bearing-Angle-Bild" vor. Jedes Pixel des Tiefenbilds wird farblich abhängig vom Winkel zwischen dem zugehörigen Laserstrahl und der Verbindungsstrecke zwischen je zwei benachbarten Scanpunkten kodiert. Die gesuchten Winkel werden durch

$$\alpha_i = \arccos \frac{d_i - d_{i-1} \cos \Delta\beta}{\sqrt{d_i^2 - d_{i-1}^2 - d_i d_{i-1} \cos \Delta\beta}} \quad (5.63)$$

bestimmt und können entweder in vertikaler, horizontaler oder diagonaler Richtung ermittelt werden, wobei  $d_i$  die Länge des  $i$ -ten Laserabtaststrahls und  $\cos \Delta\beta$  den Winkel zwischen zwei benachbarten Laserstrahlen entlang der betrachteten Richtung definiert.

Obwohl die so entstehende Visualisierung der geometrischen Daten eine starke Verbesserung des manuellen Auswahlprozesses bedeutet, wurde in der vorliegenden Arbeit ein anderer Ansatz gewählt, um den manuellen Auswahlprozess noch einfacher und intuitiver zu gestalten. Unter Ausnutzung des Softwareframeworks OpenSceneGraph<sup>5</sup> zur Visualisierung von dreidimensionalen Daten wird ein mit den in Kapitel 5.2 vorgestellten Methoden erstelltes Oberflächenmodell angezeigt, das interaktiv vom Benutzer rotiert, verschoben und skaliert werden kann.

Dies hat den großen Vorteil, dass ein idealer Blickwinkel für die Markierung jedes Referenzpunktes gewählt werden kann. Zusätzlich dazu können auch

---

<sup>5</sup><http://www.openscenegraph.org/>

Oberflächenpunkte zwischen den Abtastpunkten des Laserscanners mit dieser Methode markiert werden, da ein Oberflächenmodell genutzt wird. Diese Methode hat sich in diversen Experimenten als äußerst intuitiv und effektiv erwiesen, wenn das Ziel ist ohne besondere Vorbereitungsmaßnahmen in einer natürlichen Szene Punktkorrespondenzen zu erzeugen. Abbildung 5.20 zeigt das Ergebnis eines beispielhaften Markierungsvorgangs mit der vorgestellten Methode.

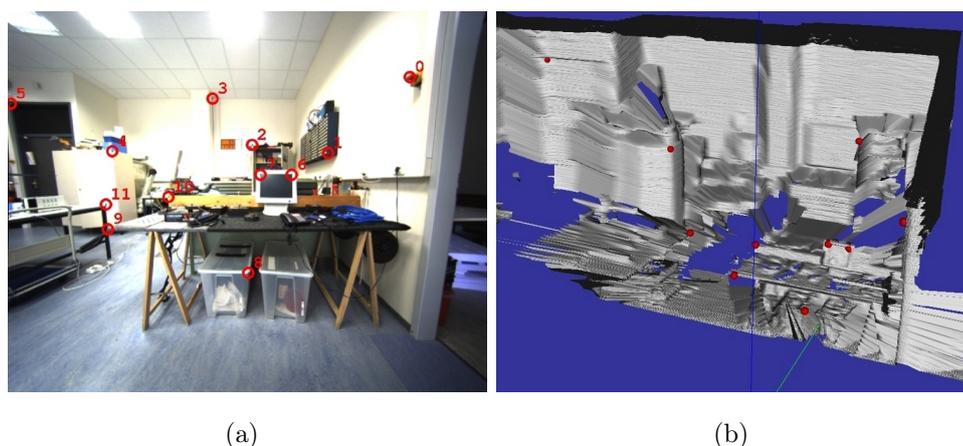


Abbildung 5.20: **Beispielszene mit manuell gewählten 2D/3D-Punktkorrespondenzen.** Punkte im 2D-Kamerabild (a). Punkte im 3D-Umgebungsmodell (b).

Problematisch ist bei der Markierung von Punkten in natürlichen Szenen jedoch die begrenzte örtliche Auflösung des Laserscanners, sodass gerade an Objektkanten oder -ecken, die sich gut für die Markierung von Referenzpunkten eignen, keine präzise Positionierung von Referenzpunkten möglich ist. Abbildung 5.21 zeigt einige Beispiele für diese Problematik.

Dies schlägt sich in der Genauigkeit der extrinsischen Kalibrierung nieder. Die Genauigkeit wird in vielen Arbeiten mit dem sogenannten Reprojektionsfehler quantifiziert, der die Übereinstimmung von markierten Bildpunkten und projizierten markierten Weltpunkten beschreibt. Der durchschnittliche Reprojektionsfehler zwischen den Projektionen  $P'_i \in \mathbb{R}^3$  einer Menge von Welt-Referenzpunkten  $P_i \in \mathbb{R}^2$  auf die Bildebene und den zugehörigen manuell

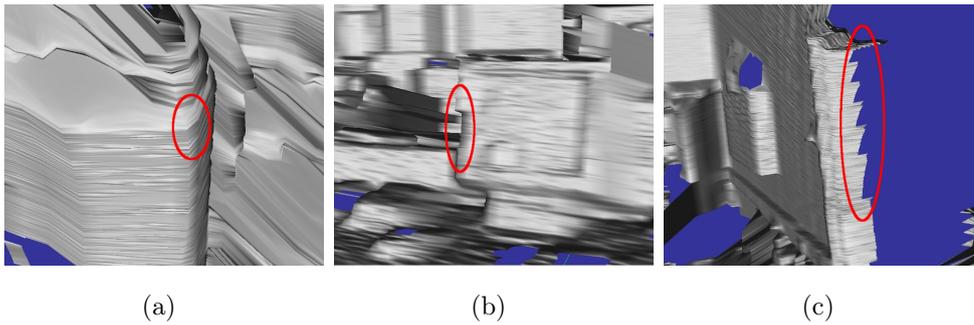


Abbildung 5.21: **Probleme bei der manuellen Auswahl von Punktkorrespondenzen im geometrischen Umgebungsmodell**, die durch die begrenzte örtliche Auflösung des Laserscanners entstehen.

gewählten Bild-Referenzpunkten  $R_i \in \mathbb{R}^2$  wird bestimmt durch

$$e_p = \frac{\sum_i \|R_i - P'_i\|_2}{|P_i|}. \quad (5.64)$$

Bei manueller Wahl der Referenzpunkte konnte mit der oben beschriebenen Methode in einer Reihe von Experimenten ein minimaler durchschnittlicher Reprojektionsfehler von 3,7 Pixeln erreicht werden. Um diesen Fehler weiter zu verringern, wird hier zur präziseren Kalibrierung von Kamera und Laserscanner eine Methode vorgeschlagen, die auf der Nutzung eines künstlichen Kalibrierobjekts basiert. Die Grundidee für die Form des eingesetzten Kalibrierobjekts entstammt der Arbeit von Schroeter und Newman [171]. Die Abbildungen 5.22 (a) und 5.22 (b) zeigen die beiden während dieser Arbeit entstandenen Kalibrierobjekte. Abbildung 5.22 (a) zeigt die erste Version, eine vereinfachte Adaption des in [171] vorgeschlagenen Pyramidenstumpfs. Abbildung 5.22 (b) zeigt eine optimierte Variante, welche die Maximierung der Anzahl der Abtastpunkte auf dem Kalibrierobjekt zum Ziel hat.

Die Konstruktion der Kalibrierobjekte basiert auf der Grundidee, dass ein Laserscanner Flächen deutlich besser abtasten kann als punktförmige Merkmale. Um aus den vermessenen Flächen wiederum Referenzpunkte zu generieren, wird der Schnittpunkt von je drei benachbarten Flächen ermittelt. So können die geometrischen Eckpunkte der zentralen Fläche des Kalibrierobjekts sehr präzise bestimmt werden. Außerdem können durch die farbige Markierung der zentralen Fläche die Eckpunkte auch im Kamerabild problemlos markiert wer-

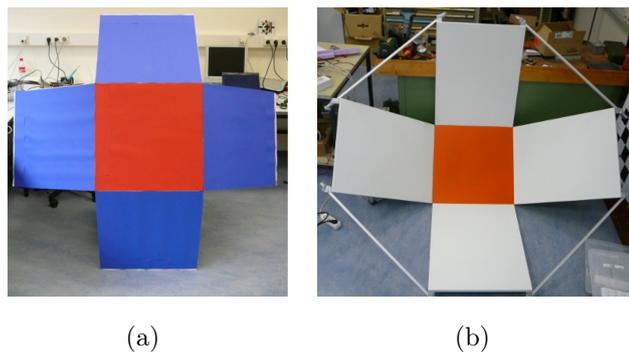


Abbildung 5.22: **Künstliche Kalibrierobjekte für die präzise Ermittlung von 2D/3D-Punktkorrespondenzen. [123]**

den. Die Schätzung der Ausgleichsebenen auf Basis der Abtastpunkte auf den unterschiedlichen Flächen geschieht mit dem in Kapitel 5.2.4 vorgestellten Verfahren, die den Normalenvektor  $n$  der zu schätzenden Ebene zum Ergebnis hat. Der Schnittpunkt  $S \in \mathbb{R}^3$  von je drei Ebenen ist dann definiert durch

$$S = \begin{pmatrix} n_1^T \\ n_2^T \\ n_3^T \end{pmatrix}^{-1} \begin{pmatrix} n_1 P_1 \\ n_2 P_2 \\ n_3 P_3 \end{pmatrix} \quad (5.65)$$

mit den Ebenennormalen  $n_1, n_2, n_3 \in \mathbb{R}^3$  und den Ebenenpunkten  $P_1, P_2, P_3 \in \mathbb{R}^3$ . Mit den fünf verfügbaren Ebenen des Kalibrierobjekts können vier Referenzpunkte bestimmt werden. Durch mehrfaches Aufnehmen des Kalibrierobjekts in verschiedenen Posen können so beliebig viele präzise Referenzpunkte für die extrinsische Kalibrierung ermittelt werden. In Abb. 5.23 ist das Ergebnis der Bestimmung von Referenzpunkten durch das soeben beschriebene Verfahren visualisiert.

Für die extrinsische Kalibrierung mit Kalibrierobjekt konnte eine deutliche Verbesserung des Reprojektionsfehlers auf 1,7 Pixel bei einer vergleichsweise hohen Kameraauflösung von  $1280 \times 960$  Pixeln erreicht werden. Im Folgenden wird nun beschrieben, wie aus den erzeugten 2D/3D-Punktkorrespondenzen die relative Lage zwischen den Sensorkoordinatensystemen von Laserscanner und Kamera errechnet werden kann.

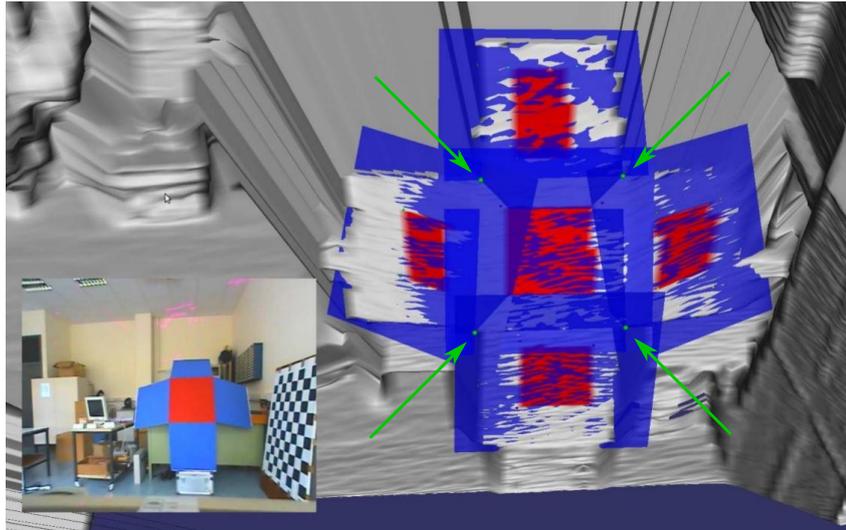


Abbildung 5.23: **Ebenenanpassung zur präzisen Bestimmung von Kalibrierpunkten auf einem künstlichen Kalibrierobjekt.** Aus selektierten Abtastpunkten (rot) geschätzte Ebenen (blau) auf einem künstlichen Kalibrierobjekt. Der Schnittpunkt von je drei Ebenen erzeugt einen Referenzpunkt (grün) für die extrinsische Kalibrierung. [123]

### Das PnP-Problem

Die Bestimmung der Orientierung und der Position einer perspektivischen Kamera mit bekannter intrinsischer Kalibrierung und bekannten Welt-Bild-Punktkorrespondenzen ist ein seit Langem bekanntes und oft bearbeitetes Problem. Fischler und Bolles [59] bezeichnen das Problem zum ersten Mal als PnP<sup>6</sup>-Problem, wobei  $n$  die Anzahl der bekannten Punktkorrespondenzen definiert. Dieser Begriff ist mittlerweile etabliert und es existiert eine Vielzahl an Lösungsansätzen für das PnP-Problem. Darunter sind iterative, nichtiterative, lineare und nichtlineare Ansätze. Ist weder Rotation noch Translation der Kamera im Weltkoordinatensystem bekannt, werden mindestens drei Punkte zur geschlossenen Lösung des PnP-Problems benötigt, wobei Lösung im Fall von  $n = 3$  bedeutet, dass bis zu vier, also endlich viele, Lösungsalternativen errechnet werden können. Durch Rückprojektion eines vierten Punkts kann jedoch eine eindeutige Lösung ermittelt werden. Haralick et al. geben in [74] einen Überblick über die bis zum damaligen Zeitpunkt (1991) bestehenden

<sup>6</sup>engl.: Perspective-n-Point

Lösungsansätze für das Problem. Darunter befinden sich in chronologischer Reihenfolge die Algorithmen von Grunert [70], Finsterwalder und Scheufele [58], Merritt [139], Fischler und Bolles [59], Linnainmaa et al. [128] und Grafarend et al. [68]. Gao et al. stellen in [63] fest, dass die Untersuchung des P3P-Problems zwei Aspekte beinhaltet. Erstens geht es darum, einen schnellen und stabilen Algorithmus zu entwickeln, der alle oder zumindest einige Lösungen des Problems ermittelt. Zweitens ist genau zu analysieren, unter welchen Bedingungen, also bei welchen Punktkonfigurationen, ein, zwei, drei oder vier Lösungen für das P3P-Problem existieren. Vor allen Dingen der zweite Aspekt ist in den vorherigen Arbeiten unterrepräsentiert, sodass Gao et al. in ihrer Arbeit sowohl einen Algorithmus zur Lösung des P3P-Problems als auch eine Klassifikation des P3P-Problems präsentieren. DeMenthon und Davis [46] [47] stellen eine im Vergleich zu anderen Arbeiten beschleunigte Lösungsvariante vor, die auf der Approximation der exakten perspektivischen Abbildung durch sogenannte para- bzw. orthoperspektivische Abbildungen basieren. Dies stellt einen Kompromiss zwischen Komplexität und Präzision dar. Quan und Lan [158] reduzieren das Problem auf eine biquadratische Gleichung mit Sylvester-Resultante und präsentieren darauf aufbauend eine Lösung des Problems mit Mitteln der linearen Algebra. Außerdem wird nachgewiesen, dass mit dieser Methode auch das PnP-Problem linear geschlossen für  $n \geq 4$  gelöst werden kann. In [124] wird von Lepetit et al. eine sehr effiziente Methode für  $n \geq 4$  mit der Komplexität  $O(n)$  beschrieben, die auf der Beschreibung der dreidimensionalen Referenzpunkte durch virtuelle Kontrollpunkte fußt. Im Gegensatz zu anderen Methoden, die ebenfalls eine Punktzahl  $n \geq 4$  verarbeiten können und vergleichbar unanfällig für Rauschen sind, ist diese Methode sehr zeiteffizient. So hat der Ansatz von Quan und Lan [158] eine Komplexität von  $O(n^5)$  und die Methode von Ansar und Daniilidis [5] sogar  $O(n^8)$ .

Die Notwendigkeit, eine lineare geschlossene (und damit zeiteffiziente) Lösung für das PnP-Problem zu finden, entsteht aus der Echtzeitanforderung vieler Anwendungen, so z.B. der visuellen 3D-Rekonstruktion [2] oder der visuellen Odometrie [147]. Ist diese Anforderung nicht gegeben, wie im Fall der hier angestrebten extrinsischen Kamerakalibrierung, kann demnach auch auf iterative Algorithmen zurückgegriffen werden. Diese sind zwar rechenintensiver, jedoch dafür deutlich präziser, wie in den vergleichenden Studien in [99], [124] und

[217] herausgestellt wird.

Wie schon in Kapitel 5.3.2 beschrieben wurde, ist für die erfolgreiche und robuste Konvergenz iterativer Optimierungsalgorithmen eine ausreichend genaue Initiallösung zu wählen. Die zuvor beschriebenen linearen, geschlossenen Lösungen des PnP-Problems (bspw. [124]) können in diesem Kontext zur Bestimmung einer Initiallösung eingesetzt werden.

Die iterative Lösung des PnP-Problems basiert auf der Minimierung eines Fehlermaßes. Klassischerweise ist dieses Fehlermaß im Bildraum definiert und entspricht einer Minimierung des Reprojektionsfehlers bei bekannten intrinsischen Kameraparametern. Als Verfahren zur Lösung des daraus entstehenden nicht-linearen Optimierungsproblems können entweder die Gauss-Newton-Methode [209] oder die Levenberg-Marquardt-Methode eingesetzt werden [144]. Die Gauss-Newton-Methode zeigt im Vergleich eine deutlich schnellere Konvergenz zur korrekten Lösung, ist aber von einer guten Initiallösung abhängig, da das Verfahren sonst scheitert. Die Levenberg-Marquardt-Methode konvergiert langsam bei schlecht gewählter Initiallösung, hat jedoch deutlich robustere Konvergenzeigenschaften und verhält sich bei guter Initialschätzung wie die Gauss-Newton-Methode. Als Alternative zu den beiden klassischen Ansätzen hat sich der Algorithmus von Lu et al. [132] etabliert, der im Gegensatz zu den beiden vorherigen generischen Optimierungsverfahren die speziellen Eigenschaften des PnP-Problems in den Iterationsprozess mit einbezieht. Das Fehlermaß für deren sogenannte Orthogonale Iterationsmethode wird nicht im Bild-, sondern direkt im Objektraum definiert. So wird eine deutlich höhere Konvergenzgeschwindigkeit bei nahezu identischer Genauigkeit erreicht.

Grundsätzlich ist noch erwähnenswert, dass eine Kombination von Lösungsverfahren für das PnP-Problem mit dem in [59] eingeführten RANSAC<sup>7</sup>-Verfahren zur Eliminierung von Ausreißern in den verrauschten Eingangsdaten stets angeraten ist, wenn es die Rechenzeitanforderungen erlauben.

---

<sup>7</sup>RANdom SAmples Consensus

### Alternative Parametrisierung des PnP-Problems

Die Verfahren zur Lösung des PnP-Problems, die in Abschnitt 5.3.3 vorgestellt wurden, gehen von einer klassischen perspektivischen Abbildungsfunktion im Sinne des Lochkameramodells aus. Um das PnP-Problem jedoch auch für Kameras mit anderen Abbildungsformen lösen zu können, schlugen Kneip et al. [100] [99] eine alternative Beschreibung von Bildinformationen vor. Ein Beispiel dafür ist das in dieser Arbeit eingesetzte katadioptrische Kamerasystem. Die Grundannahme der zentralen Kamera mit einem einzigen zentralen Blickpunkt bleibt dabei bestehen. Bildinformationen werden nach [100] nicht in zweidimensionalen Pixelkoordinaten dargestellt, sondern in Form von dreidimensionalen Einheitsvektoren  $b_i \in \mathbb{R}^3$ , den sogenannten 'Bearing-Vektoren', die ausgehend vom Zentrum des Kamerasystems in Richtung der abzubildenden Weltpunkte  $P_i \in \mathbb{R}^3$  zeigen. Die Vektoren  $b_i$  haben zwei Freiheitsgrade und können vollständig durch Azimut- und Polarwinkel beschrieben werden.

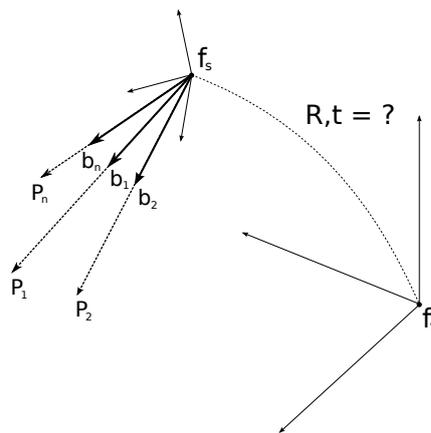


Abbildung 5.24: **Beschreibung von Bildinformationen durch *Bearing-Vektoren***.  $f_v$  bezeichnet das Referenzkoordinatensystem für die Punkte  $P_i$ .  $f_s$  ist das Sensorkoordinatensystem der genutzten Kamera. Abbildung analog zu Kneip et al. [99].

Abbildung 5.24 zeigt zusammenfassend die alternative Parametrisierung des PnP-Problems. Ergänzend sei an dieser Stelle noch erwähnt, dass beispielsweise in den Arbeiten von Chen und Chang [37] [38] die Betrachtung des PnP-Problems in Richtung einer vollkommen verallgemeinerten Beschreibung der Abbildungsvorschrift von Welt- auf Bildpunkte weiterentwickelt wurde. Da

dies jedoch bez. der in dieser Arbeit eingesetzten Sensorik nicht nötig war, wurde vom Einsatz einer Methode dieser Art abgesehen.

Mit bekannter intrinsischer Kalibrierung der Kameras (siehe Kapitel 5.3.2) kann nun ein Optimierungsproblem auf der Basis von Bearing-Vektoren  $b_i$  und zugehörigen Weltpunkten  $P_i = (x_i \ y_i \ z_i)^T$  formuliert werden. Der Bearing-Vektor

$$b_i = \frac{b'_i}{\|b'_i\|} \quad (5.66)$$

errechnet sich bei Nutzung eines entozentrischen Kameraobjektivs aus dem Bildpunkt  $P'_i = (u_i \ v_i)^T$  durch

$$b'_i = \begin{pmatrix} \frac{u_i - c_x}{f_x} \\ 1 \\ -\frac{v_i - c_y}{f_y} \end{pmatrix}. \quad (5.67)$$

Wird ein katadipotrisches Kameraobjektiv eingesetzt, ändert sich der Ausdruck wie folgt:

$$b'_i = \begin{pmatrix} b''_i \\ f(u_i, v_i) \end{pmatrix} \quad (5.68)$$

$$b''_i = \frac{1}{\det A} \text{adj}(A) \begin{pmatrix} u_i - c_x \\ -v_i + c_y \end{pmatrix}, \quad (5.69)$$

wobei  $f$  die zuvor definierte polynomialen Abbildungsfunktion definiert (siehe Gleichung 5.56).

Ab diesem Punkt können nun beide Objektivtypen einheitlich genutzt werden, da in beiden Fällen die Bildinformationen als Bearing-Vektoren  $b_i$  dargestellt sind. Das darauf aufbauende Optimierungsverfahren zur Bestimmung von Rotation und Translation des Kamerakoordinatensystems hat wiederum die Minimierung des Reprojektionsfehlers zum Ziel. Aufgrund der Darstellung der Bilddaten durch Bearing-Vektoren ist dieser jedoch anders als sonst üblich zu definieren. Der Reprojektionsfehler  $e_b^i$  ist direkt abhängig vom Winkel  $\theta_i$  zwi-

schen dem Bearing-Vektor  $b_i$  und dem Bearing-Vektor  $b_i^+$  der Rückprojektion und kann wie folgt bestimmt werden:

$$T = \begin{pmatrix} R^T & t' \end{pmatrix} \text{ mit } t' = -R^T t \quad (5.70)$$

$$b_i^+ = \frac{\hat{b}_i^+}{\|\hat{b}_i^+\|} \text{ mit } \hat{b}_i^+ = T \begin{pmatrix} P_i \\ 1 \end{pmatrix} \quad (5.71)$$

$$e_b^i = 1 - b_i \cdot b_i^+ \quad (5.72)$$

Der Gesamtfehler ist damit definiert durch die Summe der Einzelfehler. Diesen Gesamtfehler gilt es abhängig von Rotationsmatrix  $R$  und Translationsvektor  $t$ , welche die Transformation von Kamera- zu Weltkoordinatensystem definieren, zu minimieren:

$$\min_{(R,t)} \sum_i e_b^i. \quad (5.73)$$

Auch in diesem Fall wird das Levenberg-Marquardt-Verfahren eingesetzt, um eine Lösung für das Optimierungsproblem zu finden. Damit ist die Transformation zwischen Kamera- und Weltkoordinatensystem ermittelt. Das Weltkoordinatensystem entspricht im Fall der Registrierung von Daten aus einem geometrischen und einem visuellen Sensor dem Koordinatensystem des geometrischen Sensors, da dieser das Bezugssystem für die gemessenen dreidimensionalen Weltpunkte darstellt. Folglich kann mit der ermittelten Kalibrierung die Abbildung von Weltpunkten aus dem zuvor erzeugten geometrischen Modell auf Bildpunkte der visuellen Sensoren erreicht werden.

### 5.3.4 Texturierung des geometrischen Oberflächenmodells

Für jeden von einem Laserscanner gemessenen Weltpunkt kann nach erfolgreicher intrinsischer und extrinsischer Kalibrierung einer Kamera (siehe Kapitel 5.3.2 und 5.3.3) der zugehörige Bildpunkt ermittelt werden, welcher der Projektion des Weltpunktes auf die Bildebene der Kamera entspricht. Die einfachste Art, eine gescannte Punktwolke mit visuellen Daten aus einer registrierten Kamera zu kombinieren, ist damit, jeden Weltpunkt in der Punktwolke mit der

Farbinformation des zugehörigen Bildpunkts einzufärben. Durch Erweiterung der Messpunktvektoren  $P_i \in \mathbb{R}^3$  zu

$$\tilde{P}_i = (x \ y \ z \ \tilde{u} \ \tilde{v})^T \quad (5.74)$$

wird ein eingefärbter Punkt dargestellt. Die Komponenten  $\tilde{u}, \tilde{v} \in [0, 1]$  definieren dabei die normalisierten Texturkoordinaten des zugehörigen Bildpunkts. Die Umrechnung zwischen normalisierten Texturkoordinaten  $(\tilde{u}, \tilde{v})$  und Bildkoordinaten  $(u, v)$  in Pixel ist definiert durch

$$\tilde{u} = \frac{u}{w_I} \quad (5.75)$$

$$\tilde{v} = \frac{v}{h_I}, \quad (5.76)$$

wobei  $w_I$  und  $h_I$  die horizontale bzw. vertikale Auflösung des Texturbilds  $I$  in Pixel sind. Mit der Abbildungsfunktion

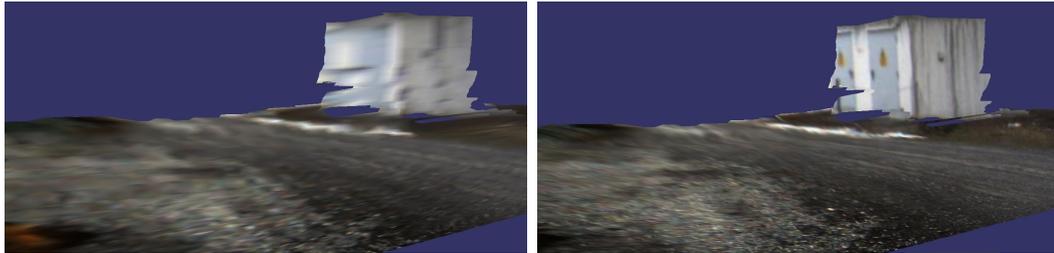
$$I : \mathbb{R}^2 \rightarrow \mathbb{R}^3, \quad I(\tilde{u}, \tilde{v}) = (c_r \ c_g \ c_b)^T \quad (5.77)$$

werden die normalisierten Farbkomponenten  $c_r, c_g, c_b \in [0, 1]$  eines Texturpixels auf der Grundlage des RGB-Farbmodells ermittelt. Für eine Punktwolke bedeutet die punktweise Einfärbung jedes Elements der Punktwolke eine maximale Informationsausbeute der registrierten visuellen Daten.

Wird allerdings ein Oberflächenmodell, bspw. ein Dreiecksnetz, als geometrisches Modell genutzt, wird bei punktweiser Einfärbung jedes Vertex des Dreiecksnetzes eine lineare Interpolation auf Basis der Farben der drei Eckpunkte des Dreiecks durchgeführt, um einen kontinuierlichen Farbübergang zu produzieren. Realisiert wird dies durch Nutzung baryzentrischer Koordinaten zur Beschreibung von Punkten auf den Dreiecken des Modells. Der Farbeinfluss jedes Vertex auf einen Punkt auf einem Dreieck nimmt dabei bez. der jeweils gegenüberliegenden Seiten proportional ab. So entsteht eine lineare Farbmischung aus den drei Eckfarben für Punkte auf den Dreiecken des Modells.

Problematisch bleibt dabei, dass die Ortsauflösung der in dieser Arbeit genutzten geometrischen Sensoren deutlich kleiner ist als die der visuellen Sensoren.

Folglich wird ein Großteil der visuellen Information bei der punktweisen Texturierung verworfen. Um dies zu verhindern, wird statt einer Interpolation im Farbraum eine Interpolation im Texturkoordinatenraum durchgeführt. So wird die visuelle Information des Kamerabildes im texturierten Oberflächenmodell optimal ausgenutzt, was sich im visuellen Ergebnis des Fusionsprozesses deutlich zeigt (siehe Abb. 5.25).



(a)

(b)



(c)

Abbildung 5.25: **Vergleich unterschiedlicher Texturinterpolationsmethoden.** Interpolation im Farbraum (a) vs. Interpolation im Texturkoordinatenraum (b). Frontkamerabild der Szene (c).

Werden wie in der vorliegenden Arbeit multiple Kameras mit unterschiedlichen Abbildungsfunktionen und/oder Öffnungswinkeln miteinander kombiniert für die Texturierung eines Oberflächenmodells verwendet, ist es bei überlappenden Öffnungswinkel nötig, die redundanten Bildinformationen auf adäquate Art zu fusionieren. Die effizienteste Variante besteht in einer festen Priorisierung der genutzten Kameras, abhängig von Auflösung und Bildqualität der entsprechenden Kameras, um eine optimale Ausnutzung der visuellen Daten zu gewährleisten. Aufgrund der überlegenen Auflösung und Bildqualität wird

daher in dieser Arbeit die weitwinklige Frontkamera der omnidirektionalen Kamera mit katadioptrischem Objektiv vorgezogen.

## 5.4 Oberflächentexturierung mit Live-Luftbildern

In Kapitel 2.2 wurde bereits detailliert der komplementäre Charakter von bodenbasierten und fliegenden Roboterfahrzeugen herausgestellt. Der fundamental unterschiedliche Blickwinkel des Flugroboters auf die aktuelle lokale Szene bietet dabei im Kontext der Umgebungsmodellierung durch Sensorfusion vollkommen neue Möglichkeiten der Anreicherung des vom Bodenroboter erzeugten geometrischen Modells mit visuellen Informationen.

Die Methoden zur Registrierung von geometrischen und visuellen Sensoren, die bisher in dieser Arbeit in Kapitel 5.3.3 eingeführt wurden, gehen von der Grundannahme aus, dass eine Abbildungsfunktion von Welt- auf Bildkoordinaten der visuellen Sensorik existiert, die statisch, also nicht über die Zeit veränderlich ist. Im Fall der Registrierung eines auf dem Bodenroboter montierten geometrischen Sensors mit einem auf einer externen Sensorplattform montierten visuellen Sensor sind vollständig neue Methoden der Registrierung einzusetzen, da die Transformation zwischen Kamerakoordinatensystem des Flugroboters und Fahrzeugkoordinatensystem des Bodenroboters stets für jede Szene neu zu bestimmen ist.

Auf dem Flugroboter PSYCHE ist die Kamera in einer aktiv nivellierenden Aufhängung montiert (siehe Kapitel 2.3). Daher wird die vereinfachende Annahme zugrunde gelegt, dass die Bildebene der Luftbildkamera des Flugroboters und die Bodenebene, auf welcher der Bodenroboter AMOR steht, parallel zueinander sind. Abbildung 5.26 fasst diesen Zusammenhang grafisch zusammen. Für kleine Winkel  $\varphi$  kann als zusätzliche Vereinfachung der Zusammenhang

$$h_c \approx h_u - h_s \quad (5.78)$$

angenommen werden.

Die Abbildung von Welt- auf Bildkoordinaten ist unter der Annahme der Pa-

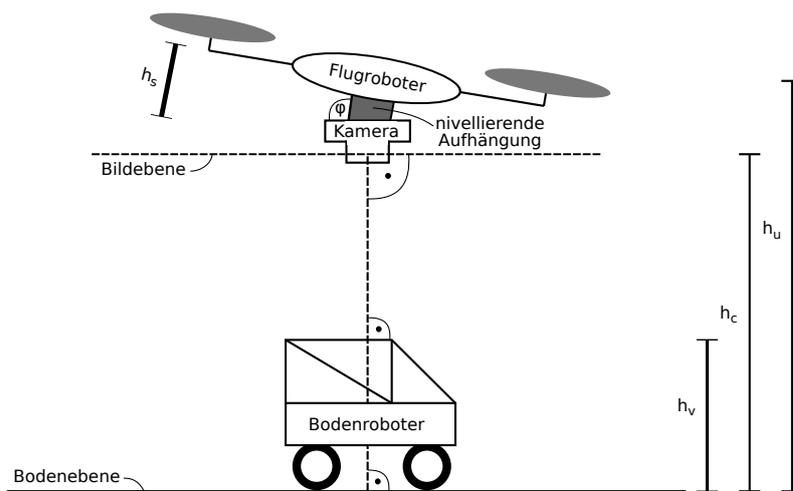


Abbildung 5.26: **Skizze der vereinfachenden Annahme des Zusammenhangs zwischen Luftbildkamera und Bodenebene bei nivellierender Kameraaufhängung.**  $h_u$  definiert die Höhe über Boden des Flugroboters,  $h_s$  beschreibt den Abstand des Kamerazentrums vom Zentrum des Fahrzeugkoordinatensystems des Flugroboters,  $h_c$  entspricht der Höhe des Kamerazentrums über Boden und  $h_v$  ist die Höhe des Bodenroboters.

rallelität von Bild- und Bodenebene abhängig von der Höhe  $h_c$  des Luftbildkamerazentrums über der Bodenebene, den intrinsischen Kalibrierungsparametern  $f = (f_x, f_y)$  und  $c = (c_x, c_y)$  der Luftbildkamera sowie der relativen Pose  $\rho'_v = (p_x^v \ p_y^v \ \phi^v)^T$ , welche die 2D-Transformation in der Bodenebene zwischen Projektion des Bildkoordinatensystems der Luftbildkamera auf die Bodenebene entlang der optischen Achse der Kamera und der Projektion des Fahrzeugkoordinatensystems auf die Bodenebene entlang der z-Achse beschreibt. Die Höhe des Kamerazentrums  $h_c$  kann mithilfe des Luftdrucksensors des Flugroboters und des zuvor beschriebenen Ausdrucks 5.78 bestimmt werden. Die intrinsischen Parameter der Kamera werden durch das in Kapitel 5.3.2 beschriebene Kalibrierungsverfahren ermittelt. Die Hauptkomplexität besteht damit in der Berechnung der relativen Pose  $\rho'_v$  des Abbilds des Bodenroboters im Kamerabild.

### 5.4.1 Posenbestimmung des Bodenroboters im Live-Luftbild

Die Posenbestimmung des Bodenroboters im Live-Luftbild wird durch Einsatz von Methoden der digitalen Bildverarbeitung erreicht. Im Rahmen dieser Arbeit wird in diesem Zusammenhang ein pixelbasierter Ansatz gewählt, um das Problem der Detektion des Bodenroboters im Live-Luftbild zu lösen, da sich dieser Ansatz unter den speziellen Randbedingungen der gegebenen Problemstellung als besonders robust gegenüber Bewegungsunschärfe, unterschiedlichen Beleuchtungsverhältnissen und perspektivischer Verzerrung des Bildes des Bodenroboters erwiesen hat.

Das in den nachfolgenden Ausführungen vorgestellte Verfahren ist speziell auf die effiziente und präzise Detektion eines apriori bekannten Bodenfahrzeugs in einem hochaufgelösten Luftbild ausgelegt. Das Verfahren fällt damit in die Kategorie der Algorithmen, die zum Ziel haben, ein spezielles Objekt aufzufinden, also kein Objekt aus einer generischen Objektkategorie. Im Fall des Bodenroboters AMOR sieht das aufzufindende Bildobjekt aus wie in Abbildung 5.27 dargestellt. Das Bildobjekt, im Folgenden Schablone genannt, wurde in diesem Fall durch manuelle Freistellung des Fahrzeugs in einem möglichst zentral aufgenommenen Luftbild erzeugt, um den Einfluss der perspektivischen Projektion auf die zu erzeugende Schablone zu minimieren.

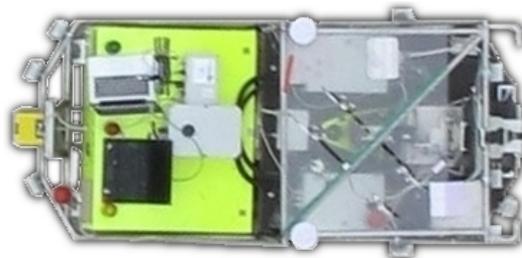


Abbildung 5.27: **Schablone für die Bestimmung der Pose des Bodenroboters im Luftbild:** Freigestelltes visuelles Modell des Roboters AMOR.

Das Detektionsverfahren unterteilt den Detektionsprozess in zwei Unterprobleme: Zunächst wird im gesamten Luftbild  $I_{\text{uav}}$  eine lokale Region-of-Interest<sup>8</sup>  $I_{\text{roi}}$  ermittelt, wobei dabei ein Farbsegmentierungsverfahren zum Einsatz kommt

---

<sup>8</sup>kurz: ROI

(siehe Abb. 5.28). Im zweiten Schritt werden mittels eines Schablonenvergleichs die Orientierung und die Position des Bodenroboters im Luftbild berechnet.

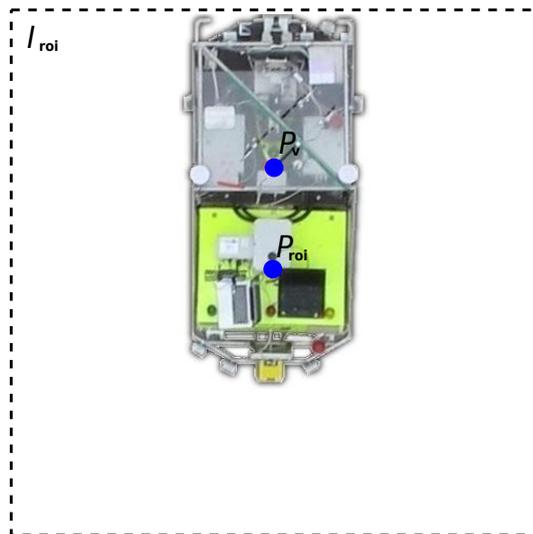


Abbildung 5.28: **Region-of-Interest für den Schablonenvergleich.**  $P_v$  definiert die Projektion des Ursprungs des Fahrzeugkoordinatensystems auf die Bildebene, der die gesuchte Position des Bodenroboters im Luftbild darstellt.  $P_{roi}$  beschreibt den Mittelpunkt der zuvor im ersten Verfahrensschritt ermittelten ROI, welche die Grundlage für den zweiten Verfahrensschritt darstellt.

### Bestimmung der Region-of-Interest für den Schablonenvergleich

Der erste Schritt macht sich die auffällige Färbung des rückwärtigen Teils des Bodenroboters zunutze und realisiert eine Farbsegmentierung des Luftbilds. Konkret bedeutet dies, dass eine pixelweise Binarisierung des im HSV<sup>9</sup>-Farbraum dargestellten Luftbilds

$$I_{uav}(u, v) = (c^h \ c^s \ c^v)^T \quad (5.79)$$

aufgrund des Farbwertvektors  $(c^h \ c^s \ c^v)^T$  des jeweils betrachteten Pixels durch-

---

<sup>9</sup>Hue Saturation Value

geführt wird. Das binarisierte Bild  $I'_{\text{uav}}$  ist definiert durch

$$I'_{\text{uav}}(u, v) = \begin{cases} 1, & \text{wenn } c^h \in [c^h_{\min}, c^h_{\max}] \wedge c^s \in [c^s_{\min}, c^s_{\max}] \wedge c^v \in [c^v_{\min}, c^v_{\max}] \\ 0, & \text{sonst.} \end{cases} \quad (5.80)$$

Die einschränkenden minimalen und maximalen Farbwerte  $c^h_{\min}$ ,  $c^h_{\max}$ ,  $c^s_{\min}$ ,  $c^s_{\max}$ ,  $c^v_{\min}$  und  $c^v_{\max}$  wurden experimentell unter Nutzung der zuvor erzeugten Schablone des Bodenroboters (siehe Abb. 5.27)) ermittelt. Zur Unterdrückung von kleinflächigen Fehldetektionen im analysierten Luftbild wird nach der Binarisierung auf Farbwert-Basis eine Tiefpassfilterung mit anschließender erneuter Binarisierung mit einem festen Schwellwert durchgeführt. Für die Tiefpassfilterung kommt dabei ein Rechteckfilter mit einer quadratischen Filtermaske  $k$  der Form

$$k = \frac{1}{nm} \left( \begin{array}{ccc} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{array} \right) \left. \vphantom{\begin{array}{ccc} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{array}} \right\} n \times m \text{ Elemente} \quad (5.81)$$

zum Einsatz. Die Seitenlängen  $n$  und  $m$  der Filtermaske in Pixel bestimmen sich abhängig von der Bodenaufösung des Luftbilds

$$\begin{pmatrix} n \\ m \end{pmatrix} = s_k h_c \begin{pmatrix} f_x \\ f_y \end{pmatrix}, \quad (5.82)$$

die mit der Seitenlänge des Rechteckfilters  $s_k$  (in m) skaliert und damit in Bildkoordinaten transformiert wird. Das Ergebnis der Anwendung der Filtermaske ist definiert durch

$$I''_{\text{uav}}(u, v) = \sum_{i=1}^n \sum_{j=1}^m I'_{\text{uav}}\left(u + i - \frac{n}{2} + 1, v + j - \frac{m}{2} + 1\right) k(i, j). \quad (5.83)$$

Die Binarisierung von  $I''_{\text{uav}}$  mit festem Schwellwert  $t$  ist festgelegt durch

$$I'''_{\text{uav}}(u, v) = \begin{cases} 1, & I''_{\text{uav}}(u, v) > t \\ 0, & \text{sonst.} \end{cases} \quad (5.84)$$

Zur Bestimmung des Mittelpunkts  $(\bar{u}_d \ \bar{v}_d)^T$  der gesuchten ROI  $I_{\text{roi}}$  werden die Bildkoordinaten der detektierten Bildpunkte im tiefpassgefilterten Bild gemittelt:

$$P_{\text{roi}} = \begin{pmatrix} \bar{u}_d \\ \bar{v}_d \end{pmatrix} = \frac{\sum_u \sum_v I'''_{\text{uav}}(u, v) \begin{pmatrix} u \\ v \end{pmatrix}}{\sum_u \sum_v I'''_{\text{uav}}(u, v)} \quad (5.85)$$

Die Seitenlängen der ROI  $l_u$  und  $l_v$  in Pixel sind analog zu Ausdruck 5.82 durch

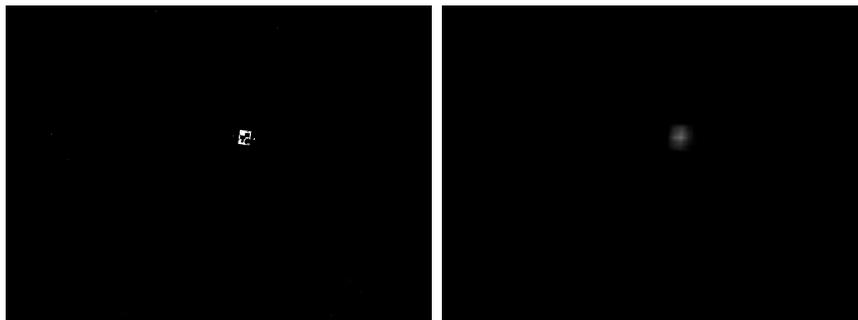
$$\begin{pmatrix} l_u \\ l_v \end{pmatrix} = s_{\text{roi}} h_c \begin{pmatrix} f_x \\ f_y \end{pmatrix}, \quad (5.86)$$

definiert.  $s_{\text{roi}}$  legt die Seitenlänge der ROI  $s_{\text{roi}}$  (in m) fest.

Das Ergebnis der verketteten Ausführung von Binarisierung, Tiefpassfilterung und Schwellwertanwendung zur Bestimmung der gesuchten ROI einer Beispielszene ist in Abbildung 5.29 zu sehen. Die gesuchte 2D-Position ist mit der Berechnung von  $I_{\text{roi}}$  jedoch noch nicht bestimmt, da sich das Zentrum des Fahrzeugkoordinatensystems von AMOR nicht im Mittelpunkt von  $I_{\text{roi}}$  befindet (siehe Abb. 5.28). Es bedarf daher einer Verschiebung der berechneten Position entlang der y-Achse des Fahrzeugkoordinatensystems von AMOR in Richtung des Schwerpunkts des Fahrzeugs. Da zu diesem Zeitpunkt jedoch die Orientierung des Bodenroboters noch nicht bekannt ist, kann die Richtung der Verschiebung noch nicht festgestellt werden. Die endgültige Bestimmung der gesuchten Position wird daher zusammen mit der Berechnung der Orientierung des Fahrzeugs im nächsten Schritt durchgeführt.

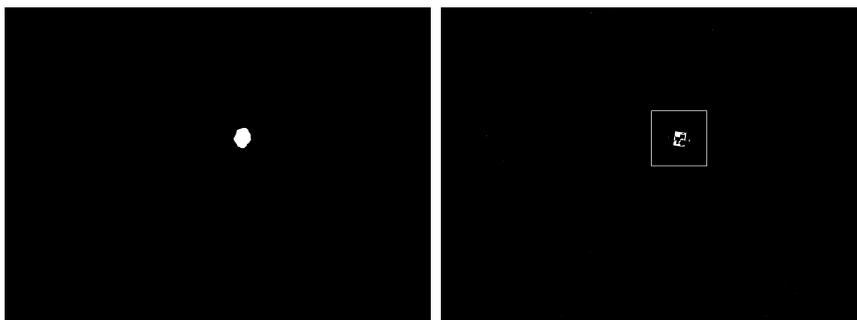


(a)



(b)

(c)



(d)

(e)

Abbildung 5.29: **Visuelle Positionsermittlung der Region-of-Interest um den Bodenroboter für den darauf folgenden Schablonenvergleich.** Ausgangsbild (a), Binarisierung durch Analyse der Farbwerte im HSV-Farbraum (b), Tiefpassfilterung mit Rechteckfilter (c), Binarisierung durch Anwendung eines festen Schwellwerts (d) und ermittelte ROI durch Mittelwertbildung über die detektierten Pixel (e).

### Bestimmung der Orientierung durch Schablonenvergleich

Die ermittelte ROI  $I_{roi}$  stellt die Grundlage für die folgende Bestimmung der Orientierung des Bodenroboters und darauf aufbauend die endgültige Position des Ursprungs des Fahrzeugkoordinatensystems im Luftbild dar. Zu diesem Zweck wird die in Abb. 5.27 dargestellte Schablone  $I_{tpl}^0$  schrittweise rotiert, um so eine Menge von Schablonen zu erzeugen, die in fester Winkelauflösung mit der Winkeländerung  $\Delta\xi$  je Schritt alle möglichen Orientierungen des Bodenroboters repräsentieren. Der Winkel  $\xi$  entspricht der gesuchten Ausrichtung des Bodenroboters, wobei die initiale Schablone dem Winkel  $\xi = 0$  zugeordnet ist. Die Transformation des Schablonenbilds  $I_{tpl}^\xi$  wird durch Rotation der Bildpunkte mit dem Winkel  $\xi$  erreicht:

$$\begin{pmatrix} u_{tpl}^\xi \\ v_{tpl}^\xi \end{pmatrix} = \begin{pmatrix} \cos(\xi) & -\sin(\xi) \\ \sin(\xi) & \cos(\xi) \end{pmatrix} \begin{pmatrix} u_{tpl}^0 \\ v_{tpl}^0 \end{pmatrix} \quad (5.87)$$

Abschließend wird aus der Menge aller rotierten Schablonen diejenige ermittelt, welche die quadratische Differenz der Farbwerte aller Bildpunkte zwischen Schablone und Region-of-Interest des Live-Luftbilds minimiert.

$$\xi_{min} = \min_{\xi} \sum_u \sum_v \sqrt{(I_{roi}(u, v) - I_{tpl}^\xi(u, v))^2} \quad (5.88)$$

Nach Bestimmung der Schablone, die den minimalen Fehler erzeugt, ist somit die Orientierung des Bodenroboters bestimmt. Abschließend wird noch die Position des Bodenroboters entlang des Orientierungsvektors verschoben, um die korrekte Position zu ermitteln (siehe 5.4.1 zur Erklärung). Der zugehörige Translationvektor  $t_{tpl}^{\xi_{min}}$  ist definiert durch

$$t_{tpl}^{\xi_{min}} = \kappa \begin{pmatrix} -\sin \xi_{min} \\ \cos \xi_{min} \end{pmatrix}, \quad (5.89)$$

wobei  $\kappa$  den aus der Schablone ermittelten Abstand zwischen dem projizierten Ursprung des Fahrzeugkoordinatensystems und dem Mittelpunkt der zuvor ermittelten Region-of-Interest darstellt. Das Ergebnis ist schließlich die Position

$P_v = P_{\text{roi}} + t_{\text{tpl}}^{\xi_{\text{min}}}$  und die Orientierung  $\xi_{\text{min}}$  des Bodenroboters im Bildkoordinatensystem. Zur vereinfachten Beschreibung der folgenden Ausführungen werden Position und Orientierung zur 2D-Pose  $\rho'_I = (P_v \ \xi_{\text{min}})^T$  zusammengefasst.

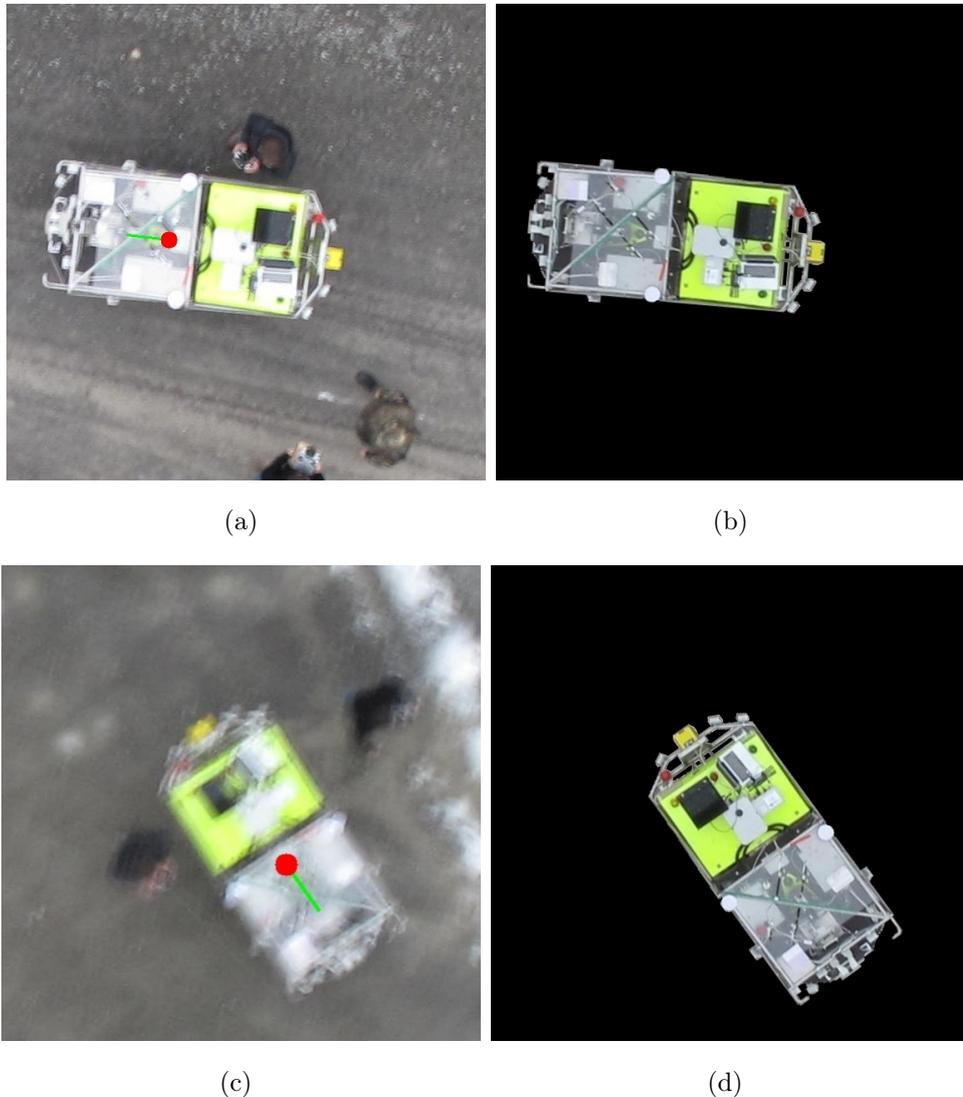


Abbildung 5.30: **Ergebnis des Schablonenvergleichs zur Bestimmung von Position und Orientierung des Bodenroboters in zwei Beispielszenen:** Position (rot) und Orientierung des Bodenroboters (grün).

Zwei beispielhafte Resultate des Schablonenvergleichs sind in Abbildung 5.30 anhand von zwei Beispielszenen dargestellt. Auch im Fall von verminderter Bildqualität des Live-Luftbilds, etwa durch Bewegungsunschärfe, variable Be-

leuchtungsverhältnisse und perspektivische Verzerrung der gesuchten Schablone (siehe Abb. 5.30 (c) und (d)) funktioniert das Verfahren robust.

Die zuvor bestimmte Position  $P_v$  wird nun mithilfe der intrinsischen Parameter der Kamera und der Höhe des Flugroboters transformiert, um die Positionskomponente der Pose  $\rho'_v$  in Weltkoordinaten zu erhalten (siehe Abschnitt 5.4 für die Definition von  $\rho'_v$ ).

$$\Delta P_v = P_v - c \quad (5.90)$$

$$\Delta P'_v = \begin{pmatrix} \cos \xi_{min} & -\sin \xi_{min} \\ \sin \xi_{min} & \cos \xi_{min} \end{pmatrix} \Delta P_v \quad (5.91)$$

$$\begin{pmatrix} p'_x \\ p'_y \end{pmatrix} = \frac{1}{h'_c} \begin{pmatrix} f_x & 0 \\ 0 & f_y \end{pmatrix} \Delta P'_v \quad \text{mit } h'_c = h_c - h_v \quad (5.92)$$

Bei bekannter Pose  $\rho'_v$  können für alle Punkte  $P_i = (x_i \ y_i \ z_i)^T \in \mathbb{R}^3$  des geometrischen Umweltmodells des Bodenroboters die zugehörigen Texturkoordinaten  $(u_i \ v_i)^T$  ermittelt werden, womit die Registrierung von visuellen und geometrischen Daten erreicht ist:

$$\begin{pmatrix} x'_i \\ y'_i \\ z'_i \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \begin{pmatrix} p'_x \\ p'_y \\ 0 \end{pmatrix} \quad (5.93)$$

$$\begin{pmatrix} x''_i \\ y''_i \\ z''_i \end{pmatrix} = \begin{pmatrix} \cos -\xi_{min} & -\sin -\xi_{min} & 0 \\ \sin -\xi_{min} & \cos -\xi_{min} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x'_i \\ y'_i \\ z'_i \end{pmatrix} \quad (5.94)$$

$$u_i = \frac{f_x x''_i}{h_c - z''_i} + c_x \quad (5.95)$$

$$v_i = \frac{f_y y''_i}{h_c - z''_i} + c_y. \quad (5.96)$$

### 5.4.2 Zeitliche Synchronisation bei bewegtem Bodenroboter

Die Texturierung des geometrischen Umgebungsmodells mit Live-Luftbilddaten besteht aus den folgenden sequenziell ausgeführten Arbeitsschritten mit den entsprechenden Ausführungszeitpunkten:

1. Auslösung der Luftbildkamera ( $t_{a_0}$ ),
2. Aufnahme eines Bilds durch die Luftbildkamera ( $t_{a_1}$ ),
3. Versendung des Luftbilds von Flug- zu Bodenroboter ( $t_{a_2}$ ),
4. Empfang und Verarbeitung des aufgenommenen Luftbilds ( $t_{b_0}$ ) und
5. Texturierung des geometrischen Umgebungsmodells des Bodenroboters ( $t_{b_1}$ ).

Zwischen den Ausführungen dieser fünf Arbeitsschritte besteht jeweils ein zeitlicher Versatz. Solange sich der Bodenroboter nicht bewegt, hat diese Tatsache keinen Einfluss auf das Ergebnis der Registrierung der Sensordaten von Luft- und Bodenroboter. Bei bewegtem Roboter jedoch entsteht ein Fehler in der Registrierung. Grund dafür ist, dass die Bestimmung der Pose des Roboters zum Zeitpunkt  $t_{b_0}$  passiert, das Luftbild, auf dessen Grundlage die Bestimmung durchgeführt wird, allerdings bereits deutlich früher, zum Zeitpunkt  $t_{a_1}$ , aufgenommen wurde. In der Zwischenzeit hat sich der Bodenroboter schon weiterbewegt. Diese Bewegung muss in die korrekte Verarbeitung der Luftbilddaten einbezogen werden. In dieser Arbeit wird die Bestimmung der entsprechenden Zeitpunkte dadurch erschwert, dass erstens die Zeitbasen auf Flug- und Bodenroboter nicht synchronisiert sind und zweitens der tatsächliche Auslösezeitpunkt der eingesetzten Kompaktkamera zur Erzeugung der Luftbilder aufgrund der Funktionsweise der Kamera nicht exakt bestimmbar ist. Daher wurde experimentell die durchschnittliche Zeitdifferenz  $\Delta \bar{t}_{a_1:b_0}$  zwischen  $t_{b_0}$  und  $t_{a_1}$  ermittelt, die beschreibt, wie viel Zeit zwischen Aufnahme eines neuen Luftbilds und Empfang des Luftbilds auf dem Bodenroboter im Mittel vergeht. Da die Standardabweichung der verschiedenen experimentell ermittelten Zeitdifferenzen  $\Delta t_{a_1:b_0}$  hinreichend klein ist, wird die durchschnittliche Zeitdifferenz als konstante Bilderzeugungszeit auf dem Flugroboter angenommen. Durch diese

Annahme kann der Zeitpunkt der Bildaufnahme ausschließlich bezüglich der Zeitbasis des Bodenroboters wie folgt beschrieben werden:

$$t_k = t_{b_0} - \Delta \bar{t}_{a_1:b_0}. \quad (5.97)$$

Aufbauend auf der in Kapitel 5.2.3 eingeführten Hierarchie von Koordinatensystemen wird ein weiterer Knoten  $f_u \in \mathbb{F}$  zum Baum der Koordinatensysteme  $G$  hinzugefügt. Dieses Koordinatensystem beschreibt die relative Lage der dem Luftbild zugeordneten Pose, bezogen auf das aktuelle Szenenkoordinatensystem  $f_n \in \mathbb{F}$  (siehe Abb. 5.31).



Abbildung 5.31: **Vergleich von Luftbild- ( $f_u$ ) und Szenenkoordinatensystem ( $f_n$ ) in einer Beispielszene.** Der Ursprung des Luftbildkoordinatensystems liegt stets an der Stelle des detektierten Bodenroboters im Bild. Der Ursprung des Szenenkoordinatensystems ist bei bewegtem Bodenroboter in den meisten Fällen dagegen verschoben.

Die Historie der von der Odometrie erzeugten Posen des Bodenroboters wird über einen begrenzten Zeitraum vorgehalten, um es zu ermöglichen, dem Luftbild die korrekte Pose des Bodenroboters aus der Posenhistorie zuzuordnen. Die aktuellste Odometriepose des Bodenroboters  ${}^v_k\check{\rho}$  zum Zeitpunkt  $t_0$  wird bezeichnet mit  ${}^v_k\check{\rho}[t_0]$ . Posen vergangener Zeitpunkte werden mit  ${}^v_k\check{\rho}[t_1], {}^v_k\check{\rho}[t_2], \dots, {}^v_k\check{\rho}[t_N]$  bezeichnet, wobei  $N$  die Länge der vorgehaltenen Historie definiert. Die Odo-

metriepose, die dem Luftbild zuzuordnen ist, entspricht derjenigen Pose, deren zugeordneter Zeitpunkt zeitlich am nächsten am Aufnahmezeitpunkt  $t_k$  liegt, d.h. die Pose  ${}^v_k\check{\rho}[t_i]$  mit dem Index  $i$ , der den folgenden Ausdruck minimiert:

$$\min_{i \in [0 \dots N]} \|t_i - t_k\|. \quad (5.98)$$

Folglich gilt für die aktuelle Szene die Definition:

$${}^u_k\check{\rho} := {}^v_k\check{\rho}[t_i]. \quad (5.99)$$

Mit der Pose  ${}^u_k\check{\rho}$  kann der Zusammenhang zwischen Luftbildkoordinatensystem und aktuellem Szenenkoordinatensystem in Form der Pose  ${}^u_n\check{\rho} = ({}^u_n\check{p}, {}^u_n\check{q})$  mit

$${}^u_n\check{p} = {}^n_k\bar{q} ({}^u_k\check{p} - {}^n_k\check{p}) {}^n_kq \quad (5.100)$$

$${}^u_n\check{q} = {}^n_k\bar{q} {}^u_k\check{q} \quad (5.101)$$

ermittelt werden. Die Pose wird zur Transformation der Punkte  $P_i$  des geometrischen Umgebungsmodells, um diese per Vorwärtstransformation (siehe Gleichungen 5.12 und 5.13) in das Luftbildkoordinatensystem  $f_u$  zu projizieren. Diese Transformation wird vor der Projektion der Punkte in das Bildkoordinatensystem der Luftbildkamera zur Bestimmung der Texturkoordinaten durchgeführt und ist definiert durch

$$\check{P}'_i = ({}^u_n\bar{q} (\check{P}_i - {}^u_n\check{p}) {}^u_n\check{q}) \text{ mit } \check{P}_i = \begin{pmatrix} 0 \\ P_i \end{pmatrix} \mid \check{P} \in \mathbb{H}. \quad (5.102)$$

## 5.5 Ergebnisse

Abschließend werden in diesem Unterkapitel noch eine Reihe von Beispielszenen gezeigt, welche die Ergebnisse der in dieser Arbeit vorgestellten Verfahren zur Umgebungsmodellierung durch Sensorfusion repräsentieren. Abbildung

5.32 gibt einen Überblick über den Ablauf der Testfahrt, aus der die folgenden Beispielszenen extrahiert wurden. Die Positionen der ausgewerteten Szenen sind in Abbildung 5.32 mit den Buchstaben A bis H markiert.

Jede Szene wird durch eine Punktwolke und ein daraus erzeugtes Dreiecksnetz dargestellt, das jeweils in Falschfarben eingefärbt sind. Zusätzlich zeigen drei weitere Dreiecksnetze die Szene mit jeweils unterschiedlicher Texturierung unter Nutzung der drei in dieser Arbeit genutzten Kameratypen (Front-, Omni- und Luftbildkamera, wobei auch die Bilder der jeweiligen Kamera zur verbesserten Übersicht jeweils passend abgebildet sind.

Wie in den folgenden Ergebnisabbildungen zu sehen ist, ist zusammenfassend festzustellen, dass die Methoden zur Umweltmodellierung durch Sensorfusion, die in dieser Arbeit entwickelt wurden, die Erzeugung von informationsreichen lokalen Umgebungsmodellen ermöglichen, die sowohl geometrische als auch visuelle Sensordaten integrieren. Es entstehen präzise registrierte Punktwolken, wobei die Registrierung durch die gleichzeitige Bewegung des Roboters und der beteiligten Sensoren, die während der Fahrt zusätzlich aktiv geschwenkt werden, eine komplexe Aufgabe darstellt. Erzeugte Punktwolken werden zu einem Dreiecksnetz weiterverarbeitet, wobei sowohl für geordnete als auch ungeordnete Punktwolken entsprechende Triangulationsmethoden zur Verfügung stehen. Das aus diesen Schritten entstehende geometrische Oberflächenmodell bietet die ideale Grundlage für die präzise Identifikation von befahrbaren und nicht befahrbaren Bereichen in der aktuellen lokalen Szene.

Die Integration visueller Informationen erweitert das geometrische Modell und stellt auf der einen Seite eine intuitive und visuell ansprechende Form der Umgebungsbeschreibung dar und eröffnet auf der anderen Seite neue Möglichkeiten für die Interpretation des Umweltmodells, da rein geometrische Informationen in einigen Situationen keine eindeutigen Schlüsse bezüglich Beschaffenheit oder Flexibilität von Flächen und Objekten im Fahrzeugumfeld zulassen. Bei der Fusion von geometrischen und visuellen Daten durch Texturierung des geometrischen Modells ist die Nutzung eines Oberflächenmodells gegenüber einer Modellierung durch Punktwolken deutlich im Vorteil, da die im Vergleich örtlich höher aufgelöste visuelle Sensorinformation vollständig genutzt werden kann. Eine Reduktion der visuellen Information auf die geometrischen Abtastpunkte wie im Fall der Punktwolkenbeschreibung entfällt.

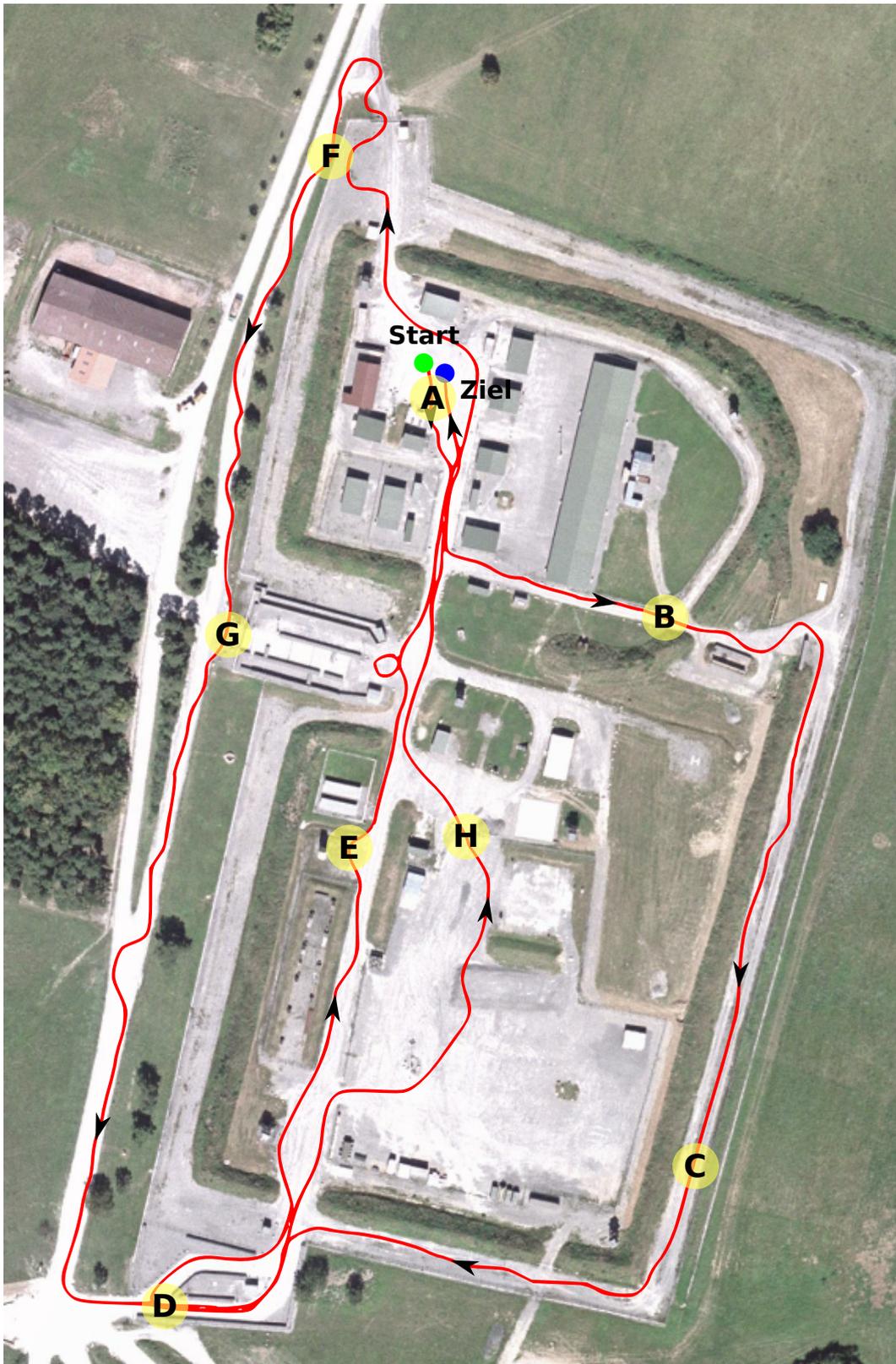
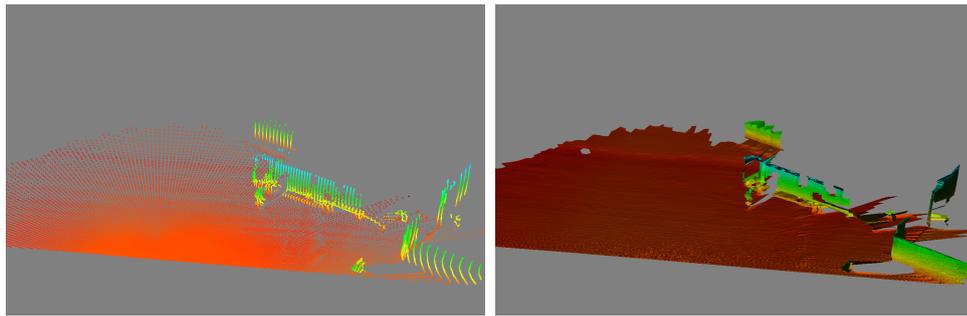


Abbildung 5.32: Überblick einer Testfahrt des Roboterteams bestehend aus AMOR und PSYCHE zur Evaluation der entwickelten Methode zur kooperativen Umweltmodellierung. Ausgewertete Szenen sind mit den Buchstaben A bis H markiert.

Ein Vergleich der verschiedenen Texturierungsvarianten durch die Nutzung unterschiedlicher Kameratypen zeigt, dass allen drei Varianten jeweils sinnvolle Anwendungsszenarien in der autonomen mobilen Außenbereichsrobotik zukommen. Die Texturierung mit einer weitwinkligen Frontkamera eignet sich immer dann besonders gut, wenn eine hochpräzise Registrierung von Umgebungsgeometrie und Bilddaten gefordert ist und der Betrachtungsfokus hauptsächlich im Bereich vor dem Roboter liegt. Gerade im nahen und mittleren Entfernungsbereich glänzt das entstehende Modell mit einem sehr hohen Detailgrad. Bei größeren Entfernungen sind die Effekte der perspektivischen Verzerrung zunehmend stärker ausgeprägt, was sich negativ auf die Qualität des Umgebungsmodells auswirkt. Ähnlich verhält es sich bei der Nutzung einer omnidirektionalen Kamera mit katadioptrischem Objektiv. Auch hier ist eine präzise Registrierung der beteiligten Sensoren durch den statischen räumlichen Zusammenhang der zugehörigen Sensorkoordinatensysteme möglich. Der große zusätzliche Vorteil ist die Rundumsicht des Sensors, die es erlaubt, auch die Bereiche neben und hinter dem Roboter zu texturieren. Eine Minderung der visuellen Modellqualität durch Verzerrungseffekte und die konstruktionsbedingte Unterausnutzung der zur Verfügung stehenden Sensorfläche ist jedoch auch bei dieser Variante festzustellen. Die Nutzung einer Luftbildkamera auf einem kooperierenden Flugroboter bietet in diesem Zusammenhang sehr attraktive neue Möglichkeiten. Durch den fundamental anderen Blickwinkel auf die Szene und die näherungsweise parallele Lage der Bildebene zur genäherten Bodenebene, auf der sich der Bodenroboter bewegt, entsteht ein deutlich anderer visueller Szeneneindruck. Verzerrungseffekte wie bei den anderen Kameras sind bei Weitem nicht so stark ausgeprägt und viele durch Verdeckungen aus der Bodenperspektive uneinsehbare Bereiche werden aus der Aufsicht der Luftbildkamera sichtbar. Die Variation der Flughöhe ermöglicht dabei zusätzlich die Einstellung der Parameter Bodenauflösung und Ausmaß der abgedeckten Bodenfläche je nach Anwendungsszenario. Die Registrierung der Sensorkoordinatensysteme ist bei Nutzung einer Luftbildkamera jedoch ungleich komplexer, da der Zusammenhang der Koordinatensysteme dynamisch ist und schritthalten während der Operation der kooperierenden Roboter aktualisiert werden muss. Die in Kapitel 5.4 beschriebenen Grundbedingungen einer nivellierenden Kameraaufhängung und eines konstanten zeitlichen Zusammenhangs zwischen

Auslösungszeitpunkt der Luftbildkamera und Zeitpunkt des Eintreffens des Kamerabildes auf dem Bodenroboter sind in Kombination mit einer moderaten Bewegungsgeschwindigkeit des Bodenroboters Garant für geringe Fehler bei der Registrierung der Luftbilddaten.

Dem im einleitenden Kapitel 1 definierten Ziel, der nachhaltigen Kartierung der erkundeten Umgebung mit einem adäquaten Modell während der Explorationsmission des Robotersteams, wird mit den in dieser Arbeit entwickelten Methoden eine leistungsfähige Lösung zugeführt. Dies wird durch die folgenden Beispielszenen nochmals besonders ausführlich unterstrichen und dokumentiert.

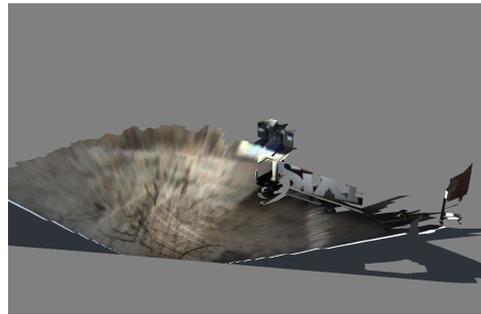


(a) Punktwolke

(b) Dreiecksnetz



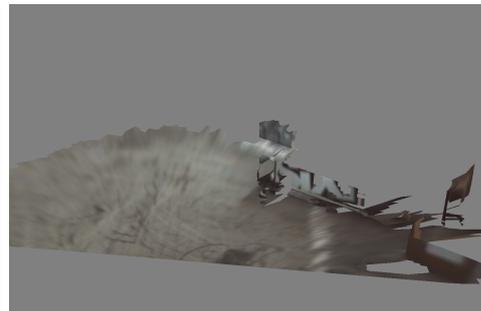
(c) Frontkamerabild



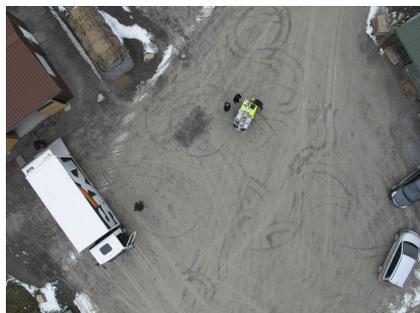
(d) Texturiertes Dreiecksnetz (Frontkamera)



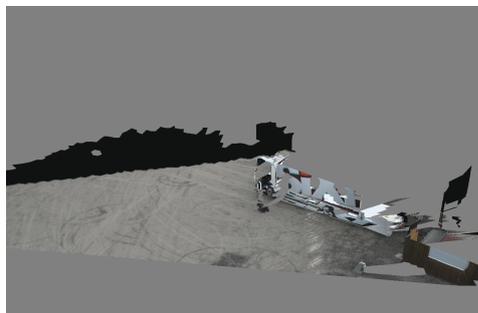
(e) Omnikamerabild



(f) Texturiertes Dreiecksnetz (Omnikamera)

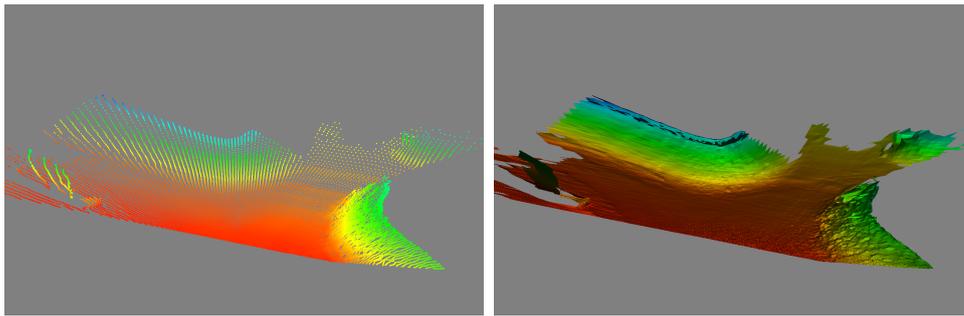


(g) Luftbild



(h) Texturiertes Dreiecksnetz (Luftbildkamera)

Abbildung 5.33: **Szene A**: Ergebnisse der Kooperation Umweltmodellierung durch Sensorfusion.



(a) Punktwolke

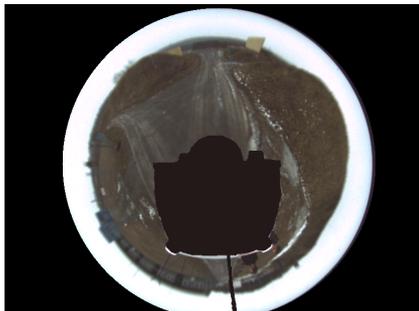
(b) Dreiecksnetz



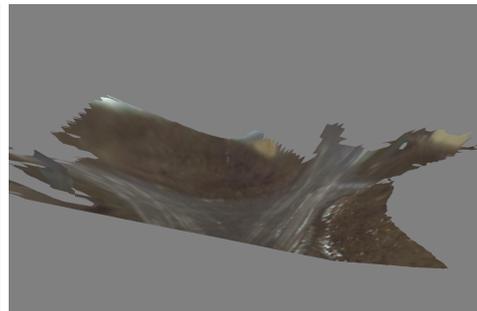
(c) Frontkamerabild



(d) Texturiertes Dreiecksnetz (Frontkamera)



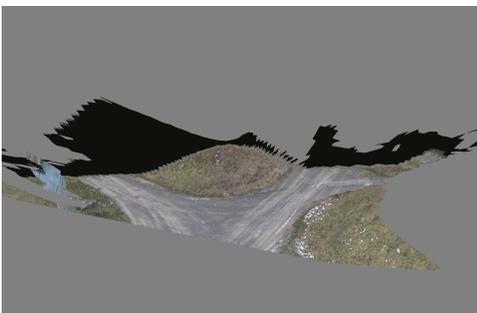
(e) Omnikamerabild



(f) Texturiertes Dreiecksnetz (Omnikamera)

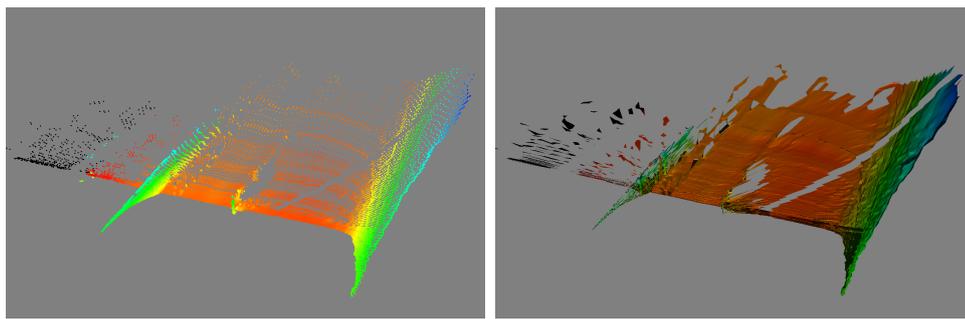


(g) Luftbild



(h) Texturiertes Dreiecksnetz (Luftbildkamera)

Abbildung 5.34: **Szene B**: Ergebnisse der Kooperation Umweltmodellierung durch Sensorfusion.



(a) Punktwolke

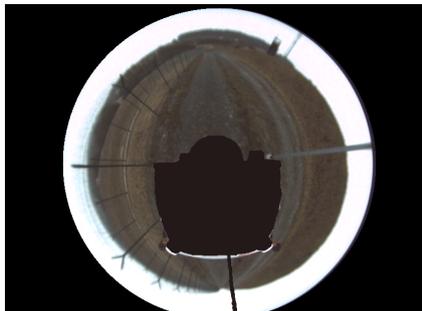
(b) Dreiecksnetz



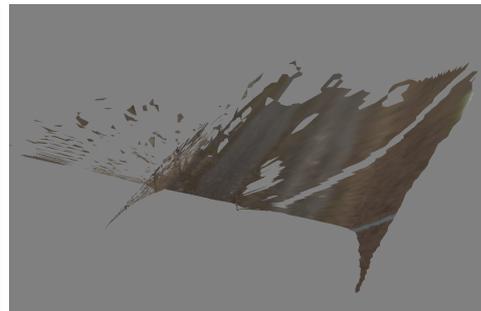
(c) Frontkamerabild



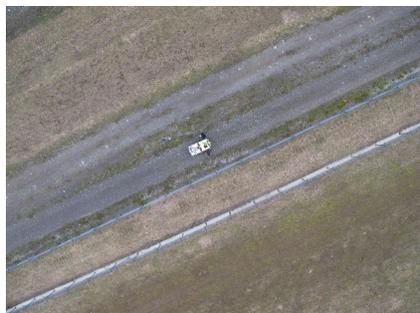
(d) Texturiertes Dreiecksnetz (Frontkamera)



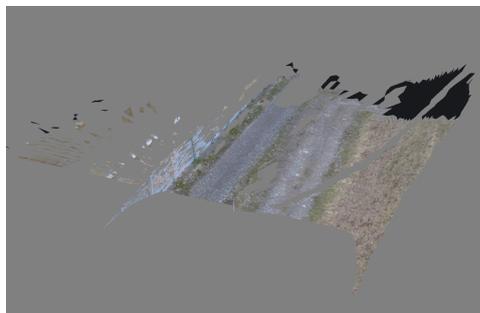
(e) Omnikamerabild



(f) Texturiertes Dreiecksnetz (Omnikamera)

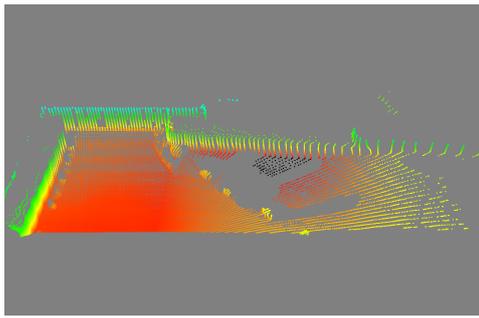


(g) Luftbild

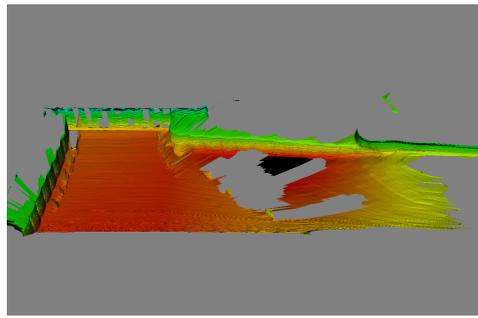


(h) Texturiertes Dreiecksnetz (Luftbildkamera)

Abbildung 5.35: **Szene C**: Ergebnisse der Kooperation Umweltmodellierung durch Sensorfusion.



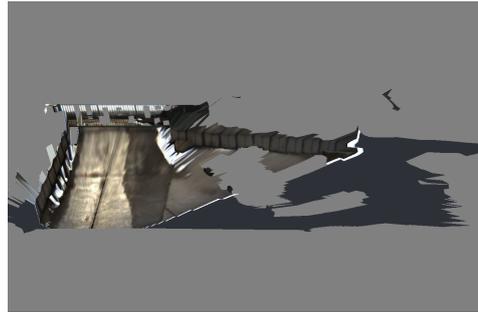
(a) Punktwolke



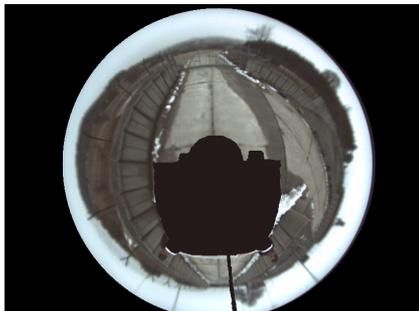
(b) Dreiecksnetz



(c) Frontkamerabild



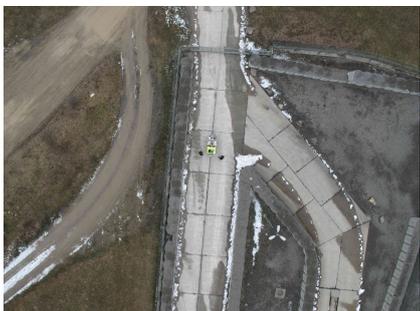
(d) Texturiertes Dreiecksnetz (Frontkamera)



(e) Omnikamerabild



(f) Texturiertes Dreiecksnetz (Omnikamera)

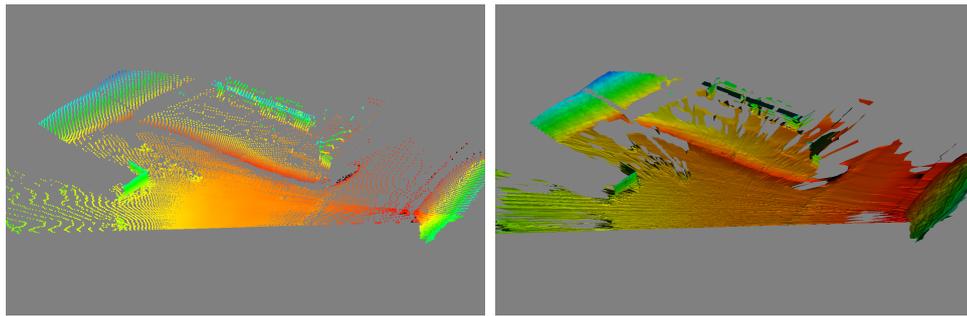


(g) Luftbild



(h) Texturiertes Dreiecksnetz (Luftbildkamera)

Abbildung 5.36: **Szene D**: Ergebnisse der kooperativen Umweltmodellierung durch Sensorfusion.

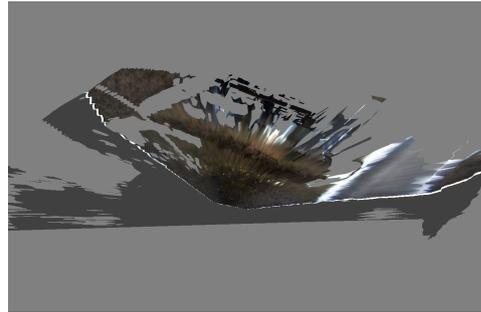


(a) Punktwolke

(b) Dreiecksnetz



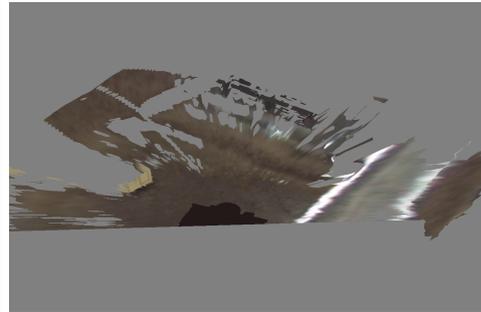
(c) Frontkamerabild



(d) Texturiertes Dreiecksnetz (Frontkamera)



(e) Omnikamerabild



(f) Texturiertes Dreiecksnetz (Omnikamera)

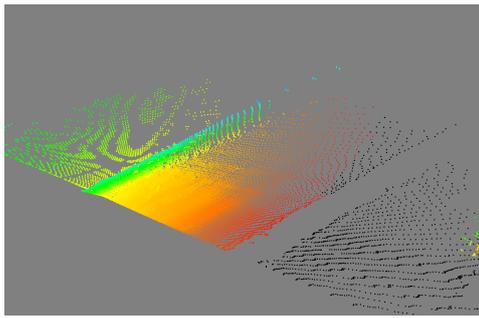


(g) Luftbild

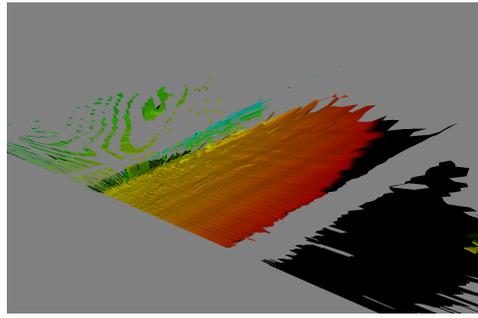


(h) Texturiertes Dreiecksnetz (Luftbildkamera)

Abbildung 5.37: **Szene E**: Ergebnisse der Kooperation Umweltmodellierung durch Sensorfusion.



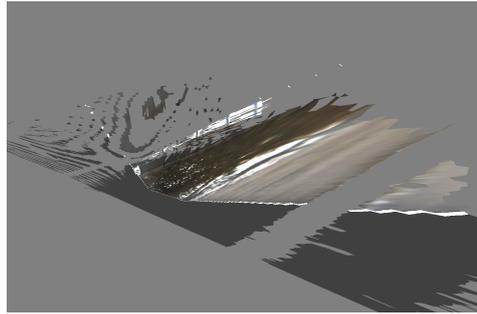
(a) Punktwolke



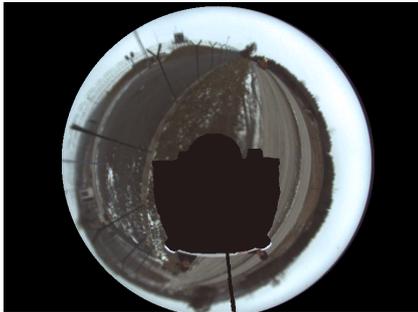
(b) Dreiecksnetz



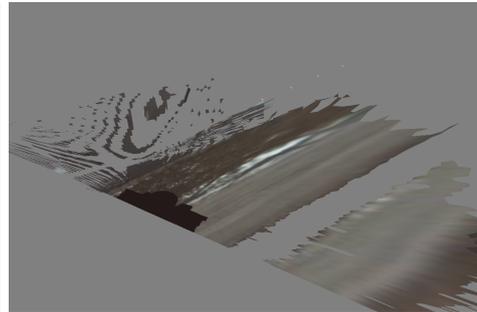
(c) Frontkamerabild



(d) Texturiertes Dreiecksnetz (Frontkamera)



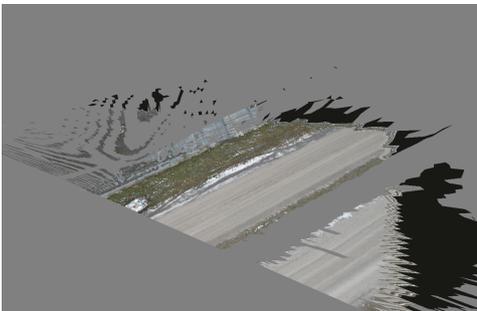
(e) Omnikamerabild



(f) Texturiertes Dreiecksnetz (Omnikamera)



(g) Luftbild



(h) Texturiertes Dreiecksnetz (Luftbildkamera)

Abbildung 5.38: **Szene F**: Ergebnisse der kooperativen Umweltmodellierung durch Sensorfusion.

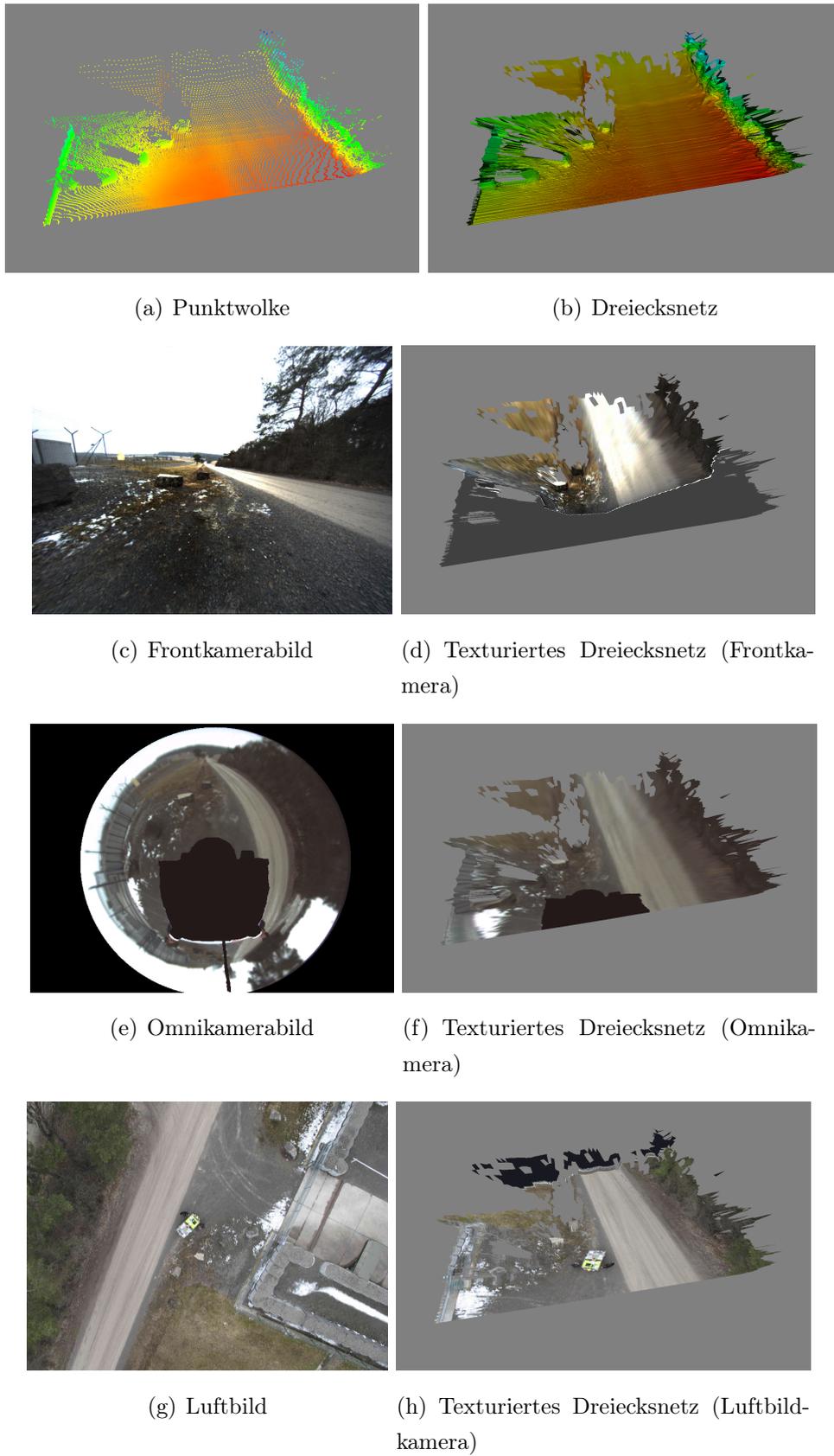
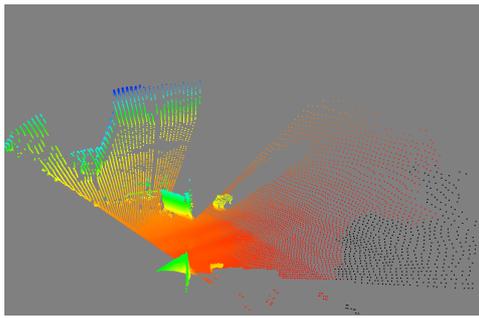
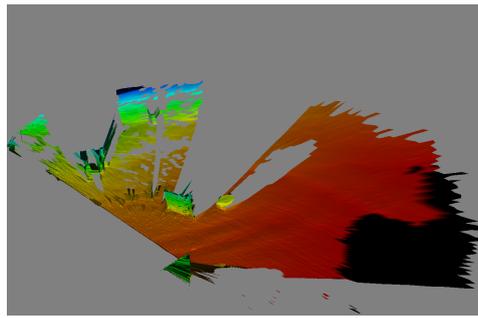


Abbildung 5.39: **Szene G**: Ergebnisse der Kooperation Umweltmodellierung durch Sensorfusion.



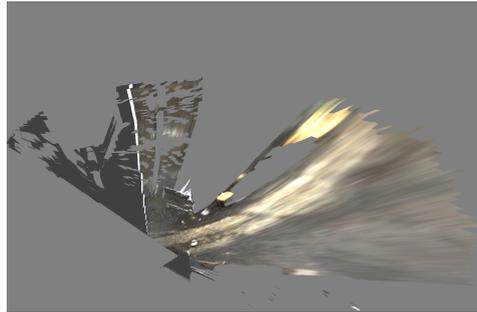
(a) Punktwolke



(b) Dreiecksnetz



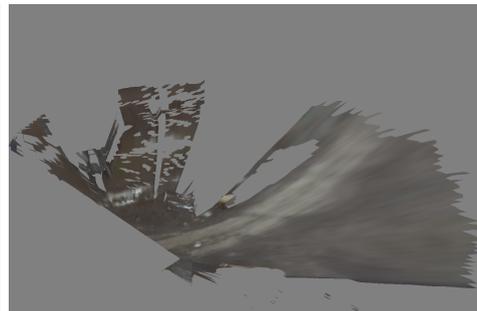
(c) Frontkamerabild



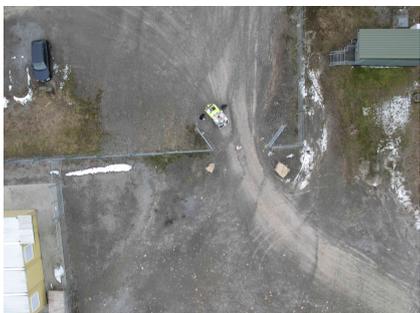
(d) Texturiertes Dreiecksnetz (Frontkamera)



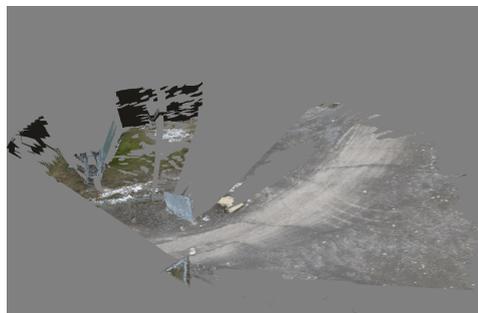
(e) Omnikamerabild



(f) Texturiertes Dreiecksnetz (Omnikamera)



(g) Luftbild



(h) Texturiertes Dreiecksnetz (Luftbildkamera)

Abbildung 5.40: **Szene H**: Ergebnisse der kooperativen Umweltmodellierung durch Sensorfusion.

# Kapitel 6

## Lokale explorative Pfadplanung

Die lokale Pfadplanung stellt neben der Lokalisierung und der Umgebungsmodellierung die dritte wichtige Säule der Realisierung eines autonom agierenden Roboters im unstrukturierten Außenbereich dar. Ohne diese zentrale Komponente wäre eine effiziente, kollisionsfreie Bewegung des Roboters nicht möglich. Da es eine Vielzahl von Herangehensweisen an das Thema der lokalen Pfadplanung gibt, sollen hier zunächst einige Grundanforderungen für die Implementierung der Pfadplanung in dieser Arbeit festgelegt werden, die vor allen Dingen aus der Analyse des Explorationsszenarios im Außenbereich entstanden sind:

1. Es sollen speziell Pfade generiert werden können, die den Roboter in bisher unbekanntes Terrain führen.
2. Im Gegensatz zu klassischen Start-Ziel-Planungsalgorithmen, sollen **alle** Pfade generiert werden, welche die vorherige Anforderung erfüllen.
3. Es sollen Pfade erzeugt werden, welche die Kinematik des Roboters korrekt berücksichtigen. Es wird von einem nicht-holonomen Roboter ausgegangen.
4. Die Repräsentation der Pfade soll eine nachträgliche Auswahl des unter den aktuellen lokalen Umständen besten Pfads ermöglichen.
5. Die Planungsmethode soll im unstrukturierten Außenbereich robust funktionieren.

6. Die Planungsmethode soll die Echtzeitanforderungen eines mobilen Außenbereichsroboters zu jeder Zeit erfüllen.

Auf Basis dieser Anforderungen zeigt sich bei der Untersuchung verwandter Arbeiten, dass es keine Ansätze gibt, die alle diese Anforderungen in einem integrierten Ansatz zufriedenstellend erfüllen, wie die folgende Auswertung des Stands der Technik im Bereich der Pfadplanung für mobile Roboter zeigt. Diese Lücke wird durch die in dieser Arbeit neu entwickelte explorative Pfadplanungsmethode geschlossen.

Die ersten Ansätze zur Pfadplanung von mobilen Robotern reduzieren das Problem durch Repräsentation des Roboters als punktförmigen Körper im Konfigurationsraum [131]. Der Roboter wird als holonom angenommen, d.h., er kann sich uneingeschränkt jederzeit in jede Richtung bewegen. Dies vereinfacht die Planungsbemühungen deutlich, da keinerlei differentielle Bedingungen bezüglich der Bewegung des Roboters in die Planung einbezogen werden müssen. Klassische Planungsverfahren für holonome, punktförmige Roboter teilen sich hauptsächlich in die Kategorien: Skelletierung (bspw. Sichtbarkeitsgraphen [9] und Voronoi-Diagramme [10]), Zellzerlegung [148] und Potentialfeldmethoden [142][151]. Da die Beschreibung des Roboters als holonomes Fahrzeug jedoch der zu Beginn des Kapitels festgestellten Grundanforderung 3 widerspricht, scheidet holonome Planungsverfahren für die Umsetzung der lokalen Pfadplanung in dieser Arbeit aus. Für einen ausführlichen Überblick über diverse Verfahren zur Planung von Pfaden für holonome Roboter sei an dieser Stelle auf [86] verwiesen.

In [118] formuliert Laumond zum ersten Mal das Problem der Pfadplanung für ein nicht-holonomes Fahrzeug. Dies basiert auf der Berücksichtigung von Differentialbedingungen während des Pfadplanungsprozesses. Die Differentialbedingungen können geometrischer oder sogar dynamischer Natur sein. Kinematische Modelle, die geometrische Differentialbedingungen modellieren, sind daher gebräuchlich, um das Pfadplanungsproblem für einen nicht-holonomen Roboter zu lösen [15].

In einer hindernisfreien Umgebung konnte bereits von Dubins [52] gezeigt werden, dass der kürzeste Pfad, der einen Start- mit einem Zielzustand verbindet, durch die Aneinanderreihung von Kurvensegmenten erzeugt werden kann. Dies gilt für ein vorwärtsfahrendes Fahrzeug mit Ackermann-Lenkung, das sich mit

gleichbleibender Geschwindigkeit bewegt. Dabei werden die Kurvensegmente aus einer endlichen Menge von sechs Kurvensegmenttypen gewählt. Reeds und Shepp [163] erweitern diesen Grundansatz und weisen nach, dass durch Zusammensetzung von maximal fünf Segmenten aus einer Menge von 48 unterschiedlichen Segmenttypen, die entweder gerade Linien oder Kurven mit minimalem Radius sind, ein minimaler Pfad für ein Fahrzeug erzeugt werden kann, das sich vorwärts und rückwärts bewegen kann und zusätzlich einen beschränkten Lenkwinkel besitzt. Sussmann und Tang [183] konnte diesen Ansatz leicht optimieren und die Basismenge der benötigten Kurvensegmente auf 46 reduzieren.

Die Berücksichtigung von Hindernissen in der Pfadplanung stellt den nächsten, naheliegenden Schritt dar. Auf Basis der Modellierung der Hindernisse in Form zweidimensionaler polygonaler Objekte haben Fortune und Wilfong in [60] nachgewiesen, dass die Fragestellung der Erreichbarkeit einer Zielkonfiguration für ein Lenkwinkel-beschränktes, punktförmiges Fahrzeug einer Lösung zugeführt werden kann. Auf Basis dieser Ergebnisse wurde in [89] ein Verfahren vorgestellt, das nicht nur die Existenz einer Lösung dieses Start-Ziel-Planungsproblems feststellt, sondern diese auch näherungsweise berechnet. Werden Hindernisse in der Umgebung des Roboters ausschließlich als konvexe Flächen mit differenzierbarer äußerer Kontur mit konstantem Krümmungsradius ("moderate obstacles" nach [1]) beschrieben, kann nach [24] eine exakte Lösung des Pfadplanungsproblems mit Hindernissen und beschränkten Kurvenradien in polynomialer Zeit ermittelt werden. Dieses Verfahren setzt zwar eine deutliche Einschränkung der darstellbaren Hindernisse voraus, diese Einschränkung ist jedoch in vielen Anwendungsszenarien eine praktikable Näherung der realen Hinderniskonfiguration.

Die Beschreibung des Fahrzeugs als rigiden Körper mit gewissen festen Ausmaßen anstatt als punktförmigen Körper stellt eine weitere Verbesserung des Modells dar. Laumond [118, 119] zeigt für diesen Fall, dass ein zweischrittiger Ansatz das zugehörige Pfadplanungsproblem lösen kann. Der erste Schritt besteht in der Planung eines Pfads unter der vereinfachenden Annahme, dass ein holonomes Fahrzeug zum Einsatz kommt. Im zweiten Schritt wird das Ergebnis des ersten Schritts in eine ausführbare Trajektorie transformiert, indem vom nicht-holonomen Fahrzeug ausführbare Manöver so verknüpft wer-

den, dass der im ersten Schritt generierte Pfad möglichst gut angenähert wird. Dreischrittige Ansätze wie [90] und [117] erweitern dies um einen weiteren Optimierungsschritt, der anhand unterschiedlicher Kriterien die Verbesserung von zuvor zwar grundsätzlich ausführbaren, jedoch eventuell nicht besonders effizienten Pfaden erreicht. Mehrschrittige Verfahren dieser Art erzeugen jedoch keine optimalen Lösungen, da die jeweiligen Optimierungsschritte nur lokale Anpassungen des geplanten Pfades durchführen und dabei die globale Optimalität des Pfades nicht berücksichtigen.

Der überwiegende Anteil der zuvor in diesem Abschnitt besprochenen Planungsalgorithmen ist jeweils nur auf eine spezifische Klasse von Fahrzeugen anwendbar. Barraquand und Latombe beschreiben im Gegensatz dazu in [15] einen Algorithmus, der als generischer Ansatz auf eine Vielzahl an Problemklassen anwendbar ist. Auf der Grundlage eines diskretisierten Konfigurationsraums wird ausgehend vom Zustand des Fahrzeugs zu Beginn der Planung ein Graph durch inkrementelles Wachstum aufgebaut. In jedem Iterationsschritt wird der Planungsgraph um jeweils einen Knoten erweitert. Jeder neue Knoten wird anhand einer Kostenfunktion so gewählt, dass stets der Knoten mit den niedrigsten Kosten hinzugefügt wird. Weiterhin werden neue Knoten durch Integration von Bewegungskommandos über einen festen Zeitraum, ausgehend von einem bereits bestehenden Knoten des Graphen, erzeugt. Erreicht ein Knoten des Graphen hinreichend nahe den vorher definierten Zielzustand, so ist das Pfadplanungsproblem gelöst. Dieser Algorithmus erzeugt näherungsweise optimale Pfade und ist asymptotisch optimal für automobilartige Fahrzeuge; abhängig von korrekt gewählter fester Integrationszeit der Bewegungskommandos und fester Zellengrößen zur Diskretisierung des Konfigurationsraums. Die Methode der dynamischen Programmierung stellt die Grundlage dieses Algorithmus dar und somit ist er nur für nieder-dimensionale Probleme anwendbar [19].

Um auch höher-dimensionale Probleme im Bereich der Pfadplanung bzw. Trajektoriengeneration lösen zu können und den sogenannten "Fluch der Dimensionalität"<sup>1</sup> zu überwinden, wurden randomisierte Algorithmen für die Pfadplanung eingeführt. Der erste Ansatz dieser Art stammt von Barraquand und Latombe [14] und entkommt lokalen Minima bei der Pfadplanung mit einem

---

<sup>1</sup>engl.: curse of dimensionality nach [19]

Potentialfeldansatz durch Ausführung von Zufallsbewegungen eines Fahrzeugs mit bis zu 31 Freiheitsgraden. In diesem Zusammenhang wurde in [114] die probabilistische Konvergenz nachgewiesen und eine Schätzung für die Zeit bis zur Konvergenz des Verfahrens ermittelt. Der grundlegende Nachteil der Nutzung von reinen Zufallsbewegungen ist die schlechte Vorhersagbarkeit der Zuverlässigkeit der daraus resultierenden Methoden. Probabilistic Roadmaps (PRM) [184] [93] [84] [27] konstruieren einen ungerichteten Graph durch Verbindung von benachbarten kollisionsfreien Zuständen im Konfigurationsraum mit einfachen (bspw. geradlinigen) Pfadsegmenten, die ebenfalls kollisionsfrei sind. Die Auswahl der zu verbindenden Zustände geschieht dabei randomisiert. Dieser Ansatz erzeugt deutlich verlässlichere Ergebnisse, allerdings gestaltet sich die Integration von Differentialbedingungen bei Verfahren dieser Art schwierig. Gleiches gilt für den von Yang und LaValle [213] eingeführten Random Neighborhood Graph (RNG), der durch die Definition einer globalen Navigationsfunktion entsteht, die auf einer überlappenden Aufteilung des Konfigurationsraums in kugelförmige Nachbarschaften basiert. Um Pfadplanungsprobleme unter differentiellen Einschränkungen auch in höher-dimensionalen Konfigurationsräumen realisieren zu können, wurde der Rapidly Exploring Random Tree (RRT) von LaValle [120] [122] eingeführt. Da diese Methode die Grundlage des in dieser Arbeit entwickelten Pfadplanungsverfahrens darstellt, wird im folgenden Kapitel detailliert auf diese Methode eingegangen.

## 6.1 Rapidly Exploring Random Trees

Um die nachfolgenden Erklärungen zu vereinfachen, werden zunächst die zentralen Begriffe der Pfadplanung mit Rapidly Exploring Random Trees definiert, wobei die Notation analog zur Arbeit von LaValle über Planungsalgorithmen [121] gewählt wurde:

### Zustandsraum $X$

$$X = \{x = (x_1, x_2, \dots, x_n)\} \subset \mathbb{R}^n$$

Eine  $n$ -dimensionale, abgeschlossene, beschränkte, differenzierbare Mannigfal-

tigkeit.

### Funktion der Gültigkeit der globalen Bedingungen $D$

$$D : X \rightarrow \{1, 0\}$$

Die Funktion  $D$  ermittelt, ob ein Zustand  $x$  bez. der global definierten Bedingungen des spezifischen Pfadplanungsproblems gültig oder ungültig ist. Mit der Funktion  $D$  können die Subräume  $X_f$  und  $X_h$  definiert werden.  $X_f = \{x | x \in X, D(x) = 1\}$  ist der Subraum der gültigen kollisionsfreien Zustände.  $X_h = \{x | x \in X, D(x) = 0\}$  ist der Subraum der ungültigen kollidierenden Zustände.

### Grenzbedingungen $x_s, X_z$

$$x_s \in X_f, X_z \subset X_f$$

$x_s$  ist der Startzustand zu Beginn der Planung und  $X_z$  definiert die Zielregion (Teilmenge kollisionsfreier Zielzustände), die durch den geplanten Pfad erreicht werden soll. Wobei nur ein Element von  $X_z$  als Zielzustand erreicht werden muss, um einen Planungsdurchlauf als erfolgreich zu klassifizieren.

### Endliche Menge von Inputs $U$

$$U = \{u = (u_1, u_2, \dots, u_m)\} \subset \mathbb{R}^m$$

$U$  beschreibt die Menge der möglichen Inputs, die einem Fahrzeug kommandiert werden können. Die Elemente von  $U$  sind stets unabhängig von  $x$  und durch die Grenzinputs

$$u^{min} = (u_1^{min}, u_2^{min}, \dots, u_m^{min}) \in U \text{ und}$$

$$u^{max} = (u_1^{max}, u_2^{max}, \dots, u_m^{max}) \in U$$

komponentenweise beschränkt.

### Zustandsübergangsfunktion $f$

$$f : X \times U \times \mathbb{R} \rightarrow X, x_{new} = f(x, u, t)$$

Die Zustandsübergangsfunktion  $f$  definiert die Propagation eines Zustands  $x$  über den Zeitraum  $t$  durch Integration des Inputs  $u$ .

### Metrik $\Xi$

$$\Xi : X \times X \rightarrow [0, \infty)$$

Die Metrik  $\Xi$  definiert den Abstand zweier Zustände im Zustandsraum und wird dazu genutzt, um zu berechnen, welcher Zustand einem anderen Zustand am nächsten ist. Es ist wichtig, an dieser Stelle festzustellen, dass die Wahl der richtigen Metrik entscheidenden Einfluss [39] auf die Qualität des Planungsergebnisses hat und die Wahl einer einfachen Metrik wie der euklidischen Abstandsmetrik gerade bei nicht-holonomen Planungsproblemen keine akzeptablen Ergebnisse erzeugt.

### Feste Propagationsperiode $\Delta t$

$$\Delta t \in \mathbb{R}, \Delta t > 0$$

$\Delta t$  definiert die feste Zeitperiode, in welcher der Input  $u \in U$  angewendet wird, um den Zustand  $x$  mit der Zustandsübergangsfunktion  $f$  fortzuschreiben.

### Feste Lösungstoleranz $\tau$

$$\tau \in \mathbb{R}, \tau > 0$$

$\tau$  definiert den für eine als erfolgreich klassifizierte Planung maximal zulässigen Abstand zwischen dem Endzustand des geplanten Pfads und der Zielregion  $X_z$ .

Das Ziel der Pfadplanung besteht nun in der Bestimmung einer stückweise konstanten Kontrollfunktion  $\hat{u} : [t_0, t_L] \rightarrow U$ , die dem Intervall der Pfaderweiterungszeitpunkte  $[t_0, t_L]$  jeweils einen festen Input zuordnet, der bis zum darauf folgenden Pfaderweiterungszeitpunkt angewendet wird. Aus der

sequenziellen Anwendung der stückweise konstanten Inputs entsteht die Lösung  $\pi : [t_0, t_L] \rightarrow X_f$  des Pfadplanungsproblems, die einen aus kollisionsfreien Zuständen zusammengesetzten Pfad darstellt. Für  $\pi$  gelten folgende weitere Bedingungen:

$$\hat{u}(t) \in U, \forall t \in [t_0 + k\Delta t, t_0 + (k+1)\Delta t), k = 0, 1, \dots, K-1$$

$$\pi(t) \in X_f, t \in [t_0, t_L],$$

$$\pi(t) = \begin{cases} x_s, & \text{wenn } t = t_0. \\ \pi(t_0 + k\Delta t) + \int_{t_0+k\Delta t}^t f(x, u, t), & \text{wenn } [t_0 + k\Delta t, t_0 + (k+1)\Delta t), \\ & k = 0, 1, \dots, K-1. \end{cases}$$

$$\exists x_z \in X_z, \Xi(\pi(t_L), x_z) \leq \tau$$

Aufbauend auf dieser grundlegenden Definition der inkrementellen Pfadplanung bieten die Rapidly Exploring Random Trees (RRTs) eine Methode zur Organisation und Durchführung des Planungsprozesses, sodass eine effiziente Exploration von hindernisbehafteten Konfigurationsräumen auch mit nicht-holonomen Fahrzeugen möglich wird, selbst wenn diese eine hohe Anzahl an Freiheitsgraden besitzen.

Die zentrale Datenstruktur bei der Pfadplanung mit einem RRT ist der gerichtete Planungsbaum

$$G_{RRT} = (V_{RRT}, E_{RRT}, x_s). \quad (6.1)$$

$V_{RRT} = \{x^i = (p_x^i p_y^i \phi^i)^T \in X | i = 1, 2, \dots, N_{V_{RRT}}\}$  bezeichnen die Knoten des Baums, die Zuständen im Konfigurationsraum entsprechen.  $E_{RRT} = \{e_j = (x_{e_j}^1, x_{e_j}^2, u_j)^T | (x_{e_j}^1, x_{e_j}^2) \in [V_{RRT}]^2, j = 1, 2, \dots, N_{E_{RRT}}\}$  definieren die Kanten, welche die Verbindungen zwischen Knoten des Baums festlegen.  $x_s$  ist der Wurzelknoten des Planungsbaums  $G_{RRT}$ . Gültige Pfade werden konstruiert, indem man durch Kanten verbundene Knoten in gültiger Richtung aneinanderreicht. Man geht dabei von den Wurzelknoten des Baums aus. Der Planungsbaum selbst wird durch den folgenden Algorithmus aufgebaut (siehe [122]):

---

**Algorithmus 2** Aufbau eines RRT
 

---

```

 $G_{RRT}.init(x_s);$ 
for  $k \leftarrow 1$  to  $K$  do
   $x_{rand} \leftarrow RANDOM\_STATE();$ 
   $x_{near} \leftarrow NEAREST\_NEIGHBOR(x_{rand}, G_{RRT});$ 
   $u \leftarrow SELECT\_INPUT(x_{rand}, x_{near});$ 
  if  $NEW\_STATE(x_{rand}, x_{near}, x_{new}, u_{new})$  then
     $G_{RRT}.add\_vertex(x_{new});$ 
     $G_{RRT}.add\_edge(x_{near}, x_{new}, u_{new});$ 
  end
end

```

---

Der Baum wird zunächst mit dem Startknoten  $x_s$  als Wurzel initialisiert. Ausgehend davon wird der Baum mit der nun folgenden, stets gleichen Methode um maximal einen Knoten pro Schritt erweitert. Zuerst wird mit der Funktion  $RANDOM\_STATE$  ein zufälliger Zustand  $x_{rand}$  aus dem Zustandsraum  $X$  gezogen. Danach wird der dem zufälligen Zustand  $x_{rand}$  am nächsten liegende Zustand  $x_{near}$  gesucht, wobei dabei in der Funktion  $NEAREST\_NEIGHBOUR$  die zuvor beschriebene Metrik  $\xi$  zur Abstandsbestimmung verglichener Zustände eingesetzt wird. Die Funktion  $NEW\_STATE$  bestimmt daraufhin durch Anwendung aller Inputs  $u \in U$  auf den Zustand  $x_{near}$  über die feste Propagationsperiode  $\Delta t$  und gleichzeitiger Prüfung auf Kollisionsfreiheit mithilfe der zuvor beschriebenen Funktion  $D$  den Zustand  $x_{new}$ , der dem Zustand  $x_{rand}$  am nächsten liegt. Konnte ein Input ermittelt werden, der eine gültige Erweiterung des Baums in Richtung des Zustands  $x_{rand}$  erlaubt, wird der Baum um den Knoten  $x_{new}$  und die Kante  $(x_{near}, x_{new}, u_{new})$  erweitert. Dieser Vorgang wird solange wiederholt, bis die feste Anzahl an Baumerweiterungsschritten  $K$  durchgeführt worden ist. Abbildung 6.1 zeigt zur Verdeutlichung des Ablaufs des Aufbaus eines RRTs einen exemplarischen Erweiterungsschritt.

Die wichtigste Eigenschaft von RRTs, die diese gleichzeitig für die Nutzung im Rahmen der explorativen Pfadplanung prädestiniert, ist die klare Tendenz des Baumwachstums in Richtung bisher unerkundeter Bereiche des Zustandsraums. Dies unterscheidet die RRTs klar von Methoden, die ein naives zufälliges Baumwachstum im Sinne einer zufälligen Wahl von zu erweiterndem Knoten

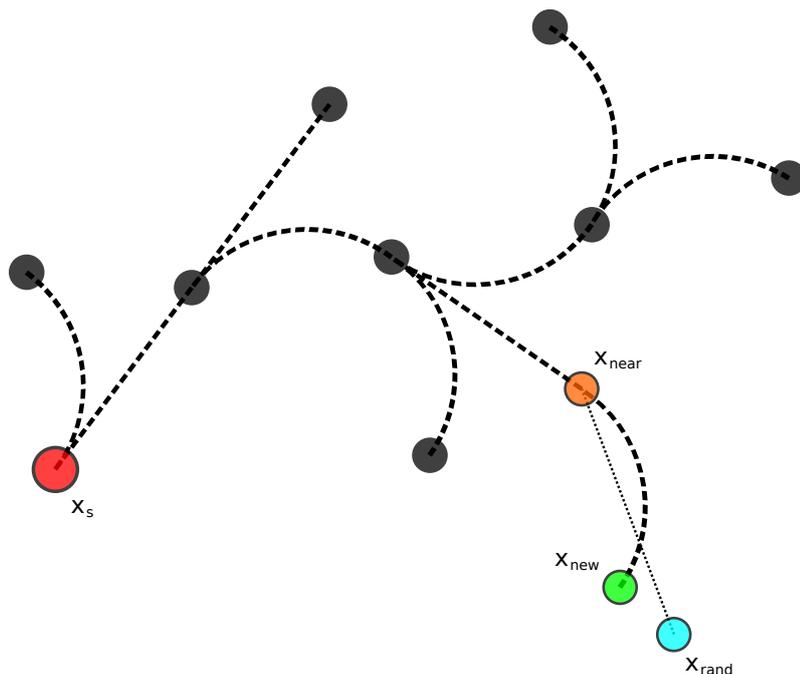


Abbildung 6.1: **Erweiterungsschritt beim Aufbau eines RRTs.**

und anzuwendenden Inputs umsetzen. Wie Abb. 6.2 klar zeigt, konzentrieren sich die Knoten im Fall naiven, zufälligen Wachstums auf bereits erkundete Bereiche. Im Gegensatz dazu sorgt RRT-basiertes Baumwachstum für ein schnelles Erkunden unbekannter Regionen des Zustandsraums.

Werden die Positionen der Knoten des Planungsbaums als 2D-Punktwolke betrachtet, kann auf Basis dieser ein Voronoi-Diagramm erzeugt werden (siehe Abb. 6.3). Dieses Voronoi-Diagramm ermöglicht die Erklärung des äußerst wünschenswerten Explorationsverhaltens der RRTs. Aufgrund der Tatsache, dass stets der Knoten erweitert wird, der dem in diesem Schritt zufällig gezogenen Knoten am nächsten ist, gilt gleichzeitig, dass immer der Knoten erweitert wird, in dessen Voronoi-Region der zufällig gezogene Knoten liegt. Je größer die entsprechende Voronoi-Region ist, desto höher ist die Wahrscheinlichkeit, dass ein zufällig gezogener Knoten in dieser Region liegt. Gleichzeitig bedeutet die Erweiterung derjenigen Knoten, die in großen Voronoi-Regionen liegen, eine effiziente Erkundung der unbekannteren Teile des Zustandsraums, da eine große Voronoi-Region gleichbedeutend mit einem großen Teil des noch nicht erkundeten Zustandsraums ist. Dieser Effekt kann durch die Integration verschiedener Heuristiken weiter verstärkt werden.

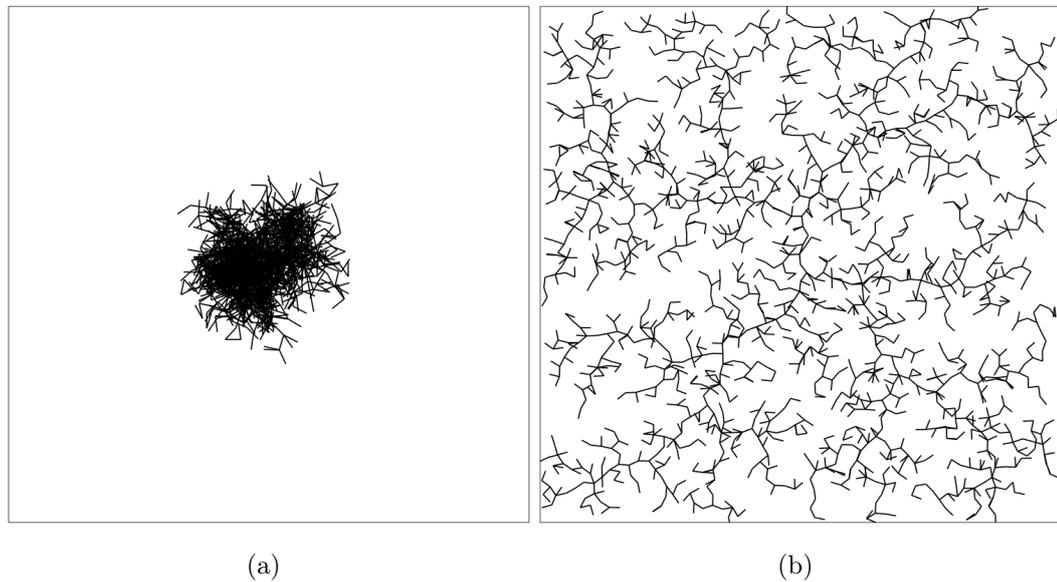


Abbildung 6.2: **Vergleich verschiedener Methoden des zufälligen Baumwachstums.** Naives zufälliges Wachstum (a). RRT-basiertes zufälliges Wachstum (b). Beide Bäume enthalten 2000 Knoten. Quelle: [122]

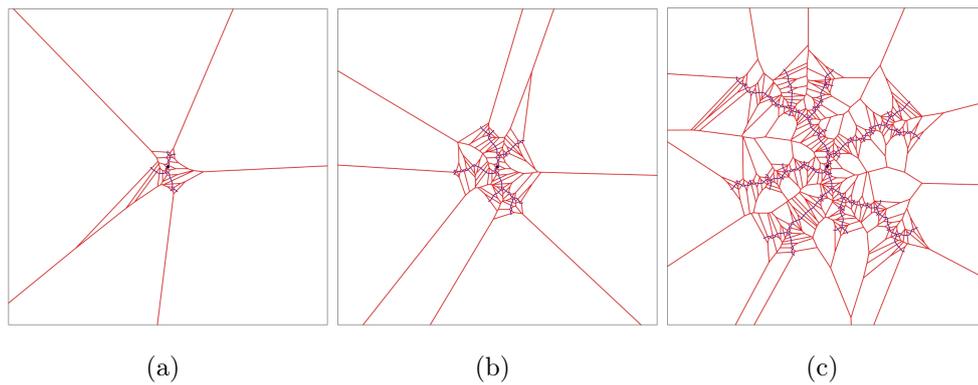


Abbildung 6.3: **Voronoi-Diagramme der Knoten des RRTs während unterschiedlicher Erweiterungsstadien des Planungsbaums.** Quelle: [122]

Da die Methode der Rapidly Exploring Random Trees als generische Methode der Pfadplanung eine Vielzahl an Ausgestaltungen erlaubt, wird in den folgenden Unterkapiteln die in dieser Arbeit entwickelte Methode der Pfadplanung beschrieben, die zwar auf der RRT-Methode aufbaut, jedoch durch verschiedene Erweiterungen und Anpassungen speziell für das Explorationsszenario in der Außenbereichsrobotik mit kooperierendem Boden- und Luftroboter ausgestaltet und optimiert wurde.

## 6.2 Lokale Hinderniskarte

Die Prüfung von einzelnen Elementen des Zustandsraums auf Gültigkeit bezüglich der global definierten Bedingungen, die von der bereits zuvor beschriebenen Funktion  $D$  modelliert werden, ist eine zentrale Aufgabe während des Wachstums des Pfadplanungsbaums. Der Hauptbestandteil dieser Aufgabe besteht in der Feststellung der Kollisionsfreiheit des aktuell betrachteten Zustands. Dies geschieht in dieser Arbeit auf der einen Seite auf Basis einer zweidimensionalen Gitterkarte mit annotierten Höheninformationen und Hindernisklassifizierung je äquidistanter Gitterzelle, die aus der schritthaltend erzeugten, dreidimensionalen Punktwolke der Umgebung (siehe Kapitel 5.2) erzeugt wird. Auf der anderen Seite werden erneut die Daten des kooperierenden Flugroboters eingesetzt, um in diesem Fall eine zusätzliche Hindernisdetektion im Umfeld des Bodenroboters durch eine visuelle Analyse der Umgebung durchführen zu können. Dadurch wird es möglich, Untergründe verschiedener Kategorien zu unterscheiden (z.B. befestigte und unbefestigte Straße), die sich zuvor durch die Analyse von rein geometrischen Umgebungsdaten nicht differenzieren ließen. Ein effizienteres Verhalten der entwickelten Pfadplanungsmethoden ist die Folge dieser Herangehensweise.

### 6.2.1 Geometrische Hindernisdetektion

Die in Kapitel 5.2 beschriebenen Methoden zur geometrischen Umgebungsmodellierung erzeugen eine Punktwolke, die ein Bezugskordinatensystem besitzt, dessen z-Achse parallel zur Flächennormalen der lokal angenäherten Bodene-

bene verläuft. Die Punktwolke setzt sich aus den Punkten  $P_k \in \mathbb{R}^3$  zusammen. Die gesuchte Gitterkarte wird durch die Matrix  $\Gamma^{m \times m} = (g_{ij})$  beschrieben. Die Elemente der Matrix  $g_{ij}$  werden als Zellen des Gitters bezeichnet. Mit der Funktion  $\zeta : (i, j) = g_{ij}$  wird einem Gitterkoordinatentupel  $(i, j) \in \mathbb{N}^2$  ein Gitterzellenwert  $g_{ij}$  zugeordnet, der im Fall der Nutzung der Gitterkarte als kombinierte Höhen- und Hinderniskarte einem Tupel  $g_{ij} = (h_{ij}, c_{ij})$  entspricht. Das Tupel setzt sich aus Höhenwert  $h_{ij}$  und Gitterzellenklassifikation  $c_{ij}$  zusammen. Der Höhenwert definiert die Höhe einer Gitterzelle über der Bodenebene, die durch den Koordinatenursprung der aktuellen Szene festgelegt ist. Die Menge der Gitterzellenklassifikationen ist definiert durch  $C = \{\text{FREI, BELEGT, UNBEKANNT, INTERPOLIERT, HINDERNIS}\}$ . Die Implementierung der Funktion  $\zeta$  umfasst verschiedene Schritte und geschieht durch den folgenden Algorithmus 3, der als Hilfsdatenstruktur eine geordnete Menge  $G_{ij}$  je Gitterzelle  $g_{ij}$  nutzt, um darin während des Aufbaus der Karte  $\Gamma$  den Gitterzellen zugeordnete Punkte abzulegen.

Die Gitterkarte  $\Gamma$  hat lokal begrenzte Ausmaße. Falls ein Punkt  $P_k$  nicht innerhalb der lokalen Grenzen der Karte liegt, wird er direkt verworfen. Dies wird mit der Funktion *OUT\_OF\_BOUNDS* geprüft. Für Punkte, die innerhalb der lokalen Grenzen liegen, werden durch den Ausdruck

$$\text{w2g}(i, j) = \begin{pmatrix} i \\ j \end{pmatrix} = \left\lfloor \begin{pmatrix} x - x_{min} \\ y - y_{min} \end{pmatrix} r^{-1} \right\rfloor \quad (6.2)$$

aus dem Punkt  $P_k = (x \ y \ z)^T \in \mathbb{R}^3$  die Gitterkoordinaten  $(i \ j)^T \in \mathbb{N}^2$  errechnet, wobei  $x_{min}$  und  $y_{min}$  die minimal gültigen lokalen Punktkoordinaten in  $x$ - bzw.  $y$ -Richtung und  $r$  die Seitenlänge einer Gitterzelle in Weltkoordinaten bezeichnen (*GET\_GRID\_COORDINATES*). Mit den ermittelten Gitterzellenkoordinaten werden die Punkte den entsprechenden Gitterzellen zugeordnet (*append*) und abhängig von der  $z$ -Koordinate des jeweiligen hinzugefügten Punkts sortiert (*sort\_by\_z*). Diese entspricht der Höhe des Punktes über bzw. unter der Bodenebene. Die minimale Höhe jeder Gitterzelle wird nun genutzt, um weitere invalide Punkte in jeder Gitterzelle vor der Ausführung der folgenden Verarbeitungsschritte zu eliminieren. Dies geschieht durch einen Schwellwertvergleich der Differenz der Höhe des analysierten Punktes und

---

**Algorithmus 3** Konstruktion der lokalen geometrischen Höhenkarte
 

---

```

Γ.init();
for k ← 1 to K do
  | if ¬ OUT_OF_BOUNDS(Pk) then
  |   | (i, j) ← GET_GRID_COORDINATES(Pk);
  |   | Gij.append(Pk);
  |   end
end
end
for i ← 1 to Ni do
  | for i ← 1 to Nj do
  |   | Gij.sort_by_z();
  |   | Gij.remove_invalid_local_z();
  |   | gij ← MAX_Z(Gij);
  |   end
end
end
for i ← 1 to Ni do
  | for i ← 1 to Nj do
  |   | if Gij.empty() then
  |   |   | gij ← INTERPOLATE_NN();
  |   |   end
  |   end
end
end
end

```

---

der minimalen Höhe in dieser Zelle (*remove\_invalid\_local\_z*). So können Überdachungen oder überhängende Objekte ignoriert werden, die vom Bodenroboter nicht erreicht werden können und damit keine Hindernisse für diesen darstellen. Abschließend wird die jeweils maximale Höhe jeder Zelle als deren Höheannotation für die folgenden Schritte eingetragen.

An dieser Stelle kann bereits eine erste Klassifikation der Gitterzellen der Gitterkarte durchgeführt werden: in Gitterzellen, die mindestens einen zugeordneten Punkt enthalten, also BELEGT sind, und Gitterzellen, die keine Punkte enthalten und daher als UNBEKANNT klassifiziert werden. Kleinere Flächen unbekannter Gitterzellen können durch die Anwendung von Interpolationsmethoden mithilfe belegter benachbarter Zellen ermittelt werden. Die Nächster-Nachbar-Interpolation ist in diesem Zusammenhang eine einfache, aber effektive Form der Interpolation, die bei moderater Anwendung eine gute Näherung des realen Fahrzeugumfelds erzeugt (*INTERPOLATE\_NN*). Interpolierte Zellen werden der Kategorie INTERPOLIERT zugeordnet.

Die Unterscheidung der belegten Zellen in die Klassen FREI und HINDERNIS geschieht während des folgenden Schritts des Aufbaus der Hinderniskarte aus der interpolierten Höhenkarte. Dabei wird nur der Teil der Funktion  $\zeta : (i, j) = g_{ij}$  betrachtet, der die Höheninformationen  $h_{ij}$  der zugehörigen Gitterzelle  $g_{ij}$  erzeugt. Die zugehörige Teilfunktion ist als  $\eta : (i, j) = h_{ij}$  definiert. Um zu ermitteln, ob eine Gitterzelle ein Hindernis beinhaltet oder frei von Hindernissen ist, wird der Inter-Zellenhöhengradient

$$\nabla\eta(i, j) = \begin{pmatrix} \frac{\partial\eta}{\partial i} \\ \frac{\partial\eta}{\partial j} \end{pmatrix} \quad (6.3)$$

bzw. dessen Betrag  $\|\nabla\eta(i, j)\|$  betrachtet. Durch Vergleich mit einem festen, empirisch ermittelten Schwellwert, der abhängig von den Eigenschaften des eingesetzten Fahrzeugs ist, wird entschieden, ob die betrachtete Gitterzelle als hindernisbehaftet markiert wird. Die grundlegende Idee hinter dieser Vorgehensweise ist, dass ein Fahrzeug nur eine bestimmte lokale Steigung bewältigen kann, die durch den Betrag des Höhengradienten lokal angenähert wird.

Abbildung 6.4 zeigt zusammenfassend den Entstehungsprozess einer Hindernisgitterkarte aus einer beispielhaften Punktwolke.



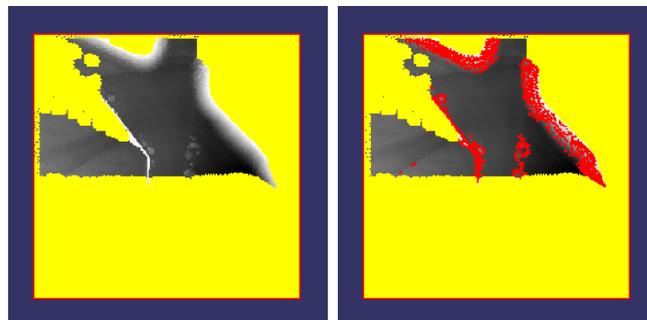
(a)



(b)

(c)

(d)



(e)

(f)

Abbildung 6.4: **Erzeugung einer Hindernisgitterkarte aus einer lokalen Punktwolke.** Videobild der analysierten Szene (a). Punktwolke mit anhand deren jeweiliger Höhe eingefärbten Punkten und lokale Begrenzungen (roter Rahmen) (b). Aus der Punktwolke entstandene Gitterkarte mit Gitterzellen der Kategorien BELEGT (grün), UNBEKANNT (gelb) und INTERPOLIERT (hellbraun) (c) und (d). Interpolierte Gitterkarte mit anhand deren jeweiliger Höhe eingefärbten Gitterzellen (e). Interpolierte Gitterkarte mit detektierten Hinderniszellen (rot) (f).

## 6.2.2 Fahrkorridorerkennung durch Luftbilddauswertung

Die im vorhergehenden Kapitel beschriebene, rein geometrische Form der Hinderniserkennung im direkten Umfeld des Roboters ist in der Lage, im überwiegenden Teil der Situationen im Außenbereichsszenario eine robuste Modellierung bzw. Interpretation der Umgebung zu erzeugen. So wird eine sichere, kollisionsfreie Operation des Roboters gewährleistet. Nichtsdestotrotz gibt es Szenen, in denen sich verschiedene unterschiedliche Bereiche des geometrischen Umgebungsmodells nicht auf rein geometrischer Basis differenzieren lassen. Die Unterscheidung verschiedener Untergrundtypen (siehe Abb. 6.5) ist neben der Klassifikation flexibler Hindernisse bezüglich deren Durchfahrbarkeit [110] ein wichtiges Praxisbeispiel für diesen Problembereich. Zwar führt eine rein geometrische Detektion von Hindernissen nicht zu einer Gefährdung des Roboters, jedoch profitiert gerade die lokale, explorative Pfadplanung deutlich von einer differenzierteren Klassifikation des lokalen Umfelds des Bodenroboters. In vielen Fällen kann so deutlich zielgerichteter und effizienter geplant werden, da weniger lohnenswerte Bereiche im Umfeld des Roboters nicht erkundet werden müssen. Aufgrund der entstehenden Kontinuität des Untergrundtyps, auf dem sich der Roboter bewegt, führen die daraus resultierenden Pfade zu einem zusätzlich optimierten Verhalten des Roboters, während er diese umsetzt.

Auf Basis dieser Vorüberlegungen wurde im Rahmen dieser Arbeit eine Methode entwickelt, die durch Analyse der registrierten Luftbilddaten eine Detektion des befahrbaren Pfades umsetzt, die der kooperierende Flugroboter schritthaltend zur Verfügung stellt. Im Vergleich zu bestehenden Methoden, die bodengebundene Kameras nutzen (siehe bspw. [62] oder [188]), bedeutet die Änderung der Kameraperspektive auf eine Bildebene, die näherungsweise parallel zur beobachteten Bodenebene liegt, einen deutlich geringeren Einfluss des verzerrenden Effekts der perspektivischen Projektion der Kameraabbildung. Zusätzlich liegt die Position der Kamera deutlich oberhalb der Bodenebene, was eine gute Übersicht und damit eine gute Vorausschau garantiert (vgl. Abb. 6.5).

In Kapitel 5.4 (siehe Gleichung 5.91 bis 5.96) wurde bereits ausführlich beschrieben, wie die Abbildung ermittelt werden kann, die einen Punkt  $P = (x \ y \ z)^T$  des geometrischen Modells des Bodenroboters auf das Pixelkoordi-

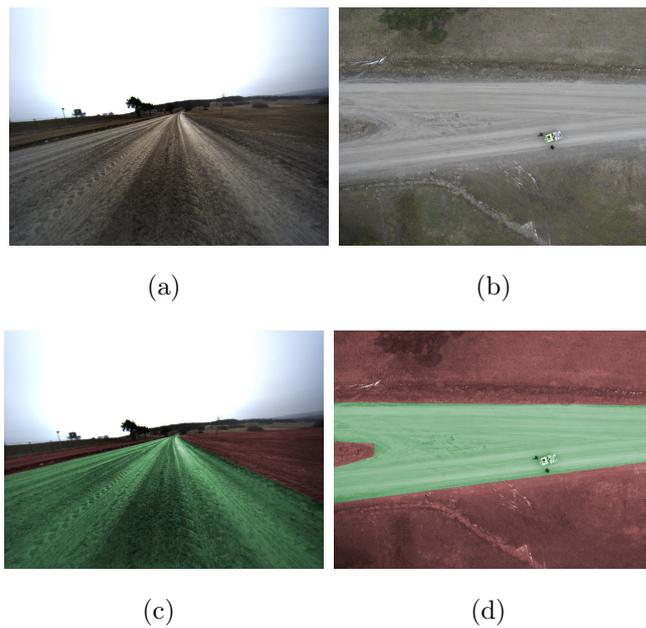


Abbildung 6.5: **Unterschiedlichen Untergrundtypen in einer Außenbereichsszene, die aufgrund geringer lokaler Höhenunterschiede nicht auf Basis geometrischer Merkmale unterscheidbar sind.** Frontkamera (a) und (c). Luftbildkamera (b) und (d).

natentupel  $(u \ v)^T$  des aktuellen Luftbilds der Luftbildkamera abbildet. Mit Gleichung 6.3 ist weiterhin bekannt, wie Punkte des geometrischen Modells auf Gitterzellen der lokalen Hinderniskarte abgebildet werden können. Da somit die Abbildung zwischen Zellen der lokalen Hindernisgitterkarte und visuellen Daten der Luftbildkamera bekannt ist, kann die Aufgabe der visuellen Hindernisklassifikation der Zellen der lokalen Hindernisgitterkarte durch eine klassische digitale Bildverarbeitungsaufgabe beschrieben werden, die auf den Luftbilddaten durchgeführt werden kann. Konkret soll das Zwei-Klassen-Klassifikationsproblem gelöst werden, das jedes Pixel des Luftbilds entweder der Klasse BEFAHREN oder der Klasse MEIDEN zuordnet und durch die Klassifikationsfunktion

$$\bar{R} : \mathbb{R}^2 \rightarrow \bar{C}, (i \ j)^T = \bar{c} \text{ mit } \bar{c} \in \bar{C} \quad (6.4)$$

definiert ist, wobei  $\bar{C} = \{\text{BEFAHREN}, \text{MEIDEN}\}$  gilt. Die fusionierte Hindernisklassifikation  $c'_{ij} \in C$ , die geometrisch basierte und visuell basierte Klassifi-

kation vereint, wird schließlich durch die Funktion

$$R : C \times \bar{C} \rightarrow C, (c_{ij}, \bar{c}_{ij}) = c'_{ij} \text{ mit } c'_{ij} \in C \quad (6.5)$$

$$c'_{ij} = \begin{cases} \text{HINDERNIS,} & \text{wenn } c_{ij} \neq \text{UNBEKANNT} \wedge \bar{c}_{ij} = \text{MEIDEN} \\ c_{ij}, & \text{sonst.} \end{cases} \quad (6.6)$$

beschrieben.

Zur Umsetzung der Funktion  $\bar{R}$  werden zwei unterschiedliche Segmentierungsmethoden implementiert. Auf der einen Seite ein Regionenwachstumsverfahren, das durch inkrementelles lokales Wachstum, ausgehend von einem Saatpunktpixel, eine zusammenhängende Region mit ähnlichen visuellen Eigenschaften markiert [156]. Auf der anderen Seite ein pixelbasiertes Verfahren, das für jedes Pixel die Ähnlichkeit zu einer zu Beginn festgelegten Region-Of-Interest ermittelt. Das Ähnlichkeitsmaß je Pixel wird durch Rückprojektion des Farbwert-Histogramms in das Ausgangsbild ermittelt, das auf Basis der Region-Of-Interest bestimmt wurde [186]. Wie in Abbildung 6.6 (a) beispielhaft zu sehen ist, wird der Saatpunkt des inkrementellen Regionenwachstumsverfahrens stets abhängig von der detektierten Pose des Bodenroboters im Bild so gewählt, dass er kurz vor dem Bodenroboter in dessen Fahrtrichtung positioniert ist. Der Saatpunkt stellt gleichzeitig den Mittelpunkt eines orientierten Rechtecks dar, dessen Ausrichtung ebenfalls mit der Pose des Bodenroboters bestimmt wird und das die Grundlage für die Histogrammberechnung für die zweite Methode darstellt. Die Ergebnisse beider Ansätze werden mit der bereits aus den Kapiteln 4.4.2 und 5.4.1 bekannten Methode nachbearbeitet, die zusammengefasst eine Tiefpassfilterung mit anschließender Binarisierung durch Schwellwertvergleich realisiert.

Abbildung 6.6 stellt die Ergebnisse beider Methoden in einer Beispielszene gegenüber. Aufgrund einer Analyse der Ergebnisse beider Methoden auf mehreren Testsequenzen wurde das Regionenwachstumsverfahren der Histogramm-Rückprojektion vorgezogen, da es im Mittel eine zuverlässigere Detektion des befahrenen Wegs in den getesteten Sequenzen ermöglicht.

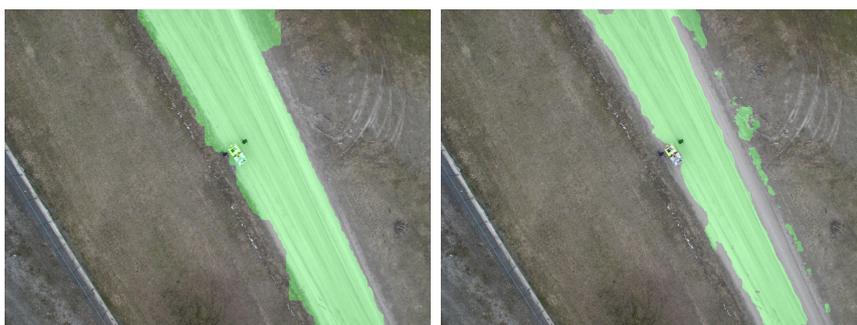


(a)



(b)

(c)



(d)

(e)

Abbildung 6.6: **Segmentierungsergebnisse der visuellen Wegdetektion im Live-Luftbild.** Inkrementelles Regionenwachstum (b) und (d) bzw. Histogramm-Rückprojektion (c) und (e) mit anschließender tiefpassgefilterter Binarisierung.

Es bleibt zu betonen, dass die Segmentierung des befahrbaren Wegs im aktuellen Luftbild der Kamera des Flugroboters kein zentrales Forschungsthema dieser Arbeit darstellt und daher auf bekannte, wenn auch nicht in besonderem Maße auf das spezielle Anwendungsszenario angepasste Methoden zurückgegriffen wurde. Es soll vielmehr der Nachweis der Nutzbarkeit von Sensordaten eines kooperierenden Flugroboters in der Pfadplanung eines autonomen Bodenroboters erbracht werden.

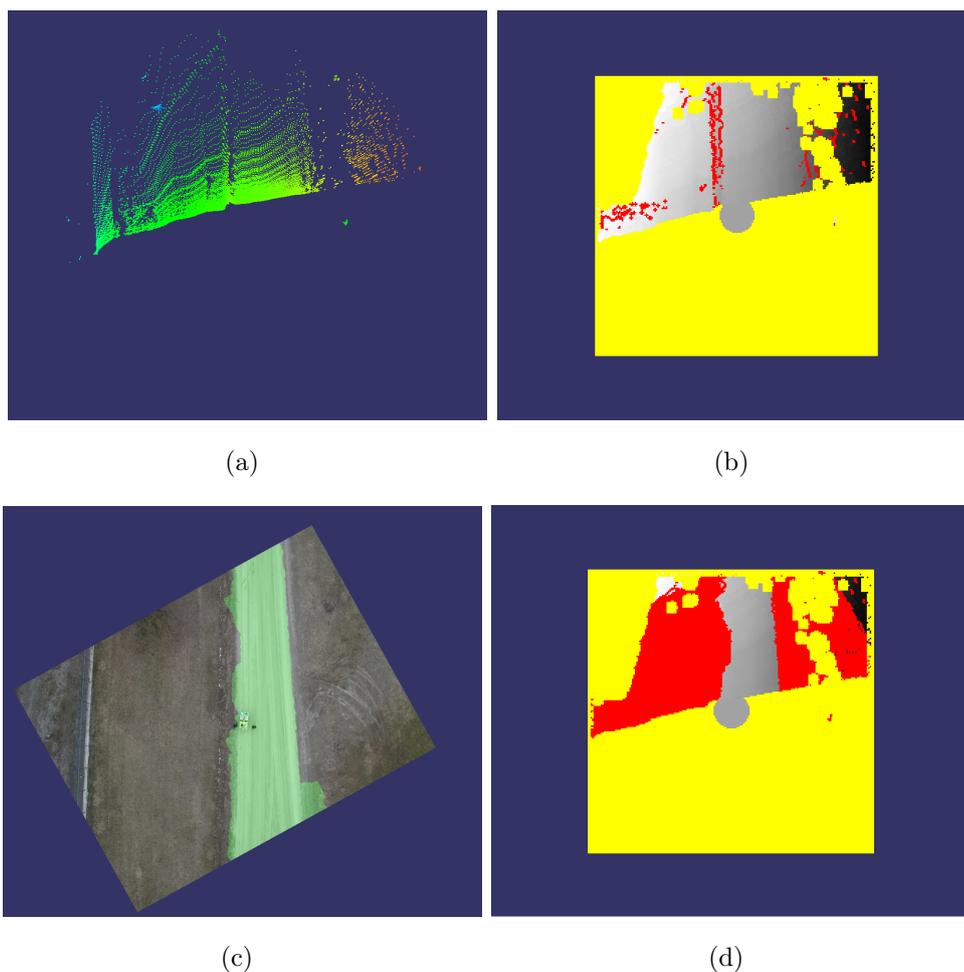


Abbildung 6.7: **Fusion der Ergebnisse aus geometrischer und visueller Hindernisdetektion.** Durch den Bodenroboter erzeugte Punktwolke (a). Aus der Punktwolke erzeugte Hindernisgitterkarte (b). Segmentiertes, registriertes Luftbild (c). Fusionierte Hindernisgitterkarte (d).

Das Ergebnis der Fusion von geometrischer und visueller Hindernisdetektion führt zu einer optimierten lokalen Hinderniskarte (siehe Abb. 6.7). Fläche Be-

reiche im geometrischen Modell des Bodenroboters, die nicht auf befestigtem Weg liegen und zuvor als befahrbar klassifiziert wurden, sind nun als nicht zu befahren markiert. Die positiven Auswirkungen der zusätzlichen Nutzung der visuellen Hindernisdetektion auf die lokale Pfadplanung werden in den folgenden Kapiteln im Zusammenhang mit der Beschreibung der Details der in dieser Arbeit entwickelten Methode zur Pfadplanung weiter erörtert (siehe Kap. 6.5).

## 6.3 Spezialisiertes Baumwachstum

In diesem Abschnitt wird die Ausgestaltung derjenigen Anteile des Algorithmus besprochen, die den genauen Wachstumsprozess des Pfadplanungsbaums definieren. Die Grundlage für die folgenden Abschnitte bleibt stets der klassische RRT-Algorithmus (siehe Kapitel 6.1 bzw. Algorithmus 2), der als generische Designvorschrift zu verstehen ist. Die konkrete Ausgestaltung des Algorithmus ist als Spezialisierung bzw. Erweiterung des klassischen Algorithmengrundgerüsts zu sehen, welche die Anforderungen der lokalen explorativen Pfadplanung im Außenbereich speziell aufgreift und umsetzt.

### 6.3.1 Zustandsraum

Der Roboter wird als nicht-holonomes Fahrzeug mit drei Freiheitsgraden modelliert. Die Elemente  $x$  des Zustandsraums  $X$  sind infolgedessen durch das Tupel  $\rho' = (p_x, p_y, \phi)$  definiert, das bereits in Kapitel 4 eingeführt wurde. Das Tupel beschreibt die Lage des Fahrzeugs als 2D-Position  $(p_x, p_y)$  mit zugehöriger Orientierung  $\phi$  in der Ebene.

### 6.3.2 Zustandsübergangsfunktion

Auf Basis der Definition des Zustandsraums wird die Zustandsübergangsfunktion  $f$  definiert. Die Funktion beschreibt die Anwendung eines Inputs  $u \in U$  auf den Fahrzeugzustand  $x$  über den Zeitraum  $t$  hinweg. Ein Input ist definiert durch das Tupel  $u = (s, \alpha)$ , das sich aus der Geschwindigkeit  $s$  und dem

aktuellen Lenkwinkel  $\alpha$  des Fahrzeugs zusammensetzt. Der Zeitraum  $t$  entspricht der Konstanten  $\Delta t$ , wie bereits zuvor in Kapitel 6.1 festgestellt wurde. Der Herleitung in [121] folgend, ist das Inkrement  $\dot{f}(x, u) = (\dot{p}_x, \dot{p}_y, \dot{\phi})$  eines Zustands  $x$  durch Anwendung des Inputs  $u$  definiert durch

$$\dot{p}_x = s \cos \phi \quad (6.7)$$

$$\dot{p}_y = s \sin \phi \quad (6.8)$$

$$\dot{\phi} = \frac{s}{L} \tan \alpha \quad (6.9)$$

für  $\Delta t \rightarrow 0$ .  $L$  entspricht dem Achsabstand des Fahrzeugs entlang der Fahrzeuglängsachse.

Für größere Werte von  $\Delta t$  gilt dieser exakte Zusammenhang nicht mehr und die Abbildung 6.15 entspricht einer mit anwachsendem  $\Delta t$  zunehmend ungenaueren Approximation des Inkrements des Fahrzeugzustands. Für kleine Werte von  $\Delta t$  wäre die Anwendung des expliziten Euler-Verfahrens zur numerischen Integration im Sinne der Abbildungsvorschrift

$$f(x, u, t) = x + \Delta t \dot{f}(x, u) \quad (6.10)$$

aufgrund des geringen Genauigkeitsverlusts daher vertretbar. Im Rahmen dieser Arbeit sollte jedoch die Möglichkeit erhalten werden, auch längere Integrationszeiträume wählen zu können. Daher wurde auf das Runge-Kutta-Verfahren zur numerischen Integration zurückgegriffen. Die Zustandsübergangsfunktion ist demnach durch

$$k_1 = \dot{f}(x, u) \quad (6.11)$$

$$k_2 = \dot{f}\left(x + \frac{\Delta t}{2} k_1, u\right) \quad (6.12)$$

$$k_3 = \dot{f}\left(x + \frac{\Delta t}{2} k_2, u\right) \quad (6.13)$$

$$k_4 = \dot{f}(x + \Delta t k_3, u) \quad (6.14)$$

$$f(x, u, t) = x + \frac{\Delta t}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (6.15)$$

definiert, wobei dies dem klassischen Runge-Kutta-Verfahren der vierten Ordnung entspricht. So wird eine deutlich präzisere Approximation des Inkrements des Fahrzeugzustands auch bei größeren Werten für  $\Delta t$  ermöglicht.

### 6.3.3 Kollisionsprüfung

Neben der Zustandsübergangsfunktion  $f$  muss weiterhin die konkrete Ausprägung der Funktion der globalen Bedingungen  $D$  entworfen werden. Der Entwurf zerlegt sich in zwei Teile:

Es ist zu prüfen, ob das betrachtete Element des Zustandsraums einen kollisionsfreien Zustand darstellt. Zu diesem Zweck wird die bereits in Kapitel 6.2 beschriebene Klassifikationsfunktion  $R$  verwendet, die aufgrund von geometrischen und visuellen Sensordaten eine Klassifikation von Gitterzellen der lokalen Hinderniskarte  $\Gamma$  bezüglich deren Befahrbarkeit implementiert.

Der Zustandsraum ist nach der allgemeinen Definition für RRTs aus Kapitel 6.1 zunächst unbeschränkt. In der Praxis der lokalen Pfadplanung gelten jedoch neben der Kollisionsfreiheit weitere Einschränkungen für Elemente des Zustandsraums, die durch untere und obere Schranken beschrieben werden können.

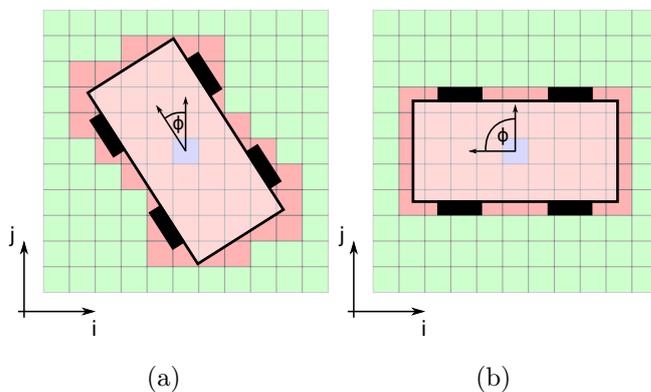


Abbildung 6.8: **Zellabdeckung des Bodenroboters bei verschiedenen Orientierungen.** Zentrale Bezugszelle (blau), abgedeckte Zellen (rot) und freie Zellen (grün).

Da der Roboter nicht als einfaches punktförmiges Fahrzeug modelliert ist, muss die räumliche Ausdehnung des Fahrzeugs mit in die Feststellung der Kollisions-

freiheit eines betrachteten Fahrzeugzustands einbezogen werden. Abhängig von der Orientierung des Fahrzeugs werden unterschiedliche Gitterzellen um die zentrale Bezugsgitterzelle herum abgedeckt (siehe Abb. 6.8). Die Ermittlung der zentralen Bezugsgitterzelle geschieht durch die Projektion des Positionstupels des betrachteten Zustands auf die Gitterkartenebene. Das Positionstupel entspricht der Position des Roboters im szenenfesten Koordinatensystem (siehe Gleichung 6.3). Die Funktion zur Prüfung der Kollisionsfreiheit ermittelt ausgehend von den obigen Überlegungen die vom Roboter abgedeckten Gitterzellen.

Da die Kollisionsprüfung den größten Teil der Rechenzeit während des Baumwachstums einnimmt [182], ist auf einen hohen Optimierungsgrad bezüglich ihrer Implementierung zu achten. Eine in dieser Arbeit speziell entwickelte Methode hilft, diese Anforderung umzusetzen. Die Methode basiert auf der sequenziellen Prüfung der vom Roboter abgedeckten Gitterzellen.

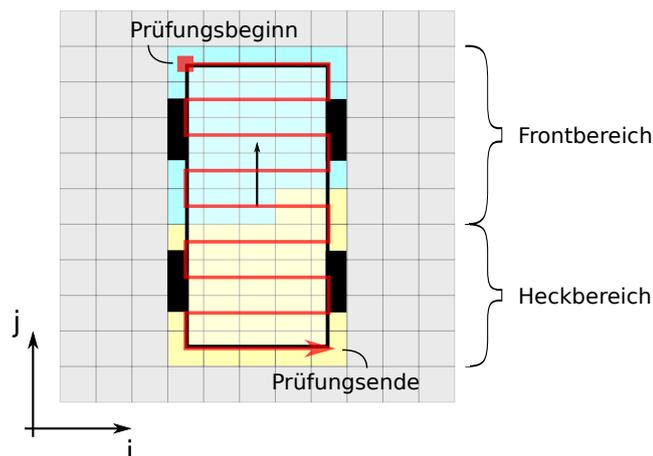


Abbildung 6.9: **Prüfungsreihenfolge abgedeckter Gitterzellen während der Kollisionsprüfung eines betrachteten Zustands.**

Abbildung 6.9 gibt einen zusammenfassenden Überblick über den Ablauf des Prüfungsvorgangs der betroffenen Zellen. Zunächst ist festzustellen, dass die Prüfung von der Fahrzeugfront ausgehend zum Heck des Fahrzeugs hin durchgeführt wird, da bei einem sich in den allermeisten Fällen vorwärts bewegendem Fahrzeug mit einer deutlich höheren Wahrscheinlichkeit mit einer Kollision im vorderen Bereich des Fahrzeugs zu rechnen ist. Sobald eine als HINDERNIS markierte Zelle während der Prüfung detektiert wird, wird der Prüfungsvor-

gang abgebrochen und der betrachtete Zustand als kollidierend markiert. Die weitere Prüfung der restlichen Zellen ist nicht mehr nötig und entfällt. Weiterhin wird bei der Prüfung zwischen der vorderen und der hinteren Hälfte der abgedeckten Gitterzellen unterschieden (vgl. Front- und Heckbereich in Abb. 6.9). Dies ist durch die Tatsache bedingt, dass geprüfte Gitterzellen, die als UNBEKANNT klassifiziert sind, auf zwei unterschiedliche Arten interpretiert werden können. Zum einen stellen unbekannte Zellen eine potenzielle Gefahr dar, da nicht eindeutig festgestellt werden kann, ob sie als FREI oder als HINDERNIS markiert sind. Aus Sicherheitsgründen werden sie daher im Zweifel wie eine als HINDERNIS klassifizierte Zelle behandelt. Zum anderen entsprechen als UNBEKANNT klassifizierte Zellen Teilen der lokalen Gitterkarte, in welchen der Roboter aus der lokalen Karte entkommen kann, was das zentrale Ziel der explorativen lokalen Pfadplanung darstellt. Es ist daher nötig, beide Fälle während der Kollisionsprüfung zu unterscheiden.

Zu diesem Zweck wird eine gesonderte Untersuchung der in Abbildung 6.9 dargestellten Zellen im Front- bzw. Heckbereich des Fahrzeugs durchgeführt. Dieser Untersuchung liegt die Annahme zugrunde, dass für die Detektion eines nutzbaren Austrittspunkts aus der lokalen Karte der überwiegende Teil der abgedeckten Zellen im vorderen Teil des Fahrzeugs zur Klasse UNBEKANNT zugeordnet sein sollte und gleichzeitig die Zellen im hinteren Bereich des Fahrzeugs zum größten Teil als FREI bzw. INTERPOLIERT klassifiziert sein sollten. Sind beide Bedingungen erfüllt, ist ein lokales Explorationsziel gefunden. Algorithmus 4 fasst die soeben beschriebene Kollisionsprüfungsmethode in kompakter Form zusammen, wobei die finale Prüfung des aktuellen Zustands gegen feste untere und obere Schranke in Form der Funktion `OUT_OF_BOUNDS` bereits integriert ist.

Die Funktion `GET_CELL_COVERAGE` liefert dabei den Vektor der abgedeckten Zellen  $cv_{g_{ij}}$  zurück, der abhängig vom aktuell ausgewerteten Fahrzeugzustand  $x$  ist (vgl. Abb. 6.8). Um ein möglichst effizientes Laufzeitverhalten des Algorithmus zu gewährleisten, werden basierend auf einer feinen Diskretisierung des Orientierungswinkels des Fahrzeugs bei bekannter fester Gitterzellenauflösung Abdeckungsvektoren vorberechnet. Somit entsteht ein Vektor von Koordinatentupeln je gewähltem diskretem Orientierungswinkel, wobei jedes Tupel als Koordinatenversatz relativ zur zentralen Bezugszelle zu interpretie-

---

**Algorithmus 4** Ablauf der Prüfung auf Kollisionsfreiheit mit gleichzeitiger Feststellung des Typs der Kollision

---

```

if OUT_OF_BOUNDS(x) then
  | return EXPLORATION;
end

cvgij ← GET_CELL_COVERAGE(x);
nfront ← 0;
nback ← 0;
for each g in cvgij do
  | if gij = HINDERNIS then
  | | return KOLLISION;
  | end
  | else
  | | if gij = UNBEKANNT ∧ cvgij.belongs_to_front(g) then
  | | | nfront ← nfront + 1;
  | | | end
  | | if ( gij = FREI ∨ INTERPOLIERT ) ∧ cvgij.belongs_to_back(g)
  | | then
  | | | nback ← nback + 1;
  | | | end
  | end
end

if nfront < (cvgij.size()/2) * kcvg ∧ nback > (cvgij.size()/2) * kcvg then
  | return KOLLISIONSFREI;
else
  | return EXPLORATION;
end

```

---

ren ist. Die Versatztupel müssen dann lediglich mit der ebenfalls aus dem Fahrzeugzustand  $x$  bestimmten Bezugsgitterzelle entlang der Achsen der Gitterkarte verschoben werden, um den aktuellen Abdeckungsvektor relativ zum Koordinatenursprung der Gitterkarte zu erhalten.  $k_{\text{cvg}} \in [0, 1]$  stellt einen festen Parameter dar, der beeinflusst, wie groß der zugelassene Anteil der nicht bedingungskonformen Zellen im Front- bzw. im Heckbereich des Fahrzeugs sein darf. Algorithmus 4 unterscheidet demnach zwischen kollisionsfreien (KOLLISIONSFREI), kollidierenden (KOLLISION) und explorativen (EXPLORATION) Zuständen bei der Kollisionsprüfung und definiert die Abbruchbedingung des Baumwachstums. Wird ein kollidierender oder explorativer Zustand gefunden, so wird das Baumwachstum nicht weiter fortgeführt, ein weiteres Blatt des Planungsbaums wurde gefunden.

### 6.3.4 Auswertung der Inputs

Im Unterschied zum klassischen RRT-Algorithmus wurde in dieser Arbeit eine alternative Variante des Baumwachstums implementiert, die sich in der Form der Anwendung und Auswertung der Inputs manifestiert. Der Grundgedanke an dieser Stelle besteht darin, direkt nach dem Hinzufügen eines neuen Knotens zum Planungsbaum alle Inputs ausgehend von dem neuen Knoten auszuwerten. Integration der Inputs (siehe Kap. 6.3.2) und Kollisionsprüfung (siehe Kap. 6.3.3) der entstehenden Zustände werden also direkt für jeden neuen Knoten durchgeführt und nicht erst bei einer geplanten Erweiterung des Knotens, wie es der klassische RRT-Algorithmus vorsieht.

Alle in diesem Schritt überprüften, kollisionsfreien Zustände werden einem Vektor von gültigen Zuständen hinzugefügt. Im Erweiterungsschritt während des Baumaufbaus wird im Folgenden nicht der Input gesucht, der den Zustand erzeugt, der einem zuvor gewählten, zufälligen Element des Zustandsraums am nächsten ist. Stattdessen wird das Element des Vektors der überprüften kollisionsfreien Zustände gesucht, das den kleinsten Abstand zum zufällig gewählten Zustand repräsentiert. Abstände zwischen Elementen des Zustandsraums werden durch Anwendung der Metrik  $\Xi$  ermittelt, deren Definition im nächsten Abschnitt besprochen wird.

### 6.3.5 Abstandsmetrik

Die Abstandsmetrik  $\Xi$  bildet zwei Elemente  $x^1$  und  $x^2$  des Zustandsraums auf eine reelle Zahl ab, die ein Maß für den Abstand zwischen diesen Elementen repräsentiert. Die Metrik stellt ein weiteres wichtiges Designkriterium für die konkrete Umsetzung dar und ist in dieser Arbeit definiert durch:

$$\Delta\phi = \begin{cases} \|\phi^1 - \phi^2\|, & \text{wenn } \|\phi^1 - \phi^2\| < 2\pi - \|\phi^1 - \phi^2\| \\ 2\pi - \|\phi^1 - \phi^2\|, & \text{sonst.} \end{cases} \quad (6.16)$$

$$\Xi(x^1, x^2) = \sqrt{(p_x^1 - p_x^2)^2 + (p_y^1 - p_y^2)^2 + \left(\frac{k_{\Xi}}{\pi\Delta\phi}\right)^2} \quad (6.17)$$

$k_{\Xi}$  ist dabei ein frei wählbarer Parameter, der den Einfluss der Orientierungswinkeldifferenz auf die Metrik anpasst. Diese Metrik kombiniert den euklidischen Abstand der 2D-Positionen der verglichenen Knoten mit einem Abstandsmaß für die Orientierung des Knotenzustands. Die Einbeziehung der Orientierung ist beim Einsatz eines nicht-holonomen Roboters besonders wichtig, da ein nicht-holonomer Roboter nur durch Veränderung seiner Position seine Orientierung ändern kann. Mit der hier definierten Metrik wird als Ergebnis der Planung sowohl die richtige Position als auch die korrekte Orientierung im gesuchten Zielzustand erreicht.

## 6.4 Pfadselektion

Im Gegensatz zur klassischen Start-Ziel-Planung wurde in dieser Arbeit eine Methode entwickelt, die alle relevanten Pfade ermittelt, die das Entkommen des Bodenroboters aus einer lokalen Hinderniskarte ermöglichen. Der Selektion des besten Pfades aus der Menge aller entkommenden Pfade kommt somit eine zentrale Rolle zu. Anhand welcher Kriterien diese Entscheidung getroffen werden kann, wird im Folgenden erörtert.

### 6.4.1 Merkmale

Ein Pfad  $A = ({}^A x^0, {}^A x^1, \dots, {}^A x^n)$  mit  ${}^A x^i \in V_{RRT}$  stellt eine Teilmenge der Menge der Knoten  $V_{RRT}$  des Planungsbaums  $G_{RRT}$  dar. Wobei die Elemente von  $A$  durch Kanten also Elemente von  $E_{RRT}$  des Planungsbaums, so verbunden sind, dass die verbindenden Kanten einen zusammenhängenden Pfad vom initialen Element  ${}^A x^0$  (in der Regel ist dies die Wurzel  $x_s$  des Planungsbaums) zum abschließenden Element  ${}^A x^n$  des Pfads erzeugt. Verschiedene Merkmale können zur Auswahl des besten Pfades herangezogen werden. Sie ergeben sich entweder direkt aus der Analyse des Pfads oder werden durch Integration von Submerkmalen ermittelt, die je Knoten bestimmt werden. Die Akkumulation der knotenweisen Submerkmale über den ganzen Pfad ergibt das zugehörige Merkmal für den gesamten Pfad. Direkte Pfadmerkmale sind

- die Länge eines Pfads und
- die Distanz von Start- zu Zielknoten des Pfads.

Submerkmale, die für jeden Knoten des Pfads ermittelt werden können, sind

- die Krümmung des Pfads im betrachteten Knoten,
- die Distanz zum nächsten Hindernis perpendicular zum Pfad,
- die Lagewinkel des Fahrzeugs, ermittelt aus den Aufsatzpunkten der vier Räder,
- die Unebenheit des Bodens unterhalb des Fahrzeugs und
- die Anzahl interpolierter Gitterzellen unterhalb des Fahrzeugs.

#### Direkte Pfadmerkmale

Zur Approximation der Länge des Pfads werden die Längen aller Liniensegmente summiert, die aus der linearen Verbindung jedes Pfadknotens  ${}^A x^k$  mit seinem jeweiligen Nachfolgeknoten  ${}^A x_{k+1}$  entstehen.

Die Distanz zwischen Start- und Zielknoten des Pfads ergibt sich im einfachsten Fall aus der euklidischen Distanz beider Knoten, kann jedoch auch mittels

der in Kapitel 6.3.5 eingeführten Metrik bestimmt werden, welche die Orientierung beider Knoten einbezieht.

### Indirekte Pfadmerkmale

Für die näherungsweise Bestimmung der Krümmung im Knoten  ${}^A x^k$  werden die Liniensegmente untersucht, die aus der Verbindung des Knotens  ${}^A x^k$  mit dem vorherigen Knoten  ${}^A x_{k-1}$  bzw. dem nachfolgenden Knoten  ${}^A x_{k+1}$  entstehen. Der Quotient aus dem Winkel zwischen beiden Liniensegmenten und deren akkumulierter Länge bildet das Merkmal, welches die Krümmung im Knoten  ${}^A x^k$  approximiert.

Die kleinste Distanz zum nächsten Hindernis berechnet sich durch die Verfolgung zweier Strahlen bis zum Auftreffen auf ein Hindernis. Die Richtung der Strahlen ist näherungsweise perpendicular zum Pfad im Knoten  ${}^A x^k$  und durch die positive bzw. negative Normale des Pfads im Knoten  ${}^A x^k$  definiert. Praktisch wird dieses Merkmal durch die Anwendung des Linienzeichnungsalgorithmus nach Bresenham [30] berechnet, da so eine effiziente Abtastung der Hindernisgitterkarte entlang der Normalenrichtungen, ausgehend von der Zelle, in welcher der betrachtete Pfadknoten liegt, durchgeführt werden kann. Sobald dabei eine als Hindernis markierte Zelle getroffen wird, bestimmt sich die Hindernisdistanz durch den euklidischen Abstand zwischen getroffener Hinderniszelle und der Zelle, in welcher der betrachtete Pfadknoten liegt. Die Summe der Abstände in positiver und negativer Normalenrichtung bildet schließlich das endgültige Merkmal.

Für die Bestimmung der Lagewinkel des Fahrzeugs an den entsprechenden Planungsknoten wird die bereits in Kapitel 5.2.4 eingeführte Methode zur Schätzung der Verkipfung der approximierten Bodenebene angewendet. Die Methode schätzt die Normalenrichtung der Bodenebene durch Ermittlung der Eigenvektoren der Kovarianzmatrix einer Punktwolke, die aus Punkten besteht, die in der gesuchten Bodenebene liegen. Die Punkte sind im hier vorliegenden Fall durch die Aufsatzpunkte der Räder des Fahrzeugs gegeben. Sie werden anhand der Gitterzellen, welche die Räder beinhalten, aus der Hinderniskarte ermittelt. Der Winkel zwischen dem Normalenvektor der approximierten Bodenebene und dem Gravitationsrichtungsvektor ergibt schließlich das gesuchte

Merkmal.

In den Ausführungen aus dem vorhergehenden Kapitel 6.3.3 wurde bereits der Zusammenhang zwischen Fahrzeugzustand und abgedeckten Gitterzellen der Hindernisgitterkarte erörtert. Der gleiche Mechanismus wird nun eingesetzt, um sowohl die Unebenheit des Grunds unterhalb des Roboters im Pfadknoten  $A_{x^k}$  als auch die Anzahl abgedeckter interpolierter Gitterzellen zu ermitteln. Die Varianz der den abgedeckten Gitterzellen zugeordneten Höhenwerten wird als Maß für die Unebenheit des Untergrunds genutzt, während die interpolierten Gitterzellen einfach gezählt werden können.

### **Merkmalsinterpretation und -auswertung**

Die im vorherigen Unterkapitel beschriebenen Merkmale dienen im Kontext der Pfadselektion zwei hauptsächlichen Zwecken. Zum einen werden sie genutzt, um den Pfad zu wählen, der ein effizientes, zielstrebiges Bewegungsverhalten des autonomen Roboterfahrzeugs zum Entkommen aus der lokalen Szene erzeugt. Zum anderen können die Merkmale zur Bemessung der Gefährlichkeit eines betrachteten Pfads eingesetzt werden, wobei Gefährlichkeit in diesem Zusammenhang als Wahrscheinlichkeit der Beschädigung der Umgebung oder des Roboters selbst während des Befahrens des Pfads zu interpretieren ist. Pfadlänge und -krümmung werden zur Identifikation von effizienten Pfaden eingesetzt, wohingegen Hindernisdistanz, Lagewinkel, Bodenunebenheit und Interpolationsdichte der Einschätzung der Gefährlichkeit dienen.

Das Ziel der Merkmalsauswertung besteht in der Bestimmung einer Kostenfunktion, die alle Pfadmerkmale so integriert, dass ihre Minimierung stets die Selektion des idealen Pfads in der aktuellen Szene bedeutet. Eine klassische Fusion durch eine gewichtete Summe aller Merkmale hat sich in der Praxis dieser Arbeit als nicht zielführend erwiesen, da zur Auswertung von Merkmalen, die zur Bewertung der Gefährlichkeit des betrachteten Pfads dienen, binäre Entscheidungsfunktionen im Sinne einer klaren Entscheidung zur Nutzung bzw. Eliminierung eines Pfads deutlich zu favorisieren sind. Dies begründet sich durch eindeutige Entscheidungsgrenzen bezüglich der genutzten sicherheitsrelevanten Merkmale. Überschreitet beispielsweise einer der Lagewinkel des Roboters entlang des betrachteten Pfads einen durch die Fahrzeugeigenschaften

des Roboters fest definierten Schwellwert, ist der Pfad direkt zu verwerfen, da unabhängig von der Bewertung anderer Merkmale eine Eliminierung des Pfads unumgänglich ist.

Die Gesamtkostenfunktion zur Pfadauswahl bezieht daher die gewichtete Summe der nicht-sicherheitsrelevanten Pfadmerkmale und einer binären Entscheidungsfunktion ein, die die sicherheitsrelevanten Pfadmerkmale integriert. Die Kostenfunktion des nicht-sicherheitsrelevanten Pfadmerkmals  $m_i$  wird durch die Abbildung

$$m_i : A \rightarrow [0, 1] \quad (6.18)$$

beschrieben. Die Entscheidungsfunktion des sicherheitsrelevanten Pfadmerkmals  $\hat{m}_j$  hat die Form

$$\hat{m}_j : A \rightarrow \{0, 1\}. \quad (6.19)$$

Die Gesamtkostenfunktion des Pfads  $A_k$  ist darauf aufbauend definiert durch

$$K(A_k) = \sum_{i=0}^{N_i} m_i(A_k) + \prod_{j=0}^{N_j} 1 - \hat{m}_j(A_k). \quad (6.20)$$

Die Selektion des besten Pfads  $A_{best}$  wird durch das Minimierungsproblem

$$A_{best} = \min_k \{K(A_k)\} \quad (6.21)$$

beschrieben. Zur Sicherstellung der Existenz mindestens eines auf die zugehörigen Entscheidungsfunktionen  $\hat{m}_j$  bezogenen gültigen Pfads wird das Ergebnis  $S$  des Planungsalgorithmus durch

$$S = \begin{cases} \text{Planungserfolg} & \text{wenn } \sum_{k=0}^{N_k} \prod_{j=0}^{N_j} \hat{m}_j(A_k) > 0 \\ \text{Planungsmisserfolg} & \text{sonst} \end{cases} \quad (6.22)$$

bestimmt.

## Schnittstelle zur globalen Pfadplanung

Die bisherige Betrachtung von Merkmalen zur Pfadselektion umfasst ausschließlich Merkmale, die bezogen auf die lokale Umgebung der aktuellen Szene definiert sind. Im Sinne der Entwicklung einer leistungsfähigen Methode zur lokalen explorativen Pfadplanung ist dieser Ansatz vollständig ausreichend, da nicht die Verfolgung eines globalen Ziels, sondern das effiziente, sichere Entkommen aus der lokalen Szene im Mittelpunkt steht. Im realen Einsatz des in dieser Arbeit entwickelten explorierenden, autonomen Robotersystems stehen jedoch globale Wegpunkte, die im erdfesten Koordinatensystem definiert sind, als grobe Planungsanhaltspunkte zur Verfügung. Wird diese Information genutzt, um einen globalen Handlungsplan zu realisieren, kann die Anzahl der selektierbaren Pfade aus der lokalen Pfadplanung so eingeschränkt werden, dass die verbleibenden Pfade den Roboter tendenziell in Richtung des aktuellen globalen Wegpunkts führen. Das Ziel ist dabei nicht die präzise Ansteuerung des nächsten globalen Wegpunkts, sondern vielmehr die grobe Leitung des lokalen Planungsvorhabens.

Um den aktuellen globalen Wegpunkt für die lokale Planung nutzbar zu machen, wird von diesem eine Zielpose in der lokalen Szene abgeleitet. Durch Selektion von Explorationspfaden, deren Endknoten eine maximale Distanz zur zuvor bestimmten, lokalen Zielpose nicht überschreiten, verbleiben ausschließlich Pfade in der Menge der selektierbaren Pfade, die der Exploration im Sinne der globalen Pfadplanung dienlich sind. Zur Bestimmung der gesuchten Zielpose  ${}^g_m\rho = ({}^g_m p \quad {}^g_m q)$  werden zusätzlich zu den in Kapitel 5.2.3 eingeführten Koordinatensystemen zwei weitere definiert. Zum einen das Koordinatensystem  $f_m$ , das die Lage der aktuellen Szene im erdfesten Koordinatensystem beschreibt. Zum anderen das Koordinatensystem  $f_g$ , das die Lage des aktuellen globalen Wegpunkts im erdfesten Koordinatensystem definiert. Mithilfe dieser Koordinatensysteme werden die Positions- und die Orientierungskomponente der lokale Zielpose  ${}^g_m\rho$  wie folgt ermittelt:

$${}^g_w p' = \frac{l_m ({}^g_w p - {}^m_w p)}{2 \|{}^g_w p - {}^m_w p\|} + {}^m_w p \quad (6.23)$$

$${}^g_w\phi' = 2 \arctan\left(\frac{{}^g_w p'_y}{\sqrt{{}^g_w p'^2_x + {}^g_w p'^2_y + {}^g_w p'^2_x}}\right) - \frac{\pi}{2} \quad (6.24)$$

$${}^g_w q' = \left(\cos \frac{{}^g_w\phi'}{2} \quad 0 \quad 0 \quad \sin \frac{{}^g_w\phi'}{2}\right)^T \quad (6.25)$$

$${}^g_m \check{p} = {}^m_w \bar{q} \left({}^g_w \check{p}' - {}^m_w \check{p}\right) {}^m_w q \quad (6.26)$$

$${}^g_m q = {}^m_w \bar{q} {}^g_w q', \quad (6.27)$$

wobei  $l_m$  die Seitenlänge der lokalen Hinderniskarte bezeichnet. Definiert  $x_g$  das der Zielpose entsprechende Element des Zustandsraums  $X$ , so können durch Anwendung der Abstandsmetrik  $\Xi$  mit anschließendem Schwellwertvergleich die Knoten des Planungsbaums ermittelt werden, die einen gewissen Maximalabstand zur berechneten Zielpose nicht überschreiten. Dabei kommen ausschließlich Knoten infrage, die zuvor als Explorationsknoten klassifiziert worden sind. Ist die Menge der - der globalen Planung dienlichen - Explorationsknoten gefunden, kann die bereits zuvor eingeführte Methode zur Pfadselektion anhand lokaler Pfadmerkmale benutzt werden, um den besten Pfad unter den ermittelten Kandidaten zu finden.

## 6.5 Ergebnisse

Zur anschaulichen Darstellung der Ergebnisse der in dieser Arbeit entwickelten Methode zur explorativen lokalen Pfadplanung werden in diesem Kapitel verschiedene Szenen aus einer autonomen Testfahrt des Roboters AMOR in chronologischer Reihenfolge gezeigt und ausgewertet. Zusätzlich werden Ergebnisse der lokalen Pfadplanung gezeigt, die die Auswirkungen der Nutzung der zusätzlichen visuellen Hindernisdetektion unter Verwendung der Luftbildkamera des kooperierenden Flugroboters PSYCHE zeigen.

### Autonome Testfahrt

Abbildung 6.10 zeigt eine Übersichtskarte der autonomen Testfahrt, in der sowohl der Ablauf der Testfahrt als auch die Positionen der ausgewerteten Szenen verzeichnet sind.

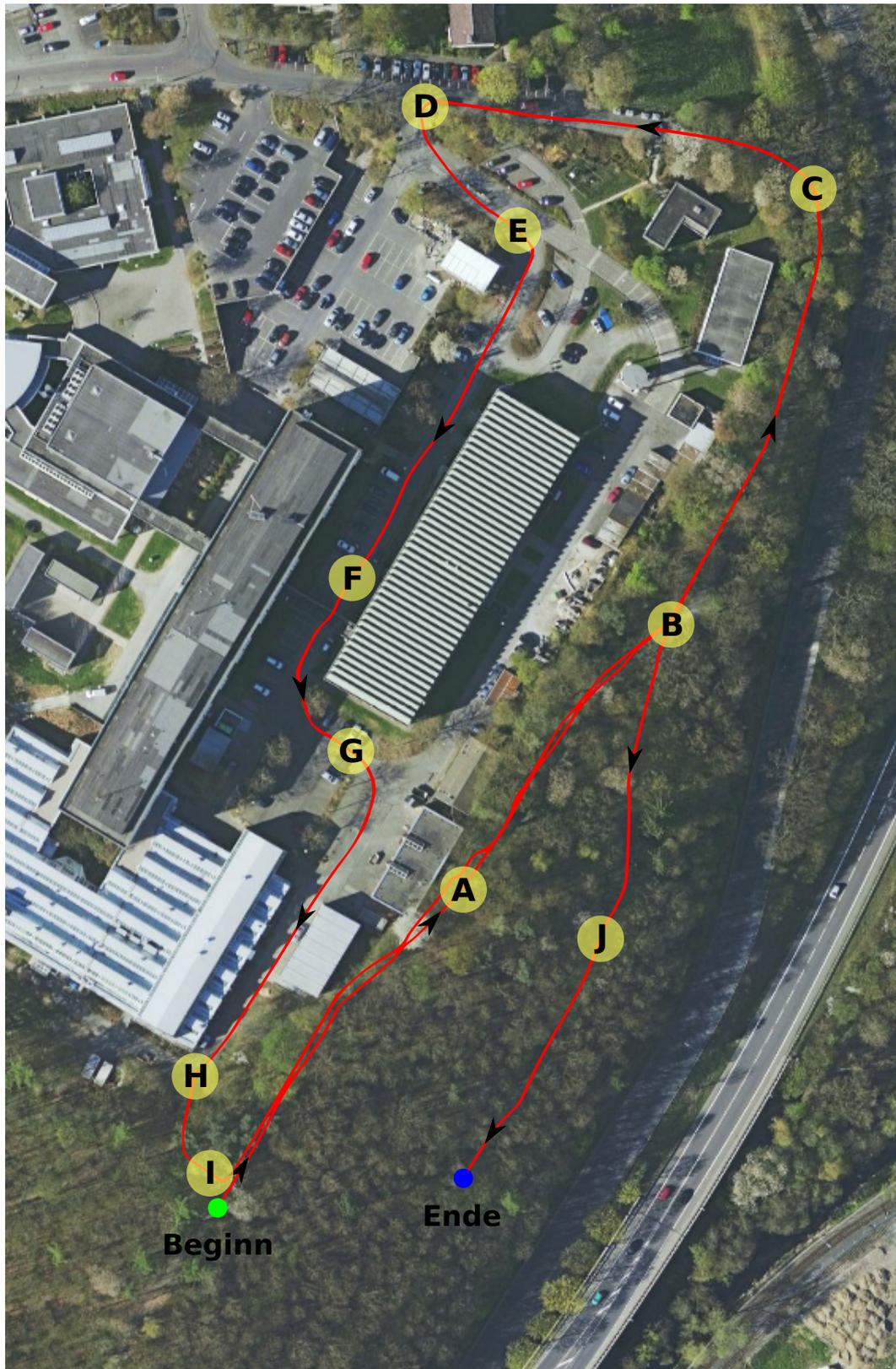


Abbildung 6.10: Autonome Testfahrt des Roboters AMOR zur Evaluation der entwickelten Methode zur explorativen lokalen Pfadplanung. Ausgewertete Szenen sind mit den Buchstaben A bis J markiert.

Jede ausgewertete Szene wird im Folgenden durch eine Hinderniskarte dargestellt, in welche die Knoten des Planungsbaums eingezeichnet worden sind. Entsprechend der bereits in Kapitel 6.2 eingeführten farblicher Kodierung sind belegte Zellen der Hinderniskarte entweder in Grauwerten oder in Rot gefärbt, je nachdem ob eine Zelle als freie Zelle oder als Hinderniszelle klassifiziert wurde. Unbekannte Bereiche der Hinderniskarte sind gelb gefärbt. Grüne Knoten kennzeichnen Baumknoten, die weder als kollidierende Knoten noch als Explorationsknoten klassifiziert wurden. Blaue Knoten sind Explorationsknoten, die jeweils einem Pfad und damit einer Handlungsoption im Sinne der explorativen lokalen Pfadplanung entsprechen. Die Kanten des Planungsbaums sind aus Gründen der Übersichtlichkeit nicht eingezeichnet. Der gewählte beste Pfad ist als schwarze Linie eingezeichnet, welche die zum Pfad zugehörigen Baumknoten verbindet. Um die Interpretation der Planungsergebnisse zu vereinfachen ist für jede ausgewertete Szene neben dem Planungsergebnis zusätzlich das Kamerabild der Frontkamera für die entsprechende Szene abgebildet.

Die Testfahrt beginnt auf einem unbefestigten Waldweg. Die Szenen A und J sind gute Beispiele für die dort anzutreffenden Umgebungsbedingungen und daraufhin zu erwartende Planungsergebnisse. Problematisch aus Pfadplanungssicht sind hier enge Durchfahrten, die durch Vegetation am Wegesrand erzeugt werden. Hier zeigt sich der Vorteil der präzisen Kollisionsdetektion der entwickelten Methode, die unter genauer Berücksichtigung der Fahrzeugausmaße auch bei komplexen Hinderniskonfigurationen erfolgreiches Planen erlaubt. Szene B zeigt den Übergang von befestigtem zu unbefestigtem Waldweg an einer Abzweigung. Einer der abgehenden Wege weist einen sehr steilen Winkel auf. Trotzdem ist die Planungsmethode in der Lage, Pfade für beide Handlungsoptionen (links bzw. rechts abbiegen) zu erzeugen. Szene C zeigt ebenfalls eine Abzweigung mit zwei Handlungsoptionen. Trotz eines großen Hindernisses, das einen wichtigen Teil der Szene verdeckt, können beide Handlungsoptionen sicher gefunden werden. In den Szenen D, E und G zeigt sich die Fähigkeit der Planungsmethode, bei offenen, freien Flächen eine gleichmäßige und zügige Abtastung der Szene mit dem Planungsbaum zu erreichen, mit dem Ergebnis, dass die der Exploration dienlichen Fluchtmöglichkeiten auch hier sicher erkannt werden können.

In Szene F finden sich diverse näherungsweise punktförmige Hindernisse (Schil-

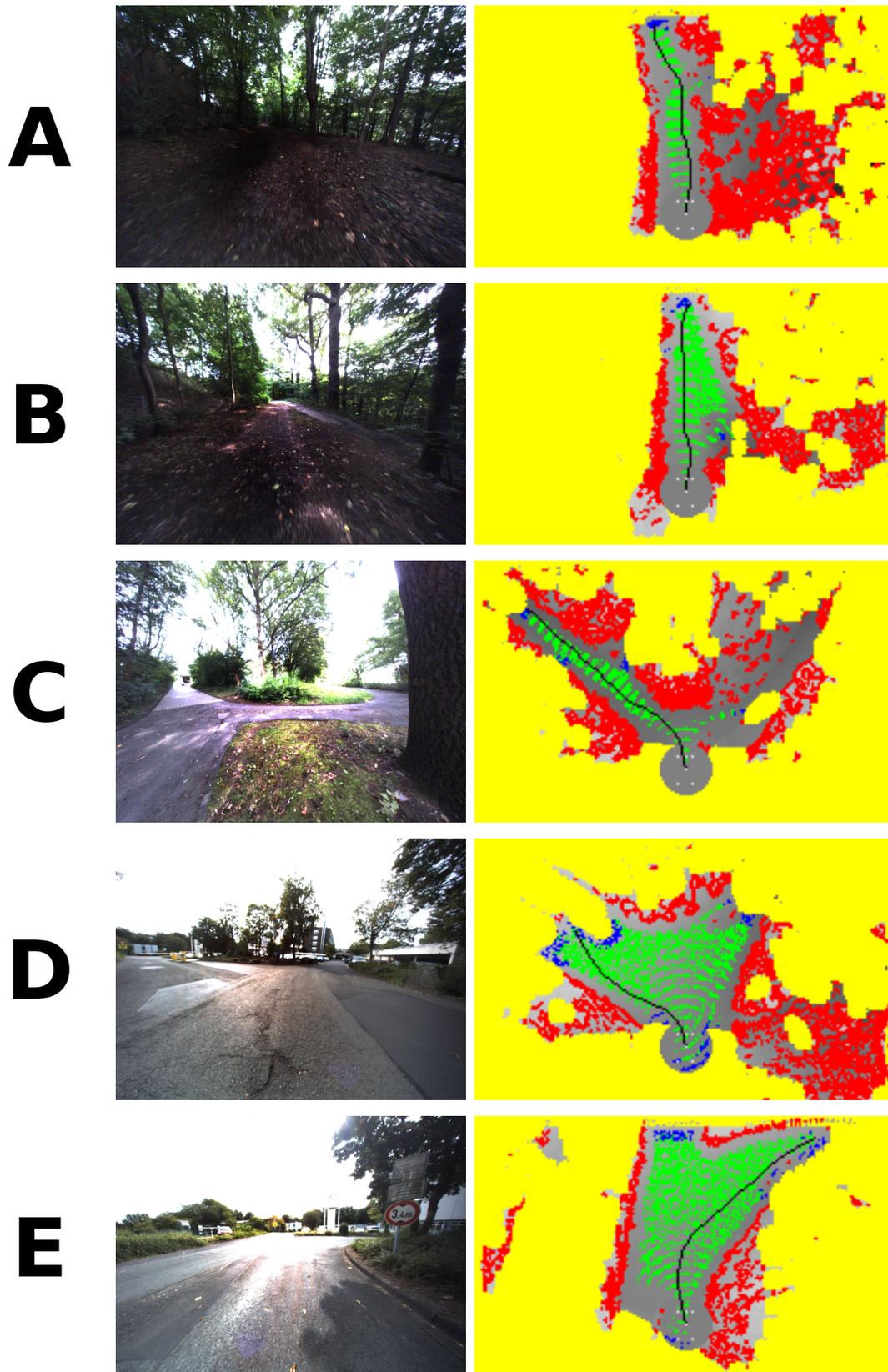


Abbildung 6.11: **Testfahrt-Szenen A bis E.** Kamerabild der Frontkamera (links). Hinderniskarte, Planungsbaumknoten und selektierter Pfad (rechts).

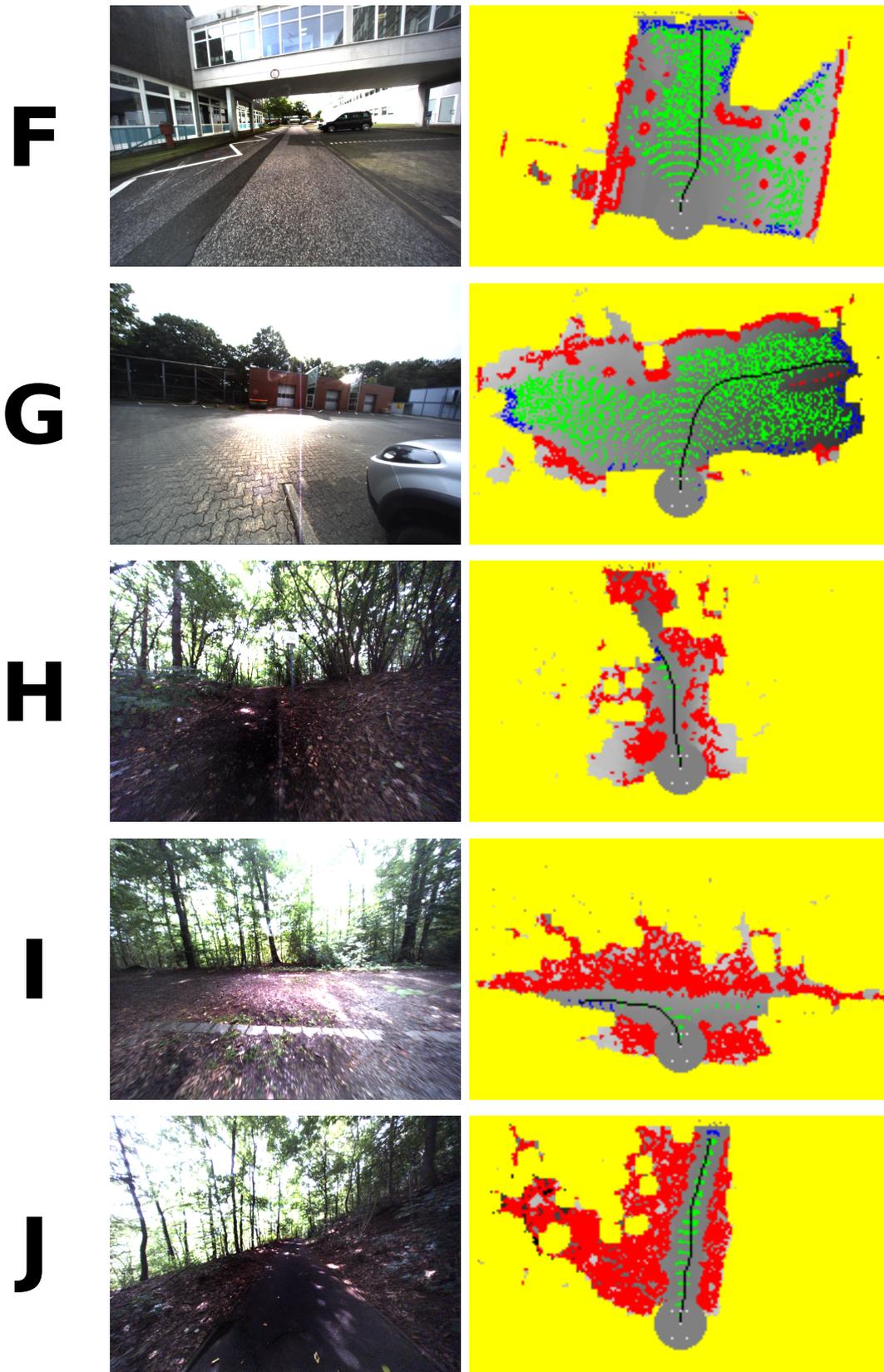


Abbildung 6.12: Testfahrt-Szenen F bis J. Kamerabild der Frontkamera (links). Hinderniskarte, Planungsbaumknoten und selektierter Pfad (rechts).

der, Säulen etc.), welche die Planung von komplexen Pfaden verlangen, um diese kollisionsfrei zu umfahren. Die entwickelte Pfadplanungsmethode ist hier problemlos in der Lage, einen den befahrbaren Raum sehr gut abdeckenden Planungsbaum zu erzeugen, der neben dem erfolgreichen Umfahren der Hindernisse die Erzeugung aller relevanten explorativen Handlungsalternativen erlaubt. Szene I zeigt eine typische Abzweigungsszene auf einem unbefestigten Waldweg. Enge Durchfahrten, die meist keinen Rangierspielraum bieten, machen eine präzise lokale Planung unumgänglich, um alle Handlungsoptionen sicher detektieren zu können und entsprechende Pfade zu generieren. Wie das Planungsergebnis der Szene I zeigt, ist die implementierte Methode in der Lage, auch diese komplexe Aufgabe erfolgreich zu lösen. Zusammenfassend ist festzustellen, dass die in dieser Arbeit entwickelte Methode zur lokalen Pfadplanung in der Lage ist, eine große Vielfalt an lokalen Hinderniskonfigurationen im urbanen wie im Offroad-Bereich erfolgreich zu meistern und Pfade zu planen, die den autonom operierenden Bodenroboter sowohl kollisionsfrei als auch effizient zu den jeweiligen lokalen Explorationszielen leiten.

### **Visuelle Fahrkorridorerkennung**

In Kapitel 6.2.2 wurde bereits detailliert beschrieben, wie Daten der Luftbildkamera des kooperierenden Flugroboters PSYCHE zur verbesserten Fahrkorridorerkennung in der Umgebung des Bodenroboters genutzt werden können. In Szenen, die aufgrund zu geringer lokaler Höhenunterschiede keine Unterscheidung von zu bevorzugendem Fahrkorridor und restlicher Umgebung anhand rein geometrischer Merkmale erlauben, wird es so möglich, deutlich zielgerichteter zu planen. Abbildung 6.13 zeigt zwei Beispielszenen, die den Vorteil der zusätzlichen Nutzung von visuellen Informationen eines kooperierenden Flugroboters für die Pfadplanung verdeutlichen.

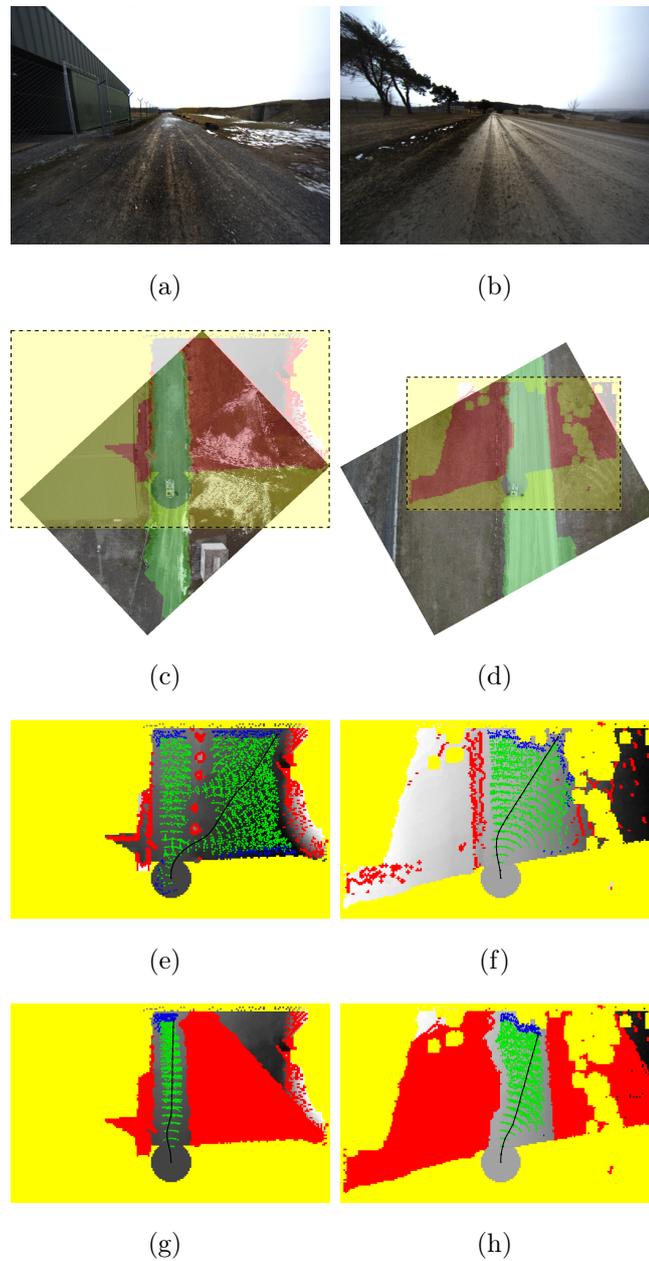


Abbildung 6.13: **Ergebnisse der Pfadplanung mit visueller Fahrkorridorerkennung im Live-Luftbild.** Frontkamerabild (a und b) und Live-Luftbild der Szene mit Overlay der Hinderniskarte mit visueller Fahrkorridorerkennung (c und d). Planungsergebnis ohne visuelle Fahrkorridorerkennung (e und f) und mit visueller Fahrkorridorerkennung (g und h).



# Kapitel 7

## Fazit

In dieser Arbeit wurde die Aufgabe der Entwicklung eines integrierten Systems zur Lösung des Problems der autonomen Exploration im unstrukturierten Außenbereichsszenario erfolgreich gelöst. Der Einsatz eines Teams, bestehend aus Bodenroboter AMOR und Flugroboter PSYCHE, stellt dabei eine vollständig neue und innovative Herangehensweise an diesen Forschungsbereich dar. Die schritthaltende Verfolgung des Bodenroboters durch den Flugroboter erlaubt den Einsatz des Flugroboters als externe Sensorplattform. Die so entstehende zusätzliche Betrachtungsperspektive der aktuellen lokalen Szene aus einem fundamental anderen Blickwinkel ermöglichte neben einer Reinterpretation und Erweiterung bekannter Methoden die Entwicklung vollkommen neuer Lösungsansätze. Der wissenschaftliche Beitrag dieser Arbeit wurde hauptsächlich in drei Themenbereichen erarbeitet: Lokalisierung, Umgebungsmodellierung und Pfadplanung. Diese drei Themenbereiche decken essentielle Teilkomponenten eines autonomen, mobilen, robotischen Systems ab. Der Aspekt der Kooperation zwischen Boden- und Luftroboter wurde neben der Beschreibung der rein bodenbasierten Verfahren im Kontext dieser drei Themengebiete jeweils aufgegriffen und anhand innovativer Anwendungen in den Rahmen dieser Arbeit eingebettet. Zusätzlich wurde durch die Konzeption und Implementierung eines funktionierenden, einsatzfähigen Gesamtsystems die Plattform für eine Evaluation der entwickelten Methoden in realen Szenarien geschaffen.

Für die Lokalisierung des Bodenroboters im erdfesten Koordinatensystem wurden zwei neue Verfahren eingeführt, die unabhängig von zwar etablierten aber

teilweise unzuverlässigen, satelliten-basierten Lokalisierungsmethoden auf dem Bodenroboter realisiert wurden. Zum einen wurde ein probabilistischer Algorithmus auf Basis eines Partikelfilters vorgestellt, der eine Schätzung der aktuellen Pose des Roboters ausschließlich auf Basis von Daten aus der Koppelnavigation des Bodenroboters und metrisch-topologischen Kartendaten aus einem Geoinformationssystem ermöglicht. Zum anderen wurde eine Registrierung des Live-Luftbilds des kooperierenden Flugroboters mit einem geo-registrierten Kartenausschnitt aus einer Luftbilddatenbank durchgeführt, um eine präzise Lokalisierung des im Live-Luftbild sichtbaren und automatisch detektierten Bodenroboters zu erreichen.

Die Umgebungsmodellierung des Roboterteams wurde durch Fusion von Daten aus multiplen, sowohl redundanten als auch komplementären Sensorquellen erreicht. Das Ergebnis der Modellierung der Umwelt mit mehreren aktiv bewegten Laserscannern und verschiedenen Kameras ist nicht wie häufig üblich eine Punktwolke, sondern ein Dreiecksnetz, welches die geometrische Beschaffenheit der Umwelt nicht als ungeordnete Menge von Punkten, sondern als Oberfläche beschreibt. Zusätzlich angereichert wird das geometrische Modell mit visuellen Informationen. Dabei kommen Kameras mit entozentrischem und katadioptrischem Objektiv auf dem Bodenroboter sowie eine Luftbildkamera auf dem Flugroboter zum Einsatz. Die Texturierung des durch den Bodenroboter erzeugten geometrischen Modells mit Luftbilddaten eines Flugroboters in Echtzeit stellt dabei einen zentralen Innovationsaspekt dieser Arbeit dar. Das resultierende Modell kann zur Lösung von Problemen herangezogen werden, die mit einem rein geometrischen Modell nicht lösbar gewesen wären, wie etwa die Unterscheidung ebener Teilflächen, die verschiedenen Untergrundformen entsprechen.

Exploration bedeutet auf die Pfadplanung eines autonomen Roboters bezogen, dass kollisionsfreie Pfade geplant werden, die den Roboter aus der bekannten Umgebung ausbrechen lassen und ins Unbekannte führen. Aufgrund der meist vorherrschenden Start-Ziel-Planungsdirektive sind explorationsgetriebene Algorithmen jedoch unterrepräsentiert. Aus diesem Grund wurde in der vorliegenden Arbeit eine neue Methode zur explorativen Pfadplanung eingeführt, die auf der Basis von Rapidly Exploring Random Trees kinematisch zulässige Pfade für einen nicht-holonomen Roboter konstruiert. Dabei liegt das Haupt-

augenmerk auf einer möglichst schnellen und vollständigen Ermittlung aller sogenannten Explorationspfade, die auf Basis einer lokalen Hinderniskarte aus dem fusionierten Umgebungsmodell erzeugt werden können. Die explorative Pfadplanung wird dabei sowohl von klassischen geometrischen Merkmalen als auch von visuellen Merkmalen im zuvor registrierten Luftbild geleitet.

Die entwickelten Algorithmen und Verfahren wurden experimentell während einer Vielzahl von intensiven Testreihen in Situationen und Beispielszenen unter realen Bedingungen im Außenbereich mit dem vollständig autonom operierenden Roboterteam verifiziert und quantifiziert.

### **Kritische Reflexion und Ausblick**

Im Rahmen der vorliegenden Arbeit wurden umfangreiche Experimente im unstrukturierten Außenbereich durchgeführt. Als fundamentale Erfahrung, sozusagen als Essenz der Experimente, lässt sich festhalten, dass dieses Einsatzszenario stets neue Überraschungen bereithält. Die extreme inhärente Komplexität des Szenarios lässt sich in einer einzigen Arbeit nicht vollständig abdecken. Unterschiedliche Landschaften, Beleuchtungsverhältnisse und Wetterlagen, um nur die wichtigsten Einflussfaktoren zu nennen, erfordern die Konzentration auf lösbare Konstellationen und gangbare Vorgehensweisen. Die sukzessive Reduzierung der damit verbundenen einschränkenden Randbedingungen kann wesentliche Hinweise auf potenzielle zukünftige Forschungstätigkeiten aufbauend auf den Ergebnissen dieser Arbeit geben. Im Folgenden werden verschiedene Ansatzpunkte für mögliche Erweiterungen der neu eingeführten Methoden diskutiert. Dies geschieht im Sinne einer kritischen Reflexion der Ergebnisse und orientiert sich an den thematischen Schwerpunkten dieser Arbeit.

Die in Kapitel 4.3 beschriebene kartenbasierte Lokalisierungsmethode ermöglicht die Positionsbestimmung des Bodenroboters, ohne auf satellitenbasierte Positionsquellen zugreifen zu müssen. Die Grundvoraussetzung für den Einsatz dieser Methode ist die Existenz von digitalen Wegenetzdaten im Einsatzgebiet des Bodenroboters. Die weitreichende Verfügbarkeit von Wegenetzdaten bildet dabei die Grundlage für die großflächige Anwendbarkeit dieser Methode. Trotzdem gibt es sicherlich Einsatzgebiete, in denen das vorhandene Wegenetz noch teilweise lückenhaft oder fehlerbehaftet ist. Eine zukünftige Weiterentwick-

lung der eingeführten kartenbasierten Lokalisierungsmethode könnte daher in der frühzeitigen Erkennung von Kartenfehlern bestehen, um die Methode der Positionsschätzung entsprechend adaptieren zu können. Der vorgestellte Lokalisierungsalgorithmus wurde bereits intentional so entworfen, dass eine Erweiterung in diesem Sinne keinen vollständigen Neuentwurf bedeuten würde. Dies liegt in der Tatsache begründet, dass im Gegensatz zu alternativen Methoden keine strikte Bindung der Partikel an die Kanten des digitalen Wegenetzes gefordert wird. Selbstverständlich besteht unabhängig davon auch stets die Möglichkeit, fehlende Daten zu ergänzen oder fehlerhafte Daten zu korrigieren. Dies kann entweder (semi)automatisch oder manuell durchgeführt werden und hat sich in der experimentellen Praxis eines begrenzten Einsatzgebietes bereits als sehr effektiv erwiesen.

Die zweite innovative Lokalisierungsmethode, die in dieser Arbeit entwickelt wurde, nutzt georegistrierte Luftbilddaten aus einer Geodatenbank, um durch einen Vergleich mit aktuellen Luftbilddaten des Flugroboters eine Korrespondenz zwischen beiden Datenquellen herzustellen. So wird es möglich, die Position des Bodenroboters im erdfesten Koordinatensystem abzuleiten, nachdem er im Live-Luftbild des Flugroboters detektiert wurde. Sowohl die Registrierung der Luftbilder als auch die Detektion des Bodenroboters im Live-Luftbild werden durch den Einsatz von Methoden der digitalen Bildverarbeitung erreicht. Die Voraussetzungen für die erfolgreiche Lokalisierung des Bodenroboters sind daher erstens die hinreichende Ähnlichkeit des visuellen Erscheinungsbilds der verglichenen Luftbilder und zweitens die Verfügbarkeit von deutlichen visuellen Merkmalen in den Luftbildern. Diese dienen als Ankerpunkte für die Registrierung. Die entwickelte Methode ist durch den großflächigen Vergleich der Luftbilder robust gegen verschiedene Störeinflüsse. Dennoch kann die Ähnlichkeit des visuellen Erscheinungsbilds der verglichenen Luftbilder beispielsweise durch unterschiedliche Jahreszeiten (Schnee, Laub etc.) so stark beeinträchtigt sein, dass das Ergebnis der Lokalisierung verfälscht wird. Die Bereitstellung von georeferenzierten Luftbildern in Geodatenbanken, die zu verschiedenen Jahreszeiten aufgenommen wurden, ist ein Ansatz zur Lösung dieses Problems. Da die Umfänge bestehender Geodatenbanken stetig ausgebaut, ist eine Erweiterung der vorgestellten Algorithmen in dieser Richtung in Zukunft vorstellbar und sinnvoll. Auch eine ausreichende Verfügbarkeit von aussage-

kräftigen Merkmalen kann im unstrukturierten Außenbereichsszenario nicht in jeder Szene garantiert werden, da zeitweise visuell sehr homogen strukturierte Szenen auftreten können (Monokulturen, geschlossene Schneedecke etc.). Erste Experimente haben jedoch gezeigt, dass die Variation der Flughöhe des Flugroboters in solchen Fällen einen Lösungsansatz bieten könnte, da so z.B. eine größere Fläche und damit potenziell mehr eindeutige Merkmale beobachtet werden können. Dies setzt jedoch ein deutlich intelligenteres, situationsangemessenes autonomes Verhalten des Flugroboters voraus. Er müsste in der Lage sein, automatisch die Qualität der Messungen aus verschiedenen Aufnahme-  
posen zu schätzen und seine Flugbahn entsprechend zu planen. Eine derartige Erweiterung der autonomen Fähigkeiten des Flugroboters hätte den Rahmen dieser Arbeit überschritten, stellt aber ein vielversprechendes Thema für zukünftige Forschungstätigkeiten dar.

Die in Kapitel 5 eingeführten Methoden zur Umweltmodellierung durch Sensorfusion ermöglichen die Erzeugung einer äußerst beschreibungsmächtigen Repräsentation der lokalen Umwelt des Bodenroboters. Ausgehend davon werden Methoden entwickelt, die eine Klassifikation der Umwelt in befahrbare und unbefahrbare Bereiche ermöglichen. Diese Klassifikation basiert auf der Geometrie der analysierten Szene, die als unveränderlich angenommen wird. Merkmale wie Verformbarkeit oder Verschiebbarkeit von Objekten werden nicht explizit modelliert. Teilweise werden sie zwar indirekt mitgeschätzt, da z.B. zwischen der geometrischen Form und der Verformbarkeit Zusammenhänge bestehen. Eine dedizierte Messung mit entsprechend optimierter Sensorik könnte aber zusätzliche hilfreiche Informationen erbringen. Die Modellierung dieser zusätzlichen Merkmale einer Szene bietet erweiterte Handlungsmöglichkeiten für den agierenden Roboter, insbesondere in hochkomplexen Szenenkonfigurationen (z.B. im Buschwerk des Waldes), und wäre somit eine sinnvolle Erweiterung der in dieser Arbeit entwickelten Methoden.

Eine grundlegende Designentscheidung für die Entwicklung der in Kapitel 6 eingeführten Methode zur explorativen Pfadplanung besteht in der Beschreibung des Roboters als starrer Körper mit drei Freiheitsgraden. Aufbauend auf dieser Definition wird die Umwelt des Roboters im Rahmen der lokalen Pfadplanung als Gitterkarte mit annotierten Höheninformationen je Gitterzelle beschrieben. Diese Auslegung des Planungsalgorithmus ist durch die Fähigkeiten

der mechanischen Plattform des Bodenroboters motiviert. Praktisch alle Szenenkonfigurationen, die sich mit dem aktuellen Bodenroboter befahren lassen, können mit der gewählten Repräsentationsform der Umwelt für die Pfadplanung hinreichend präzise beschrieben werden, sodass eine erfolgreiche Operation des Roboters stets möglich ist. Im Rahmen einer zukünftigen Erweiterung der mechanischen Plattform des Bodenroboters könnten extreme Szenenkonfigurationen beherrschbar werden. Der nächste logische Schritt wäre somit die Erweiterung der Beschreibung des Roboters als starrer Körper mit sechs Freiheitsgraden und entsprechend erweiterter Umweltbeschreibung. Die Pfadplanungsmethode ist bewusst flexibel und modular entworfen worden, sodass eine solche Erweiterung problemlos möglich ist. Es ist jedoch festzustellen, dass die aktuell auf einem mobilen Roboter zur Verfügung stehenden Rechenressourcen eine Berechnung in Echtzeit für eine in dieser Art erweiterte Planungsmethode noch nicht leisten können. Dies stellt daher ein Thema zukünftiger Forschungstätigkeiten dar.

Kurz soll noch der wichtige Aspekt der Missionsplanung für ein leistungsfähiges autonomes Roboterteam beleuchtet werden. Wie bereits an einigen Stellen dieser Arbeit betont wurde, stellt dieser Themenbereich keinen zentralen Aspekt dieser Arbeit dar. Dennoch wurde im Rahmen der Arbeit ein integriertes autonomes Roboterteam geschaffen, das die Aufgabe der autonomen Navigation im realen Außenbereichsszenario erfolgreich demonstriert. Um dies zu erreichen, wurde eine Verwaltungsarchitektur geschaffen (siehe Kapitel 3.3), die Mechanismen beinhaltet, um problematische Situationen während der Bearbeitung einer Mission aufzulösen. Beispielsweise sind Bewältigungsstrategien für lokale Sackgassen oder nicht erreichbare globale Wegpunkte implementiert. Zahlreiche Testfahrten haben gezeigt, dass mit diesem Ansatz ein Großteil der auftretenden Situationen während der autonomen Operation des Roboterteams bewältigt werden kann. Aufgrund der hohen Komplexität des unstrukturierten Außenbereichsszenarios existieren jedoch Sondersituationen, die einer spezifischen Behandlung bedürfen, die über die in dieser Arbeit implementierten Bewältigungsstrategien hinausgeht. Die sukzessive Identifikation solcher Sondersituationen und die zugehörige Erweiterung der Menge der Bewältigungsstrategien stellen somit zukünftige Forschungsthemen basierend auf den Ergebnissen dieser Arbeit dar.

Die Rolle des Flugroboters PSYCHE im kooperierenden Team ist in dieser Arbeit bewusst rein reaktiv ausgelegt. Der Flugroboter fungiert als externe Sensorplattform und ist strikt an die Bewegungen des Bodenroboters gekoppelt. Wie in dieser Arbeit an vielen Stellen gezeigt wird, stellt bereits diese Herangehensweise eine enorme Bereicherung für die Gesamtfähigkeiten des Roboterteams dar. Dennoch könnte mit einer noch größeren Leistungsfähigkeit des Roboterteams gerechnet werden, wenn der Fähigkeiten des Flugroboters in Richtung einer freieren Kopplung mit dem Bodenroboter erweitert werden würde. Unabhängigere Flugmanöver könnten in einigen Szenen eine intensivere Betrachtung von besonders relevanten Details einer Szene ermöglichen, was wiederum zu einem optimierten Verhalten des Bodenroboters führen würde. Die Komplexität dieser Aufgabe dürfte jedoch erheblich sein, da dies sowohl eine äußerst detaillierte und differenzierte Szeneninterpretation des Bodenroboters als auch die aufwendige Erweiterung der autonomen Fähigkeiten des Flugroboters voraussetzen würde.



# Literaturverzeichnis

- [1] Pankaj K Agarwal, Prabhakar Raghavan, und Hisao Tamaki. Motion planning for a steering-constrained robot through moderate obstacles. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 343–352. ACM, 1995.
- [2] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, und Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.
- [3] Omar Ait Aider, Philippe Hoppenot, und Etienne Colle. A model-based method for indoor mobile robot localization using monocular vision and straight-line correspondences. *Robotics and Autonomous Systems*, 52(2): 229–246, 2005.
- [4] Omead Amidi und Chuck E Thorpe. Integrated mobile robot control. In *Fibers' 91, Boston, MA*, pages 504–523. International Society for Optics and Photonics, 1991.
- [5] Adnan Ansar und Konstantinos Daniilidis. Linear pose estimation from points or lines. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):578–589, 2003.
- [6] Christopher Armbrust, Tim Braun, Tobias Föhst, Martin Proetzsch, Alexander Renner, H Schaefer, und Karsten Berns. Ravon - the robust autonomous vehicle for off-road navigation. In *Proceedings of the IARP international workshop on robotics for risky interventions and environmental surveillance*, pages 12–14. Citeseer, 2009.

- [7] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, und Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.
- [8] Sunil Arya und David M Mount. Approximate nearest neighbor queries in fixed dimensions. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, pages 271–280. Society for Industrial and Applied Mathematics, 1993.
- [9] Takao Asano, Tetsuo Asano, Leonidas Guibas, John Hershberger, und Hiroshi Imai. Visibility-polygon search and euclidean shortest paths. In *Foundations of Computer Science, 1985., 26th Annual Symposium on*, pages 155–164. IEEE, 1985.
- [10] Franz Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.
- [11] Markus Ax, Lars Kuhnert, Matthias Langer, Jens Schlemper, und Klaus-Dieter Kuhnert. Architecture of an autonomous mini unmanned aerial vehicle based on a commercial platform. In *41st International Symposium on Robotics (ISR) and 6th German Conference on Robotics (ROBOTIK)*, pages 1–6. VDE, 2010.
- [12] Chandrajit L Bajaj, Fausto Bernardini, und Guoliang Xu. Automatic reconstruction of surfaces and scalar fields from 3d scans. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 109–118. ACM, 1995.
- [13] Simon Baker und Shree K Nayar. A theory of catadioptric image formation. In *Computer Vision, 1998. Sixth International Conference on*, pages 35–42. IEEE, 1998.
- [14] Jerome Barraquand und Jean-Claude Latombe. Robot motion planning: A distributed representation approach. *The International Journal of Robotics Research*, 10(6):628–649, 1991.
- [15] Jerome Barraquand und Jean-Claude Latombe. Nonholonomic multibody mobile robots: Controllability and motion planning in the presence

- of obstacles. *Algorithmica*, 10(2-4):121–155, 1993. ISSN 0178-4617. doi: 10.1007/BF01891837. URL <http://dx.doi.org/10.1007/BF01891837>.
- [16] Billur Barshan und Hugh F Durrant-Whyte. Inertial navigation systems for mobile robots. *Robotics and Automation, IEEE Transactions on*, 11(3):328–342, 1995.
- [17] Herbert Bay, Tinne Tuytelaars, und Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.
- [18] Alberto Behar, Jaret Matthews, Frank Carsey, und Jack Jones. Nasa/jpl tumbleweed polar rover. In *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, volume 1. IEEE, 2004.
- [19] R. Bellman und Rand Corporation. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957. ISBN 9780691079516. URL <http://books.google.it/books?id=wdtoPwAACAAJ>.
- [20] Ryad Benosman und Sing Bing Kang. *Panoramic vision: sensors, theory, and applications*. Springer, 2001.
- [21] David Bernstein und Alain Kornhauser. An introduction to map matching for personal navigation assistants. Technical report, New Jersey TIDE Center, 1998.
- [22] J Bohren und Steve Cousins. The smach high-level executive [ros news]. *Robotics & Automation Magazine, IEEE*, 17(4):18–20, 2010.
- [23] Jean-Daniel Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics (TOG)*, 3(4):266–286, 1984.
- [24] Jean-Daniel Boissonnat und Sylvain Lazard. A polynomial-time algorithm for computing a shortest path of bounded curvature amidst moderate obstacles. In *Proceedings of the twelfth annual symposium on Computational geometry*, pages 242–251. ACM, 1996.

- [25] David Bonnafous, Simon Lacroix, und Thierry Siméon. Motion generation for a rover on rough terrains. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 2, pages 784–789. IEEE, 2001.
- [26] Philippe Bonnifait, Jean Laneurit, Clément Fouque, Gérald Dherbomez, et al. Multi-hypothesis map-matching using particle filtering. In *Proceedings of the 16th World Congress for ITS Systems and Services*, pages 1–8, 2009.
- [27] Valérie Boor, Mark H Overmars, und A Frank van der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1018–1023. IEEE, 1999.
- [28] Johann Borenstein und Liqiang Feng. Measurement and correction of systematic odometry errors in mobile robots. *Robotics and Automation, IEEE Transactions on*, 12(6):869–880, 1996.
- [29] Toby P Breckon, Stuart E Barnes, Marcin L Eichner, und Ken Wahren. Autonomous real-time vehicle detection from a medium-level uav. In *Proc. 24th International Conference on Unmanned Air Vehicle Systems*, pages 29–1, 2009.
- [30] Jack E Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems journal*, 4(1):25–30, 1965.
- [31] Oliver Brock und Oussama Khatib. High-speed navigation using the global dynamic window approach. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 1, pages 341–346. IEEE, 1999.
- [32] Richard R Brooks und Sundararaja S Iyengar. *Multi-sensor fusion: fundamentals and applications with software*. Prentice-Hall, Inc., 1998.
- [33] Duane C. Brown. Close-range camera calibration. *Photogrammetric engineering*, 37(8):855–866, 1971.

- [34] John Charles Butcher. *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods*. Wiley-Interscience, 1987.
- [35] Roberto Casas, A Marco, JJ Guerrero, and J Falco. Robust estimator for non-line-of-sight error mitigation in indoor localization. *EURASIP Journal on Applied Signal Processing*, 2006:156–156, 2006.
- [36] Anthony R Cassandra, Leslie Pack Kaelbling, and James A Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. In *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, volume 2, pages 963–972. IEEE, 1996.
- [37] Chu-Song Chen und Wen-Yan Chang. Pose estimation for generalized imaging device via solving non-perspective n point problem. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 2931–2937. IEEE, 2002.
- [38] Chu-Song Chen und Wen-Yan Chang. On pose recovery for generalized visual sensors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(7):848–861, 2004.
- [39] P. Cheng und S. M. LaValle. Reducing metric sensitivity in randomized trajectory design. In *Intelligent Robots and Systems, 2001. Proceedings. IEEE/RSJ International Conference on*, volume 1, pages 43–48. IEEE, 2001.
- [40] R Craig Coulter. Implementation of the pure pursuit path tracking algorithm. Technical report, DTIC Document, 1992.
- [41] Navneet Dalal und Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.
- [42] Belur V Dasarathy. Sensor fusion potential exploitation-innovative architectures and illustrative applications. *Proceedings of the IEEE*, 85(1): 24–38, 1997.

- [43] Pavel Davidson, Jussi Collin, John Raquet, und Jarmo Takala. Application of particle filters for vehicle positioning using road maps. In *23rd International Technical Meeting of the Satellite Division of The Institute of Navigation, Portland, OR*, pages 1653–1661, 2010.
- [44] Geert De Cubber, Daniela Doroftei, Daniel Serrano, Keshav Chintamani, Rui Sabino, und Stephane Ourevitch. The eu-icarus project: developing assistive robotic tools for search and rescue operations. In *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, pages 1–4. IEEE, 2013.
- [45] Alessandro De Luca, Giuseppe Oriolo, und Claude Samson. Feedback control of a nonholonomic car-like robot. In *Robot motion planning and control*, pages 171–253. Springer, 1998.
- [46] Daniel DeMenthon und Larry S Davis. Exact and approximate solutions of the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1100–1105, 1992.
- [47] Daniel F Dementhon und Larry S Davis. Model-based object pose in 25 lines of code. *International journal of computer vision*, 15(1-2):123–141, 1995.
- [48] James Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 2006.
- [49] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [50] Randal Douc und Olivier Cappé. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69. IEEE, 2005.
- [51] Arnaud Doucet, Nando De Freitas, und Neil Gordon. *Sequential Monte Carlo methods in practice*. Springer, 2001.

- [52] Lester E Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics*, pages 497–516, 1957.
- [53] Hugh F Durrant-Whyte. Sensor models and multisensor integration. *The International Journal of Robotics Research*, 7(6):97–113, 1988.
- [54] Herbert Edelsbrunner und Ernst P Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG)*, 13(1):43–72, 1994.
- [55] Baojie Fan, Yingkui Du, Linlin Zhu, und Yandong Tang. The registration of uav down-looking aerial images to satellite images with image entropy and edges. In *Intelligent Robotics and Applications*, pages 609–617. Springer, 2010.
- [56] Hans Jacob S Feder, John J Leonard, und Christopher M Smith. Adaptive mobile robot navigation and mapping. *The International Journal of Robotics Research*, 18(7):650–668, 1999.
- [57] David Ferguson, Aaron Morris, Dirk Haehnel, Christopher Baker, Zachary Omohundro, Carlos Reverte, Scott Thayer, Charles Whittaker, William Whittaker, Wolfram Burgard, et al. An autonomous robotic system for mapping abandoned mines. In *Advances in Neural Information Processing Systems*, page None, 2003.
- [58] S. Finsterwalder und W. Scheufele. *Das Rückwärtseinschneiden im Raum*. Verlag der K. Akademie, 1903. URL <http://books.google.de/books?id=UFUYuAAACAAJ>.
- [59] Martin A Fischler und Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [60] Steven Fortune und Gordon Wilfong. Planning constrained motion. *Annals of Mathematics and Artificial Intelligence*, 3(1):21–82, 1991.

- [61] Clément Fouque und Philippe Bonnifait. Multi-hypothesis map-matching on 3d navigable maps using raw gps measurements. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1498–1503. IEEE, 2010.
- [62] U Franks, H Loose, und C Knoppel. Lane recognition on country roads. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 99–104. IEEE, 2007.
- [63] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, und Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25 (8):930–943, 2003.
- [64] Andreas Geiger, Julius Ziegler, und Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 963–968. IEEE, 2011.
- [65] Joshua Gleason, Ara V Nefian, Xavier Bouysssonousse, Terry Fong, und George Bebis. Vehicle detection from aerial imagery. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2065–2070. IEEE, 2011.
- [66] M Gopi und Shankar Krishnan. A fast and efficient projection-based approach for surface reconstruction. In *Computer Graphics and Image Processing, 2002. Proceedings. XV Brazilian Symposium on*, pages 179–186. IEEE, 2002.
- [67] Neil J Gordon, David J Salmond, und Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, 2, pages 107–113. IET, 1993.
- [68] E. W. Grafarend, E. Lohse, und B. Schaffrin. Dreidimensionaler rückwärtsschnitt teil i: Die projektiven gleichungen. *Zeitschrift für Vermessungswesen, Geodätisches Institut, Universität Stuttgart*, pages 1–37, 1989.

- [69] John P Grotzinger, Joy Crisp, Ashwin R Vasavada, Robert C Anderson, Charles J Baker, Robert Barry, David F Blake, Pamela Conrad, Kenneth S Edgett, Bobak Ferdowski, et al. Mars science laboratory mission and science investigation. *Space Science Reviews*, 170(1-4):5–56, 2012.
- [70] J.A. Grunert. *Archiv der Mathematik und Physik*. Number Bd. 1. C. A. Koch, 1841. URL <http://books.google.de/books?id=GhIAAAAAMAAJ>.
- [71] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forsell, Jonas Jansson, Rickard Karlsson, und P-J Nordlund. Particle filters for positioning, navigation, and tracking. *Signal Processing, IEEE Transactions on*, 50(2):425–437, 2002.
- [72] John W Hager, James F Behensky, und Brad W Drew. The universal grids: Universal transverse mercator (utm) and universal polar stereographic (ups). edition 1. Technical report, DTIC Document, 1989.
- [73] David L Hall und James Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.
- [74] Bert M Haralick, Chung-Nan Lee, Karsten Ottenberg, und Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356, 1994.
- [75] David Harel. Statecharts: A visual formalism for complex systems. *Science of computer programming*, 8(3):231–274, 1987.
- [76] Peter E Hart, Nils J Nilsson, und Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2):100–107, 1968.
- [77] Mike Hazas und Andy Hopper. Broadband ultrasonic location systems for improved indoor positioning. *Mobile Computing, IEEE Transactions on*, 5(5):536–547, 2006.
- [78] Janne Heikkila und Olli Silvén. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recogni-*

- tion, 1997. *Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106–1112. IEEE, 1997.
- [79] M Himmelsbach, F von Hundelshausen, T Luettel, M Manz, A Mueller, S Schneider, und HJ Wuensche. Team mucar-3 at c-elrob 2009. In *Proceedings of 1st workshop on field robotics, civilian European land robot trial, University of Oulu, Oulu, Finland. ISBN*, pages 978–951, 2009.
- [80] Bernhard Hofmann-Wellenhof, Herbert Lichtenegger, und James Collins. Global positioning system. theory and practice. *Global Positioning System. Theory and practice.*, by Hofmann-Wellenhof, B.; Lichtenegger, H.; Collins, J.. Springer, Wien (Austria), 1993, 347 p., ISBN 3-211-82477-4, Price DM 79.00. ISBN 0-387-82477-4 (USA)., 1, 1993.
- [81] Jeroen D Hol, Thomas B Schon, und Fredrik Gustafsson. On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pages 79–82. IEEE, 2006.
- [82] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, und Werner Stuetzle. *Surface reconstruction from unorganized points*, volume 26, 2. ACM, 1992.
- [83] Armin Hornung, Kai M Wurm, und Maren Bennewitz. Humanoid robot localization in complex indoor environments. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1690–1695. IEEE, 2010.
- [84] David Hsu, Lydia E Kavraki, Jean-Claude Latombe, Rajeev Motwani, Stephen Sorkin, et al. On finding narrow passages with probabilistic roadmap planners. In *Robotics: The Algorithmic Perspective: 1998 Workshop on the Algorithmic Foundations of Robotics*, pages 141–154, 1998.
- [85] Congwei Hu, Wu Chen, Yongqi Chen, und Dajie Liu. Adaptive kalman filtering for vehicle navigation. *Positioning*, 1(04):0, 2009.
- [86] Yong K Hwang und Narendra Ahuja. Gross motion planning - a survey. *ACM Computing Surveys (CSUR)*, 24(3):219–291, 1992.

- [87] Félix Ingrand, Simon Lacroix, Solange Lemai-Chenevier, und Frederic Py. Decisional autonomy of planetary rovers. *Journal of Field Robotics*, 24(7):559–580, 2007.
- [88] Larry D Jackel, Eric Krotkov, Michael Perschbacher, Jim Pippine, und Chad Sullivan. The darpa lagr program: Goals, challenges, methodology, and phase i results. *Journal of Field Robotics*, 23(11-12):945–973, 2006.
- [89] Paul Jacobs und John Canny. Planning smooth paths for mobile robots. In *Nonholonomic Motion Planning*, pages 271–342. Springer, 1993.
- [90] Paul Jacobs, J-P Laumond, und M Taix. Efficient motion planners for nonholonomic mobile robots. In *Intelligent Robots and Systems' 91. 'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*, pages 1229–1235. IEEE, 1991.
- [91] Simon J Julier und Hugh F Durrant-Whyte. On the role of process models in autonomous land vehicle navigation systems. *Robotics and Automation, IEEE Transactions on*, 19(1):1–14, 2003.
- [92] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.
- [93] Lydia E Kavraki, Petr Svestka, J-C Latombe, und Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580, 1996.
- [94] Alonzo Kelly. *An intelligent, predictive control approach to the high-speed cross-country autonomous navigation problem*. PhD thesis, Carnegie Mellon University, 1995.
- [95] Alonzo Kelly und Anthony Stentz. Rough terrain autonomous mobility - part 2: An active vision, predictive control approach. *Autonomous Robots*, 5(2):163–198, 1998.
- [96] Wuk Kim, Gyu-In Jee, und JangGyu Lee. Efficient use of digital road map in various positioning for its. In *Position Location and Navigation Symposium, IEEE 2000*, pages 170–176. IEEE, 2000.

- [97] Genshiro Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25, 1996.
- [98] Achim Klenke. *Wahrscheinlichkeitstheorie*, volume 1. Springer, 2006.
- [99] Laurent Kneip und Paul Furgale. Opengv: A unified and generalized approach to real-time calibrated geometric vision. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014.
- [100] Laurent Kneip, Davide Scaramuzza, und Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2969–2976. IEEE, 2011.
- [101] Moritz Koblenzer. Konzeption und Implementierung einer verteilten, ereignisbasierten Softwarearchitektur für die mobile Robotik. *Diplomarbeit*, 2009.
- [102] James J Kuffner und Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 2, pages 995–1001. IEEE, 2000.
- [103] Klaus-Dieter Kuhnert und Lars Kuhnert. Light-weight sensor package for precision 3d measurement with micro uavs. In *UAV-g, Rostock, Germany*, pages 1710–1715, 2012.
- [104] Lars Kuhnert und Klaus-Dieter Kuhnert. Sensor-fusion based real-time 3d outdoor scene reconstruction and analysis on a moving mobile outdoor robot. *Künstliche Intelligenz (KI) - German Journal on Artificial Intelligence, Special Issue 'Off-Road-Robotik'*, 25(2):117–123, 2011.
- [105] Lars Kuhnert, Markus Ax, Matthias Langer, Duong-Van Nguyen, und Klaus-Dieter Kuhnert. Absolute high-precision localisation of an unmanned ground vehicle by using real-time aerial video imagery for geo-referenced orthophoto registration. In *Autonome Mobile Systeme 2009*, pages 145–152. Springer, 2009.

- [106] Lars Kuhnert, Duong-Van Nguyen, Stefan Thamke, und Klaus Dieter Kuhnert. Autonomous explorative outdoor path planning. *IASTED International Conference on Robotics (Robo 2011), Pittsburgh, USA*, 2011.
- [107] Lars Kuhnert, Klaus Müller, Markus Ax, und Klaus-Dieter Kuhnert. Object localization on agricultural areas using an autonomous team of cooperating ground and air robots. In *International Conference of Agricultural Engineering CIGR-Ageng2012, Valencia, Spain*, 2012.
- [108] Lars Kuhnert, Stefan Thamke, Markus Ax, Duong-Van Nguyen, und Klaus-Dieter Kuhnert. Cooperation in heterogeneous groups of autonomous robots. In *Mechatronics and Automation (ICMA), 2012 International Conference on*, pages 1710–1715. IEEE, 2012.
- [109] Lars Kuhnert, Benjamin Meier, und Klaus-Dieter Kuhnert. Probabilistic approach to self-localization for autonomous mobile outdoor robotics based on hybrid map knowledge. In *16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC), The Hague, The Netherlands*. IEEE, 2013.
- [110] Lars Kuhnert, Duong-Van Nguyen, und Klaus-Dieter Kuhnert. Contactless obstacle penetrability classification for mobile outdoor robotics. In *2nd RSS Workshop on Robots in Clutter, Berlin, Germany*, 2013.
- [111] James J Kumler und Martin L Bauer. Fish-eye lens designs and their relative performance. In *International Symposium on Optical Science and Technology*, pages 360–369. International Society for Optics and Photonics, 2000.
- [112] W. Kutta. Beitrag zur näherungsweise Integration totaler Differentialgleichungen. *Zeit. Math. Phys.*, 46:435–53, 1901.
- [113] Alberto Lacaze, Karl Murphy, und Mark DelGiorno. Autonomous mobility for the demo iii experimental unmanned vehicles. In *in Assoc. for Unmanned Vehicle Systems Int. Conf. on Unmanned Vehicles (AUVSI 02*. Citeseer, 2002.

- [114] Florent Lamiroux und Jean-Paul Laumond. On the expected complexity of random path planning. In *IEEE International Conference on Robotics and Automation*, pages 3014–3019. INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE), 1996.
- [115] Pierre Lamon, Sascha Kolski, und Roland Siegwart. The smarter-vehicle for fully autonomous navigation and mapping in outdoor environments. In *Proceedings of CLAWAR*, 2006.
- [116] Marco Langerwisch, Marko Reimer, Matthias Hentschel, und Bernardo Wagner. Control of a semi-autonomous ugv using lossy low-bandwidth communication. In *The Second IFAC Symposium on Telematics Applications (TA)*. Timisoara, Romania, 2010.
- [117] Jean-Claude Latombe. A fast path planner for a car-like indoor mobile robot. In *AAAI*, pages 659–665, 1991.
- [118] Jean-Paul Laumond. Feasible trajectories for mobile robots with kinematic and environment constraints. In *Intelligent Autonomous Systems, An International Conference*, pages 346–354, Amsterdam, The Netherlands, The Netherlands, 1987. North-Holland Publishing Co.
- [119] Jean-Paul Laumond. Finding collision-free smooth trajectories for a non-holonomic mobile robot. In *IJCAI*, pages 1120–1123, 1987.
- [120] Steven M LaValle. Rapidly-exploring random trees a new tool for path planning. Technical report, TR 98-11, Computer Science Dept., Iowa State Univ., 1998.
- [121] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.
- [122] Steven M LaValle und James J Kuffner. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001.
- [123] Johannes Leidheiser. Erzeugung eines texturierten 3D-Modells aus 3D-Tiefendaten und 2D-Bilddaten für die lokale Kartierung mit dem Außenbereichsroboter AMOR. *Diplomarbeit*, 2009.

- [124] Vincent Lepetit, Francesc Moreno-Noguer, und Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009.
- [125] Kenneth Levenberg. A method for the solution of certain problems in least squares. *Quarterly of applied mathematics*, 2:164–168, 1944.
- [126] Qingquan Li, Zhixiang Fang, Hanwu Li, und Hui Xiao. Integrating gps, gyro, vehicle speed sensor, and digital map to provide accurate and real-time position in an intelligent navigation system. In *MIPPR 2005 Geospatial Information, Data Mining, and Applications*, pages 60451F–60451F. International Society for Optics and Photonics, 2005.
- [127] Yuping Lin und Gérard Medioni. Map-enhanced uav image sequence registration and synchronization of multiple image sequences. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007.
- [128] Seppo Linnainmaa, David Harwood, und Larry S Davis. Pose determination of a three-dimensional object using triangle pairs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(5):634–647, 1988.
- [129] William E Lorensen und Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM Siggraph Computer Graphics*, volume 21, 4, pages 163–169. ACM, 1987.
- [130] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.
- [131] Tomas Lozano-Perez. Spatial planning: A configuration space approach. *Computers, IEEE Transactions on*, 100(2):108–120, 1983.
- [132] C-P Lu, Gregory D Hager, und Eric Mjolsness. Fast and globally convergent pose estimation from video images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(6):610–622, 2000.
- [133] Alexei A Makarenko, Stefan B Williams, Frederic Bourgault, und Hugh F Durrant-Whyte. An experiment in integrated exploration. In *Intelligent*

*Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 534–539. IEEE, 2002.

- [134] Donald W Marquardt. An algorithm for least-squares estimation of non-linear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11(2):431–441, 1963.
- [135] Zoltan Csaba Marton, Radu Bogdan Rusu, und Michael Beetz. On fast surface reconstruction methods for large and noisy datasets. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 12-17 2009.
- [136] Benjamin Meier. Entwicklung eines Systems zur globalen Navigation autonomer Außenbereichsroboter auf Basis von Selbstlokalisierung auf topologischen Kartendaten. *Diplomarbeit*, 2012.
- [137] Robert Mencl und Heinrich Müller. Interpolation and approximation of surfaces from three-dimensional scattered data points. In *Scientific Visualization Conference, 1997*, pages 223–223. IEEE, 1997.
- [138] Robert Mencl und Heinrich Müller. Graph-based surface reconstruction using structures in scattered point sets. In *Computer Graphics International, 1998. Proceedings*, pages 298–311. IEEE, 1998.
- [139] E. L. Merritt. Explicit three-point resection in space. *Photogrammetric Engineering*, XV(4):649–655, 1949.
- [140] Branislav Micusik und Tomas Pajdla. Estimation of omnidirectional camera model from epipolar geometry. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–485. IEEE, 2003.
- [141] Branislav Micušik, Daniel Martinec, und Tomáš Pajdla. 3d metric reconstruction from uncalibrated omnidirectional images. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, volume 1, pages 545–550, 2003.

- [142] Fumio Miyazaki und Suguru Arimoto. Sensory feedback based on the artificial potential for robots. In *Proceedings 9th IFAC, Budapest, Hungary*, 1984.
- [143] AH Mohamed und KP Schwarz. Adaptive kalman filtering for ins/gps. *Journal of geodesy*, 73(4):193–203, 1999.
- [144] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.
- [145] Nils J Nilsson. *Principles of artificial intelligence*. Springer, 1982.
- [146] David Nistér, Oleg Naroditsky, und James Bergen. Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–652. IEEE, 2004.
- [147] David Nistér, Oleg Naroditsky, und James Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20, 2006.
- [148] Horoshi Noborio, Tomohide Naniwa, und Suguru Arimoto. A feasible motion-planning algorithm for a mobile robot based on a quadtree representation. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 327–332. IEEE, 1989.
- [149] Masafumi Noda, Tomokazu Takahashi, Daisuke Deguchi, Ichiro Ide, Hiroshi Murase, Yoshiko Kojima, und Takashi Naito. Vehicle ego-localization by matching in-vehicle camera images to an aerial image. In *Computer Vision–ACCV 2010 Workshops*, pages 163–173. Springer, 2011.
- [150] Dragan Obradovic, Henning Lenz, und Markus Schupfner. Fusion of map and sensor data in a modern car navigation system. *Journal of VLSI signal processing systems for signal, image and video technology*, 45(1-2): 111–122, 2006.
- [151] VV Pavlov und AN Voronin. The method of potential functions for coding constraints of the external space in an intelligent mobile robot. *Soviet Automatic Control*, 17(6):45–51, 1984.

- [152] Hwei Peng und Masayoshi Tomizuka. Optimal preview control for vehicle lateral guidance. *California Partners for Advanced Transit and Highways (PATH)*, 1991.
- [153] Bishnu P Phuyal. Method and use of aggregated dead reckoning sensor and gps data for map matching. In *Proceedings of the 15th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2002)*, pages 430–437, 2001.
- [154] Oliver Pink, Frank Moosmann, und Alexander Bachmann. Visual features for vehicle localization and ego-motion estimation. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 254–260. IEEE, 2009.
- [155] Chris Prahacs, Aaron Saudners, Matthew K Smith, Dave McMordie, und Martin Buehler. Towards legged amphibious mobile robotics. *Proceedings of the Canadian Engineering Education Association*, 2011.
- [156] William K Pratt et al. *Digital image processing*. Wiley, New York, 1991.
- [157] Franco P Preparata und Michael Ian Shamos. *Computational geometry: an introduction*. Springer-Verlag, 1985.
- [158] Long Quan und Zhongdan Lan. Linear n-point camera pose determination. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(8):774–780, 1999.
- [159] Mohammed A Quddus, Washington Yotto Ochieng, Lin Zhao, und Robert B Noland. A general map matching algorithm for transport telematics applications. *GPS solutions*, 7(3):157–167, 2003.
- [160] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, und Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5, 2009.
- [161] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, Rob Playter, et al. Bigdog, the rough-terrain quadruped robot. In *Proceedings of the 17th World Congress*, pages 10823–10825, 2008.

- [162] Rajesh Rajamani. *Vehicle dynamics and control*. Springer, 2011.
- [163] James Reeds und Lawrence Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2): 367–393, 1990.
- [164] Stephan Roth, Bradley Hamner, Sanjiv Singh, und Myung Hwangbo. Results in combined route traversal and collision avoidance. In *Field and Service Robotics*, pages 491–504. Springer, 2006.
- [165] Carl Runge. Über die numerische auflösung von differentialgleichungen. *Mathematische Annalen*, 46(2):167–178, 1895.
- [166] Davide Scaramuzza, Agostino Martinelli, und Roland Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5695–5701. IEEE, 2006.
- [167] Davide Scaramuzza, Ahad Harati, und Roland Siegwart. Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 4164–4169. IEEE, 2007.
- [168] Jens Schlemper, Lars Kuhnert, Markus Ax, und Klaus-Dieter Kuhnert. Development of a high speed 3d laser measurement system for outdoor robotics. In *Research and Education in Robotics - EUROBOT 2011*, pages 277–287. Springer, 2011.
- [169] Frank E Schneider und Dennis Wildermuth. Results of the european land robot trial and their usability for benchmarking outdoor robot systems. In *Towards Autonomous Robotic Systems*, pages 408–409. Springer, 2011.
- [170] Frank E Schneider, Dennis Wildermuth, Bernd Brüggemann, und Timo Röhling. European land robot trial (elrob) towards a realistic benchmark for outdoor robotics. 2010.
- [171] Derik Schroeter und Paul Newman. On the cross-calibration of a camera and a 3d-laser range finder. In *Proceedings of the 2nd SEAS DTC Technical Conference, Edinburgh, 2007*.

- [172] Robin Schubert, Eric Richter, und Gerd Wanielik. Comparison and evaluation of advanced motion models for vehicle tracking. In *Information Fusion, 2008 11th International Conference on*, pages 1–6. IEEE, 2008.
- [173] Wolfgang Seemann und Klaus-Dieter Kuhnert. Localisation for an outdoor robot by optically measuring the ground movement. In *Proceedings of the 13th IASTED International Conference on Robotics and Applications*, pages 113–118. ACTA Press, 2007.
- [174] Robin S Sharp. Driver steering control and a new perspective on car handling qualities. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 219(10): 1041–1051, 2005.
- [175] Sanjiv Singh, Reid Simmons, Trey Smith, Anthony Stentz, Vandi Verma, Alex Yahja, und Kurt Schwehr. Recent progress in local and global traversability for planetary rovers. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 2, pages 1194–1200. IEEE, 2000.
- [176] Randall W Smith. *Department of Defense World Geodetic System 1984: its definition and relationships with local geodetic systems*. Defense Mapping Agency, 1987.
- [177] Jarrod M Snider. Automatic steering methods for autonomous automobile path tracking. *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08*, 2009.
- [178] Kil-Ho Son, Youngbae Hwang, und In So Kweon. Uav global pose estimation by matching forward-looking aerial images with satellite images. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3880–3887. IEEE, 2009.
- [179] Peter Steinhaus und Rüdiger Dillmann. Aufbau und modellierung des rosi scanners zur 3d-tiefenbildakquisition. In *Autonome Mobile Systeme 2003*, pages 42–52. Springer, 2003.

- [180] Anthony Stentz und Martial Hebert. A complete navigation system for goal acquisition in unknown environments. *Autonomous Robots*, 2(2): 127–145, 1995.
- [181] Anthony Stentz, John Bares, Thomas Pilarski, und David Stager. The crusher system for autonomous navigation. *AUVSIs Unmanned Systems North America*, 3, 2007.
- [182] Ioan Alexandru Sucas und Lydia E Kavraki. On the implementation of single-query sampling-based motion planners. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2005–2011. IEEE, 2010.
- [183] Héctor J Sussmann und Guoqing Tang. Shortest paths for the reedshepp car: a worked out example of the use of geometric techniques in nonlinear optimal control. *Rutgers Center for Systems and Control Technical Report*, 10:1–71, 1991.
- [184] Petr Svestka und Mark H Overmars. Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 2, pages 1631–1636. IEEE, 1995.
- [185] Tomáš Svoboda und Tomáš Pajdla. Epipolar geometry for central catadioptric cameras. *International Journal of Computer Vision*, 49(1): 23–37, 2002.
- [186] Michael J Swain und Dana H Ballard. Indexing via color histograms. In *Active Perception and Robot Vision*, pages 261–273. Springer, 1992.
- [187] Richard Szeliski und David Tonnesen. *Surface modeling with oriented particle systems*, volume 26, 2. ACM, 1992.
- [188] Ceryen Tan, Tsai Hong, Tommy Chang, und Michael Shneier. Color model-based real-time learning for road following. In *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pages 939–944. IEEE, 2006.

- [189] Demetri Terzopoulos und Dimitri Metaxas. Dynamic 3d models with local and global deformations: Deformable superquadrics. In *Computer Vision, 1990. Proceedings, Third International Conference on*, pages 606–615. IEEE, 1990.
- [190] Demetri Terzopoulos, Andrew Witkin, und Michael Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motion. *Artificial intelligence*, 36(1):91–123, 1988.
- [191] Sebastian Thrun, Dirk Hahnel, David Ferguson, D Montemerlo, Rudolph Triebel, Wolfram Burgard, Christopher Baker, Zachary Omohundro, Scott Thayer, und William Whittaker. A system for volumetric robotic mapping of abandoned mines. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 3, pages 4270–4275. IEEE, 2003.
- [192] Sebastian Thrun, Wolfram Burgard, Dieter Fox, et al. *Probabilistic robotics*, volume 1. MIT press Cambridge, 2005.
- [193] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9):661–692, 2006.
- [194] A Tikanmaki und J Roning. Development of mörri, a high performance and modular outdoor robot. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 1441–1446. IEEE, 2009.
- [195] Rafael Toledo-Moreo, David Bétaille, François Peyret, und Jean Laneurrit. Fusing gnss, dead-reckoning, and enhanced maps for road vehicle lane-level navigation. *Selected Topics in Signal Processing, IEEE Journal of*, 3(5):798–809, 2009.
- [196] Rafael Toledo-Moreo, David Bétaille, und François Peyret. Lane-level integrity provision for navigation and map matching with gnss, dead reckoning, and enhanced maps. *Intelligent Transportation Systems, IEEE Transactions on*, 11(1):100–112, 2010.

- [197] Aksel Andreas Transeth, Remco I Leine, Christoph Glocker, Kristin Ytterstad Pettersen, und Pal Liljeback. Snake robot obstacle-aided locomotion: Modeling, simulations, and experiments. *Robotics, IEEE Transactions on*, 24(1):88–104, 2008.
- [198] Roger Y Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344, 1987.
- [199] Chris Urmson, Joshua Anhalt, Michael Clark, Tugrul Galatali, Juan P Gonzalez, Jay Gowdy, Alexander Gutierrez, Sam Harbaugh, Matthew Johnson-Roberson, Hiroki Kato, et al. High speed navigation of unrehearsed terrain: Red team technology for grand challenge 2004. *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-04-37*, 2004.
- [200] Chris Urmson, Chris Baker, John Dolan, Paul Rybski, Bryan Salesky, William Whittaker, Dave Ferguson, und Michael Darms. Autonomous driving in traffic: Boss and the urban challenge. *AI Magazine*, 30(2):17, 2009.
- [201] Remco C Veltkamp. *Closed object boundaries from scattered points*, volume 885. Springer, 1994.
- [202] Remco C Veltkamp. Boundaries through scattered points of unknown density. *Graphical Models and Image Processing*, 57(6):441–452, 1995.
- [203] Paul Viola und Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.
- [204] Felix Von Hundelshausen, Michael Himmelsbach, Falk Hecker, Andre Mueller, und Hans-Joachim Wuensche. Driving with tentacles: Integral structures for sensing and motion. *Journal of Field Robotics*, 25(9):640–673, 2008.

- [205] Jinling Wang, Mike Stewart, and Maria Tsakiri. Kinematic gps positioning with adaptive kalman filtering techniques. In *Advances in Positioning and Reference Frames*, pages 389–394. Springer, 1998.
- [206] David Wettergreen, Deepak Bapna, Mark Maimone, and Geb Thomas. Developing nomad for robotic exploration of the atacama desert. *Robotics and Autonomous Systems*, 26(2):127–148, 1999.
- [207] Christopher E White, David Bernstein, and Alain L Kornhauser. Some map matching algorithms for personal navigation assistants. *Transportation Research Part C: Emerging Technologies*, 8(1):91–108, 2000.
- [208] Franklin E White. Data fusion lexicon. Technical report, DTIC Document, 1991.
- [209] SJ Wright und J Nocedal. *Numerical optimization*, volume 2. Springer New York, 1999.
- [210] Oliver Wulf und Bernardo Wagner. Fast 3d scanning methods for laser measurement systems. In *International conference on control systems and computer science (CSCS14)*, pages 2–5, 2003.
- [211] De Xu, Zhi-Guo Tu, und Min Tan. Study on visual positioning based on homography for indoor mobile robot. *Acta Automatica Sinica*, 31(3): 464–469, 2005.
- [212] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA '97., Proceedings., 1997 IEEE International Symposium on*, pages 146–151. IEEE, 1997.
- [213] Libo Yang und Steven M LaValle. A framework for planning feedback motion strategies based on a random neighborhood graph. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 1, pages 544–549. IEEE, 2000.
- [214] Meng Yu. *Improved positioning of land vehicle in ITS using digital map and other accessory information*. PhD thesis, The Hong Kong Polytechnic University, 2006.

- [215] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11): 1330–1334, 2000.
- [216] Tao Zhao und Ram Nevatia. Car detection in low resolution aerial images. *Image and Vision Computing*, 21(8):693–703, 2003.
- [217] Yinqiang Zheng, Yubin Kuang, Shigeki Sugimoto, Kalle Åström, und Masatoshi Okutomi. Revisiting the pnp problem: A fast, general and optimal solution. In *Proceedings of IEEE International Conference on Computer Vision (ICCV2013)*, pages 2344–2351, 2013.