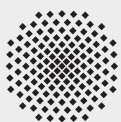
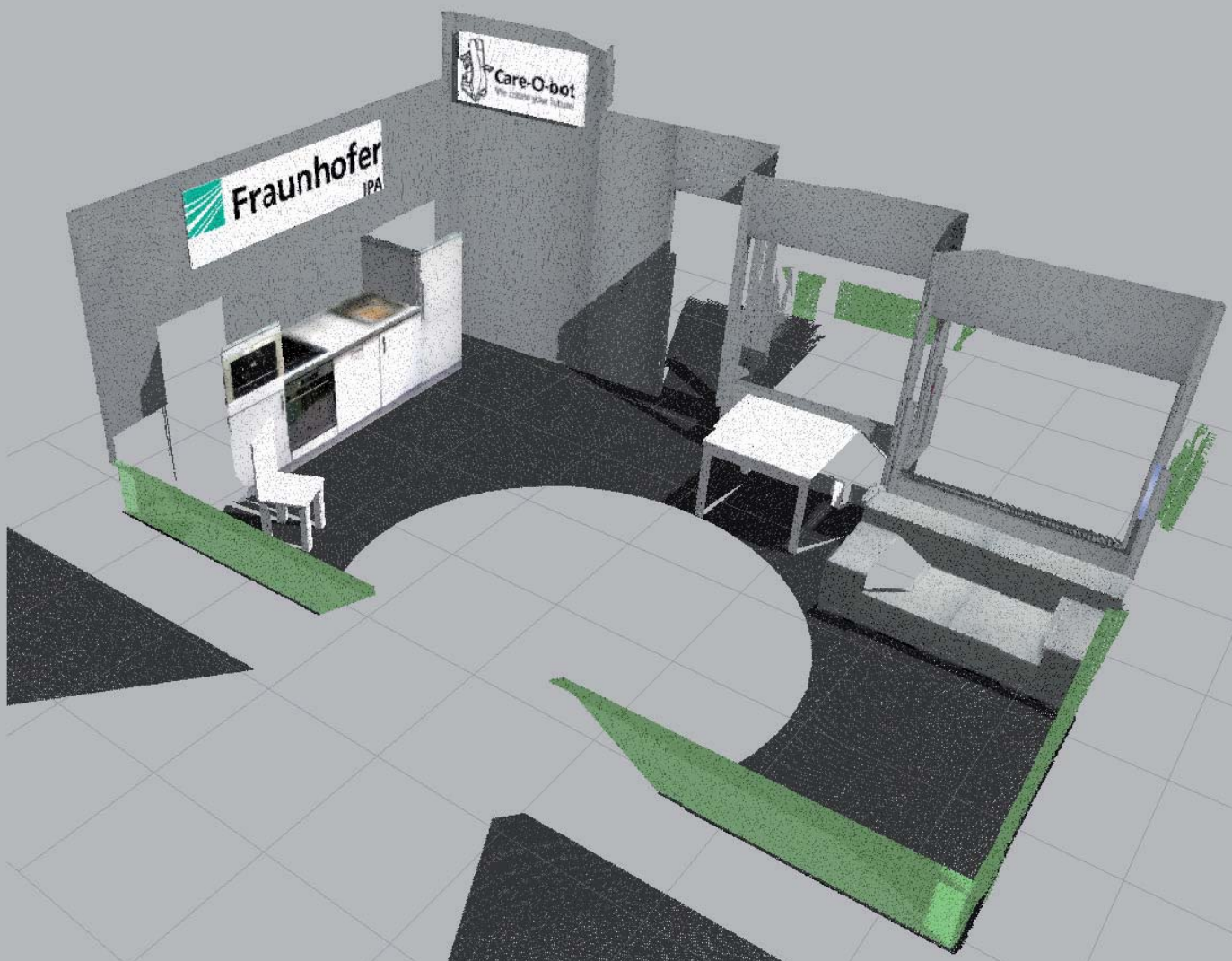


GEORG ARBEITER

3-D-Umgebungserfassung für teil-autonome mobile Roboter



Herausgeber:

Univ.-Prof. Dr.-Ing. Thomas Bauernhansl

Univ.-Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl

Univ.-Prof. a. D. Dr.-Ing. Prof. E.h. Dr.-Ing. E.h. Dr. h.c. mult. Engelbert Westkämper

Georg Arbeiter

**3-D-Umgebungserfassung für teil-autonome
mobile Roboter**

Kontaktadresse:

Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, Stuttgart
Nobelstraße 12, 70569 Stuttgart
Telefon 07 11 9 70-00, Telefax 07 11 9 70-13 99
info@ipa.fraunhofer.de, www.ipa.fraunhofer.de

STUTTGARTER BEITRÄGE ZUR PRODUKTIONSFORSCHUNG**Herausgeber:**

Univ.-Prof. Dr.-Ing. Thomas Bauernhansl
Univ.-Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl
Univ.-Prof. a. D. Dr.-Ing. Prof. E.h. Dr.-Ing. E.h. Dr. h.c. mult. Engelbert Westkämper

Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, Stuttgart
Institut für Industrielle Fertigung und Fabrikbetrieb (IFF) der Universität Stuttgart
Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW)
der Universität Stuttgart

Titelbild: © Georg Arbeiter

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISSN: 2195-2892

ISBN (Print): 978-3-8396-0674-2

D 93

Zugl.: Stuttgart, Univ., Diss., 2013

Druck: Mediendienstleistungen des Fraunhofer-Informationszentrum Raum und Bau IRB, Stuttgart
Für den Druck des Buches wurde chlor- und säurefreies Papier verwendet.

© by **FRAUNHOFER VERLAG**, 2014

Fraunhofer-Informationszentrum Raum und Bau IRB
Postfach 80 04 69, 70504 Stuttgart
Nobelstraße 12, 70569 Stuttgart
Telefon 07 11 9 70-25 00
Telefax 07 11 9 70-25 08
E-Mail verlag@fraunhofer.de
URL <http://verlag.fraunhofer.de>

Alle Rechte vorbehalten

Dieses Werk ist einschließlich aller seiner Teile urheberrechtlich geschützt. Jede Verwertung, die über die engen Grenzen des Urheberrechtsgesetzes hinausgeht, ist ohne schriftliche Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Speicherung in elektronischen Systemen.

Die Wiedergabe von Warenbezeichnungen und Handelsnamen in diesem Buch berechtigt nicht zu der Annahme, dass solche Bezeichnungen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und deshalb von jedermann benutzt werden dürften. Soweit in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z.B. DIN, VDI) Bezug genommen oder aus ihnen zitiert worden ist, kann der Verlag keine Gewähr für Richtigkeit, Vollständigkeit oder Aktualität übernehmen.

GELEITWORT DER HERAUSGEBER

Produktionswissenschaftliche Forschungsfragen entstehen in der Regel im Anwendungszusammenhang, die Produktionsforschung ist also weitgehend erfahrungsbasiert. Der wissenschaftliche Anspruch der „Stuttgarter Beiträge zur Produktionsforschung“ liegt unter anderem darin, Dissertation für Dissertation ein übergreifendes ganzheitliches Theoriegebäude der Produktion zu erstellen.

Die Herausgeber dieser Dissertations-Reihe leiten gemeinsam das Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA und jeweils ein Institut der Fakultät für Konstruktions-, Produktions- und Fahrzeugtechnik an der Universität Stuttgart.

Die von ihnen betreuten Dissertationen sind der marktorientierten Nachhaltigkeit verpflichtet, ihr Ansatz ist systemisch und interdisziplinär. Die Autoren bearbeiten anspruchsvolle Forschungsfragen im Spannungsfeld zwischen theoretischen Grundlagen und industrieller Anwendung.

Die „Stuttgarter Beiträge zur Produktionsforschung“ ersetzt die Reihen „IPA-IAO Forschung und Praxis“ (Hrsg. H.J. Warnecke / H.-J. Bullinger / E. Westkämper / D. Spath) bzw. ISW Forschung und Praxis (Hrsg. G. Stute / G. Pritschow / A. Verl). In den vergangenen Jahrzehnten sind darin über 800 Dissertationen erschienen.

Der Strukturwandel in den Industrien unseres Landes muss auch in der Forschung in einen globalen Zusammenhang gestellt werden. Der reine Fokus auf Erkenntnisgewinn ist zu eindimensional. Die „Stuttgarter Beiträge zur Produktionsforschung“ zielen also darauf ab, mittelfristig Lösungen für den Markt anzubieten. Daher konzentrieren sich die Stuttgarter produktionstechnischen Institute auf das Thema ganzheitliche Produktion in den Kernindustrien Deutschlands. Die leitende Forschungsfrage der Arbeiten ist: Wie können wir nachhaltig mit einem hohen Wertschöpfungsanteil in Deutschland für einen globalen Markt produzieren?

Wir wünschen den Autoren, dass ihre „Stuttgarter Beiträge zur Produktionsforschung“ in der breiten Fachwelt als substanziell wahrgenommen werden und so die Produktionsforschung weltweit voranbringen.

Alexander Verl

Thomas Bauernhansl

Engelbert Westkämper

3-D-Umgebungserfassung für teil-autonome mobile Roboter

Von der Fakultät für Konstruktions-, Produktions- und Fahrzeugtechnik
der Universität Stuttgart zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von

Georg Arbeiter

aus Münchberg

Hauptberichter: Univ.-Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl

Mitberichter: Univ.-Prof. Dr.-Ing. Heinz Wörn

Tag der mündlichen Prüfung: 5.11.2013

Institut für Steuerungstechnik der Werkzeugmaschinen
und Fertigungseinrichtungen der Universität Stuttgart

2014

Vorwort des Verfassers

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Produktionstechnik und Automatisierung der Fraunhofer Gesellschaft in Stuttgart. Mein besonderer Dank gilt Herrn Professor Dr.-Ing. Dr. h.c. Alexander Verl für die Unterstützung meiner wissenschaftlichen Arbeit und die Übernahme des Hauptberichts. Herrn Professor Dr.-Ing. Heinz Wörn danke ich für die Übernahme des Mitberichts zu dieser Arbeit.

Für die Durchsicht der Arbeit und der daraus entstandenen Anregungen und Diskussionen danke ich Frau Dr. Birgit Graf, Herrn Richard Bormann und Frau Luzia Schuhmacher. Des Weiteren danke ich den von mir betreuten Studenten, die im Rahmen von Studien- und Diplomarbeiten zum Gelingen dieser Arbeit beigetragen haben. Besonderer Dank gilt dabei Steffen Fuchs und Joshua Hampp, die durch großes Engagement und gründliches Arbeiten einen entscheidenden Beitrag geleistet haben.

Ich danke meinen Kollegen für die gute Zusammenarbeit. Konstruktive Diskussionen und kritische Fragen haben geholfen, die Arbeit in die richtige Richtung zu lenken. Für Motivation und Aufmunterung auch in kritischen Fragen danke ich Christian Connette und Jan Fischer.

Meiner Frau Melanie danke ich für die Unterstützung und das Verständnis, welche sie mir in den vergangenen sechs Jahren entgegengebracht hat.

Zusammenfassung

Die Servicerobotik hat in den letzten Jahren sowohl durch kostengünstige Sensorik und Aktorik als auch durch verbesserte Algorithmen einen großen Schritt nach vorne gemacht. Dabei hat sich gerade die Wahrnehmung als eine Schlüsseltechnologie für eine erfolgreiche Weiterentwicklung herauskristallisiert. Leistungsfähige 3-D-Kameras und effiziente Verfahren zur Sensordatenverarbeitung ermöglichen es Robotern, ihre Umwelt wahrzunehmen, zu interpretieren und darauf basierend Handlungen abzuleiten. Trotz dieser Fortschritte ist ein voll-autonomer Betrieb von mobilen Robotern in für Menschen gemachten Umgebungen mittelfristig nicht möglich. Ein Hauptgrund dafür ist die Vielzahl nicht vorhersehbarer Ausnahmefälle, mit denen ein Roboter in einem nur teilweise bekannten und dynamischen Umfeld konfrontiert wird.

Diese Arbeit behandelt die Erfassung der Umwelt mit 3-D-Kameras im teil-autonomen Betrieb. Ziel ist es, ein Umgebungsmodell zu erzeugen, das sowohl für die autonomen Funktionen des Roboters wie Navigation oder Manipulation verwendet werden kann als auch eine Hilfe für einen menschlichen Teleoperator bietet. Dadurch stellen sich besondere Anforderungen an die verwendete Repräsentation. Diese muss sowohl vom Roboter als auch vom Bediener verstanden werden. Weiterhin ist die Umwandlung der Sensordaten in die Repräsentation durch ein geeignetes Verfahren ein Kernpunkt dieser Arbeit. Dieses Verfahren muss nicht nur ein Modell der Umwelt mit ausreichender Genauigkeit erzeugen, sondern auch eine entsprechende Verarbeitungsgeschwindigkeit für den Einsatz auf einem Roboter ermöglichen.

Nach der Aufstellung von Anforderungen wird ein Verfahren zur 3-D-Umgebungserfassung konzipiert und umgesetzt. Dieses erzeugt ein hybrides Umgebungsmodell, bestehend aus einer Punkte- und einer Geometriekarte. Die Punktrepräsentation kann vom Roboter für die Hindernisvermeidung während Navigation und Manipulation verwendet werden, besitzt jedoch ein großes Datenvolumen und lässt sich schwer visualisieren. Deswegen werden aus den Sensordaten geometrische Primitive abgeleitet und in einer Geometriekarte gespeichert. Dies führt zu einer hohen Datenkompression ohne den Verlust wichtiger Information. Diese Repräsentation lässt sich gut zu einem menschlichen Bediener übertragen und dort anzeigen.

Die Ableitung geometrischer Basisformen wie Ebenen oder Zylinder nimmt einen Großteil dieser Arbeit ein. Zunächst werden Filter für die von den 3-D-Kameras ausgegebenen Punktwolken betrachtet. Diese können Rauschen eliminieren oder die Dichte der Punktwolke beeinflussen. Im nächsten Schritt werden die Sensordaten in einem gemeinsamen Koordinatensystem registriert. Dies ist notwendig, da sowohl die Sensordaten als auch die Roboterposition mit Fehlern behaftet sind. Die registrierten Punktwolken werden im nächsten Schritt segmentiert. Hierbei kommt ein Verfahren zum Flächenwachstum auf Basis von Normalenvektoren zum Einsatz. Die entstehenden Segmente können anhand geometrischer Eigenschaften in Basisgeometrien klassifiziert werden. Diese werden abschließend in einer Geometriekarte vereint.

Für alle Teilschritte erfolgt eine Evaluierung gegenüber Verfahren aus dem Stand der Technik hinsichtlich Genauigkeit und Rechengeschwindigkeit. Abschließend erfolgt die Validierung der Funktionsweise anhand typischer Innenraumszenen und die Veranschaulichung durch ein Anwendungsbeispiel.

Abstract

Due to affordable sensors and actuators as well as improved algorithms, service robotics has evolved greatly during recent years. Especially perception has been identified as a key technology towards successful commercialisation. Capable 3-D cameras and efficient methods for sensor data processing enable robots to perceive and understand its environment as well as plan their actions based on these results. Despite this progress, a fully autonomous operation of mobile robots in environments made for humans is still far from being robust and reliable. A major reason for that is the variety of unforeseen exceptional cases a robot can be confronted with in a partially unknown and dynamic environment. An approach to semi-autonomous robot operation can be a mid-term solution and bring service robots one step forward. In semi-autonomous mode, the robot accomplishes tasks autonomously as long as possible. In unexpected situation where autonomous operation fails, a human operator can take over control, solve the problem and hand back to the robot.

Within this work 3-D environment perception for semi-autonomous mobile robots is addressed. The goal is to create an environment model from data of 3-D cameras that is usable both by the robot, mainly for navigation, manipulation and planning and by a human operator as support for tele-operation. This yields the ability for efficient data transfer over networks, easy visualization and intuitive interaction with the model. These specific requirements have to be met by the model representation. Furthermore, aggregation of the environment model from sensor data is a major topic of this thesis. Algorithms not only have to work with high accuracy but also need to be efficient with respect to computation speed in order to allow real-time applications.

Based on the requirements, an approach for 3-D environment perception is developed. The method creates a hybrid environment model, consisting of a point and a geometric map. The point representation can be used by the robot for obstacle avoidance in navigation and manipulation. However, because of high data volume and poor visualization capabilities of point representations it is not suitable for a human operator. Thus, geometric primitives are derived from sensor data and stored in a geometric map. This leads to a high data compression without loss of important information. The geometric representation can be easily transferred to and visualized at the operator side.

Computation of geometric primitives like planes or cylinders is a major part of this work. First, filters for point clouds coming from 3-D cameras are considered. These can be used for noise reduction or downsampling. An efficient general noise filter as well as a filter cascade for improved filtering is developed and evaluated on indoor scenes. In the next step, sensor data is registered to a common coordinate system during robot movement. This is mandatory as both the sensor data and the robot pose estimate are erroneous. Because state of the art does not meet the requirements of high computation speed as well as accuracy, a new approach based on edge features and fast data association is presented. Afterwards, the aligned point clouds are segmented using a normal-based region growing. The method proposed utilizes a graph structure to optimize the segments and prevent over-segmentation. The resulting segments are classified into geometric categories using 3-D point features. Finally, the segments are aggregated in a geometric map.

For all computation steps, in-depth evaluation of accuracy and computation speed is conducted. The usability of the system is validated on typical indoor scenes and shown by an example scenario from the field of semi-autonomous operation.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemstellung und Motivation	1
1.2	Zielsetzung und Vorgehensweise	2
2	Ausgangssituation	5
2.1	Definitionen und Begriffe	5
2.2	Anwendungsfelder für multifunktionale mobile Serviceroboter	5
2.2.1	Erfüllung von Haushaltsaufgaben	6
2.2.2	Anwendungen im gewerblichen Umfeld	7
2.2.3	Erkunden und Bergen bei Katastrophen	8
2.2.4	Erkundung und Probenentnahme bei Weltraummissionen	8
2.2.5	Bewertung der Anwendungsfelder und Abgrenzung der Arbeit	9
3	Analyse der Aufgabenstellung und Ableitung von Anforderungen	13
3.1	Anforderungen für Navigation und Manipulation	13
3.2	Anforderungen für Visualisierung und Interaktion	14
3.3	Anforderungen für die Planung von Aktionen	14
3.4	Weitere Anforderungen	15
4	Stand der Technik	17
4.1	3-D-Tiefenbildkameras	17
4.1.1	Kameras nach dem Prinzip der Phasenverschiebung	17
4.1.2	Kameras nach dem Prinzip der Triangulation	20
4.1.3	Bewertung der 3-D-Tiefenbildkameras	22
4.2	Umgebungserfassung in der Servicerobotik	23
4.2.1	3-D-Umgebungserfassung in der Teleoperation	23
4.2.2	3-D-Umgebungserfassung in der autonomen Robotik	25
4.3	Einzeltechnologien für 3-D-Umgebungserfassung	28
4.3.1	Vorverarbeitung	28
4.3.2	Nachbarschaftssuche in Punktwolken	30
4.3.3	Normalenvektoren	30
4.3.4	Rauschfilter	31
4.3.5	3-D-Merkmale	32
4.3.6	Registrierung	33
4.3.7	Segmentierung	34
4.3.8	Oberflächenrekonstruktion	36
4.4	Kartenrepräsentationen	36
4.5	Übertragung von 3-D-Daten über Netzwerke	38

4.6	Ansätze zum teil-autonomen Betrieb	39
4.7	Folgerungen aus dem Stand der Technik	40
5	Entwurf des Verfahrens zur 3-D-Umgebungserfassung	43
5.1	Anwendungsszenario und Zielrobotersystem	43
5.2	Konzeption des Verfahrens zur 3-D-Umgebungserfassung	43
5.2.1	Entwurf der Repräsentation für das Umgebungsmodell	44
5.2.2	Verarbeitungsschritte zur Erzeugung der Repräsentation	46
6	Entwurf und Umsetzung der Einzelkomponenten	49
6.1	Nachbarschaftssuche in Punktwolken	49
6.1.1	Bewertung	50
6.2	Punktnormalen	51
6.2.1	Vergleich der Verfahren zur Berechnung von Normalenvektoren	51
6.3	Rauschfilter	53
6.3.1	Amplituden- und Abrisskantenfilter für TOF-Kameras	53
6.3.2	Allgemeine Rauschfilter	57
6.3.3	Aufbau einer Filterkaskade	59
6.4	Registrierung	60
6.4.1	Fast Range Image Registration (FRIR)	61
6.4.2	Evaluierung	66
6.5	3-D-Punktmerkmale	71
6.5.1	Radius-based Surface Descriptor (RSD)	72
6.5.2	Principal Curvatures (PC)	73
6.5.3	Fast Point Feature Histograms (FPFH)	74
6.5.4	Klassifizierung von Oberflächenpunkten	76
6.5.5	Evaluierung	77
6.6	Segmentierung	80
6.6.1	Ebenensegmentierung mit RANSAC (ERS)	81
6.6.2	Graph-basiertes Region Growing und Punktklassifikation (GRG)	82
6.6.3	Evaluierung	87
6.7	Aufbau einer geometrische Karte	93
6.7.1	Verarbeitung von Ebenenpolygonen	93
6.7.2	Verarbeitung von Zylindern	94
6.7.3	Evaluierung	97
7	Verknüpfung der Einzelkomponenten und Validierung	103
7.1	Verknüpfung der Einzelkomponenten	103
7.2	Übertragung des Umgebungsmodells über ein Netzwerk	103
7.3	Überprüfung der Anforderungen	105
7.4	Validierung der Funktionalität	107
7.4.1	Validierung in der Simulation	107
7.4.2	Validierung auf Care-O-bot® 3	107
7.5	Anwendungsbeispiel „Hol- und Bringdienst“	109

8 Zusammenfassung und Ausblick	111
A Evaluierung von Klassifizierungsproblemen	113
B Datensatz für die Evaluierung	115
C Principal Component Analysis (PCA)	117
D Basisgeometrien	119
D.1 Ebene	119
D.2 Zylinder	120
Literaturverzeichnis	121

Abbildungsverzeichnis

1.1	Einfluss der Wahrnehmung auf andere Basistechnologien und teil-autonomer Betrieb.	2
2.1	Multifunktionale Serviceroboter für den Haushalt und für den gewerblichen Einsatz.	7
2.2	Beispiele für Katastrophen- und Weltraumroboter.	9
4.1	Phasenverschiebung zwischen ausgesandtem und empfangenem Signal.	18
4.2	Beispiel für von den TOF-Kameras erzeugte Daten.	19
4.3	Tiefenbildkameras mit Messung der Phasenverschiebung, von verschiedenen Herstellern.	20
4.4	Messprinzip der Musterprojektion.	21
4.5	Tiefenbildkameras mit Triangulation von verschiedenen Herstellern.	22
4.6	Beispiele für Visualisierung in der Teleoperation.	25
4.7	Anwendung des Voxelfilters.	28
4.8	Verschiedene Kartenrepräsentationen.	37
5.1	Zielsystem und Anwendungsszenario.	44
5.2	Grundlegendes Konzept für das Verfahren zur 3-D-Umgebungserfassung.	45
5.3	Detailliertes Konzept für das Verfahren zur 3-D-Umgebungserfassung (zu entwickelnde Komponenten sind eingefärbt).	47
6.1	Suchmaske.	50
6.2	Berechnung des Normalenvektors nach NNB.	51
6.3	Anwendung des Amplitudenfilters auf eine Innenraumszene.	54
6.4	Geometrische Beziehungen für die Abrisskantenselektion.	54
6.5	Anwendung des Abrisskantenfilters auf eine Innenraumszene.	55
6.6	TP und FP in Abhängigkeit von Filterparametern.	55
6.7	Verlauf der Rechenzeit für die einzelnen Filter in Abhängigkeit der Punktwolkengröße.	56
6.8	Anwendung des Specklefilters auf eine verrauschte Ebene.	57
6.9	Wahr Positive (TP) und falsch Positive (FP) in Abhängigkeit von Filterparametern.	59
6.10	Verlauf der Rechenzeit für den Specklefilter (Spk) und den statistischen Ausreißerfilter (SAF) in Abhängigkeit der Punktwolkengröße.	60
6.11	Filterkaskade.	60
6.12	Abschätzung der Rotationsdistanz.	62
6.13	Extraktion von Punkten mit hohem Informationsgehalt.	63
6.14	Schwache ($\mathbf{a}_1, \mathbf{b}_1$) und starke ($\mathbf{a}_2, \mathbf{b}_2$) Korrespondenzen.	65
6.15	Beispiele für evaluierte Szenen.	68

6.16	Transformationsfehler, gemittelt über alle registrierten Punktwolken. Standardabweichung als Fehlerindikator.	69
6.17	Absoluter Transformationsfehler bei Stillstand des Sensors, gemittelt über alle registrierten Punktwolken. Standardabweichung als Fehlerindikator.	70
6.18	Relative Auswirkung der Registrierung auf den Transformationsfehler (100% entspricht keiner Registrierung).	70
6.19	Visueller Vergleich der Verfahren.	71
6.20	Der Krümmungsradius r im Punkt \mathbf{p}_q wird durch $r = d/\alpha$ angenähert. Alle Nachbarpunkte in \mathcal{P}_k werden auf eine 5×2 Matrix zugeordnet. Ein lineare Regression führt zu den Radii r_{min} und r_{max}	73
6.21	Berechnung der Hauptkrümmungen über PCA	74
6.22	Darboux frame	76
6.23	Klassifizierungsbereiche abhängig von Radius bzw. Krümmung für RSD und PC.	77
6.24	Rechenzeit für die Merkmalsextraktoren	79
6.25	Verarbeitungskette für die Ebenenextraktion.	83
6.26	Beispiel einer Graphenstruktur für vier Segmente.	85
6.27	Verarbeitungskette für die Segmentierung und Klassifikation.	86
6.28	Notwendige Parameter zur Beschreibung eines Zylindersegments.	87
6.29	Visueller Vergleich von MPS, ERS und GRG. Klassifizierung von GRG.	92
6.30	Bestimmung von Merge-Kandidaten.	94
6.31	Merge-Sequenz für Ebenen.	95
6.32	Bestimmung von Merge-Kandidaten.	95
6.33	Projektion des Zylinders.	96
6.34	Merge-Sequenz für einen Zylinder.	97
6.35	Szenen für die Evaluierung.	97
6.36	Winkelfehler des Normalenvektors bei der Ebene (jeweils mit Mittelwert \bar{x} und Standardabweichung σ).	98
6.37	Fehler des Parameters d bei der Ebene (jeweils mit Mittelwert \bar{x} und Standardabweichung σ).	99
6.38	Messung der Genauigkeit für die Rückwand in der Küchenszene (jeweils mit Mittelwert \bar{x} und Standardabweichung σ).	100
6.39	Merge-Sequenz für einen Karteneintrag.	100
6.40	Abweichung des Ursprungs beim Zylinder (jeweils mit Mittelwert \bar{x} und Standardabweichung σ).	101
6.41	Abweichung des Radius beim Zylinder (jeweils mit Mittelwert \bar{x} und Standardabweichung σ).	101
6.42	Abweichung der Symmetrieachse beim Zylinder (jeweils mit Mittelwert \bar{x} und Standardabweichung σ).	102
6.43	Rechenzeit für das Hinzufügen von Elementen in der GK.	102
7.1	Endgültige Architektur für das System zur Umgebungserfassung.	104
7.2	PK in der Simulation zu unterschiedlichen Zeitpunkten.	107
7.3	GK in der Simulation zu unterschiedlichen Zeitpunkten.	107
7.4	Zum Vergleich verwendete echte Szenen.	108
7.5	Umgebungsmodell der Küchenszene.	108

7.6	Umgebungsmodell für die Tischszene.	109
7.7	Erster Teil des Szenarios.	109
7.8	Zweiter Teil des Szenarios.	110
A.1	Maße für die Evaluierung von Zweiklassenproblemen.	114
B.1	Vorgehensweise beim manuellen Klassifizieren.	115
B.2	Farbbilder der Szenen.	116
B.3	Manuell klassifizierte Szenen.	116
D.1	Definition eines Ebenenpolygons.	120
D.2	Definition eines Zylinders.	120

Tabellenverzeichnis

2.1	Vergleich der Anwendungsfelder	10
2.2	Sensoren zur Umgebungserfassung bei existierenden Robotern.	11
2.3	Eigenschaften von Sensoren zur Umgebungserfassung.	11
4.1	Reduktion der Punktdichte im Voxelfilter bei unterschiedlichen Auflösungen . . .	28
5.1	Vergleich der Kartenrepräsentationen	45
5.2	In einem Karteneintrag enthaltene Information.	46
6.1	Genauigkeit der Verfahren zur Normalenberechnung.	52
6.2	Rechenzeit der Verfahren zur Normalenberechnung.	52
6.3	Durchschnittliche Rechenzeit der Registrierungsverfahren im Vergleich (jeweils pro registriertem Tiefenbild).	71
6.4	Parameter für jedes untersuchte Verfahren für die beiden Distanzkategorien. . . .	78
6.5	Ergebnisse der Genauigkeitsuntersuchung nach Szenen.	79
6.6	Beispielszenen für die Merkmalsextraktion	80
6.7	Genauigkeit der Segmentierung bei GRG.	88
6.8	Konfusionsmatrizen für die Klassifizierung bei GRG.	89
6.9	Genauigkeit der Klassifizierung nach Szenen.	89
6.10	Kombinierte Genauigkeit von GRG.	90
6.11	Vergleich von ERS, MPS und GRG.	91
6.12	Rechenzeit der Verfahren zur Segmentierung.	91
7.1	Datenreduktion der Kartenrepräsentationen	104
7.2	Gesamtrechenzeit als Summe der Zeiten der Einzelkomponenten.	106
7.3	Überprüfung der Erfüllung der Anforderungen.	106
B.1	Farbcodes für die Oberflächentypen.	115

Abkürzungsverzeichnis

Akronyme und Abkürzungen

ANN	Approximate Nearest Neighbor
AR	Augmented Reality
DEM	Digital Elevation Map
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V.
FPFH	Fast Point Feature Histograms
GK	Geometriekarte
GRG	Graph-basiertes Region Growing
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
MLS	Moving Least Squares
NNB	Normalenberechnung durch Nachbarschaftsbeziehungen
PC	Principal Curvatures
PCA	Principal Component Analysis
PK	Punktekarte
RANSAC	Random Sample Consensus
RG	Region Growing
RSD	Radius-based Surface Descriptor
SLAM	Simultaneous Localization and Mapping
SURFEL	Surface Element
VR	Virtual Reality

Griechische Buchstaben

α	Winkel zwischen Normalen
λ	Eigenwert

Lateinische Buchstaben

\mathcal{P}	Punktwolke bzw. Tiefenbild
\mathcal{P}_k	Nachbarschaft um \mathbf{p}_q mit k Punkten
\mathbf{C}	Kovarianzmatrix
\mathbf{c}	Schwerpunkt
\mathbf{n}_i	Normalenvektor zu Punkt \mathbf{p}_i
\mathbf{o}	Ursprung
\mathbf{p}_i	Punkt i in \mathcal{P}
\mathbf{p}_q	Abfragepunkt in einer Punktwolke
\mathbf{q}	Nachbarpunkt
\mathbf{T}	Transformation
\mathbf{v}	Eigenvektor

i_h	Index für Schwellwerte
a, b, c, d	Ebenenparameter
c	Lichtgeschwindigkeit
D	Messdistanz
E bzw. e	Fehler
f	Brennweite
f_m	Modulationsfrequenz
n	Anzahl Punkte in einer Punktwolke
r	Radius
w	Gewichtungsfaktor

1 Einleitung

1.1 Problemstellung und Motivation

In den letzten Jahrzehnten haben die Automatisierung im Allgemeinen und die Servicerobotik im Speziellen große Fortschritte gemacht. Dies liegt zum einen an der Verfügbarkeit immer leistungsfähigerer Aktorik und Sensorik und zum anderen daran, dass Basistechnologien wie Navigation, Manipulation, Planung und Wahrnehmung zielstrebig vorangetrieben wurden (Srinivasa u. a., 2008; Jain u. Kemp, 2010). Daraus resultierend gibt es bereits seit einiger Zeit kommerzielle Produkte mit Spezialfunktionen wie Reinigungs-, Transport- oder neuerdings auch Telepräsenzroboter. Multifunktionale autonome Serviceroboter zur Erfüllung umfassender Aufgaben im Haushalt oder in der Industrie sind jedoch immer noch weitgehend Gegenstand von Forschungsarbeiten. Die Hauptgründe dafür sind die mangelnde Robustheit durch die Komplexität der Robotersysteme selbst und die Einsatzbedingungen in einer unstrukturierten, für den Menschen konzipierten Umgebung (Carff u. a., 2009).

Rein teleoperierte Roboter, wie sie beispielsweise nach der atomaren Katastrophe in Fukushima oder für Weltraummissionen eingesetzt wurden, können zwar durch einen menschlichen Bediener auch komplexe und unerwartete Situationen meistern. Allerdings ist häufig ein wochenlanges Training nötig, bevor die Bedienung des Roboters gelingt und selbst dann bleibt die Erfüllung von Aufgaben ein langwieriges und mühsames Unterfangen (Spectrum, 2011; Marsiske, 2011).

Eine mittelfristige Lösung, die sowohl die Robustheit autonomer als auch die Leistungsfähigkeit und Bedienbarkeit teleoperierter Robotersysteme deutlich erhöhen kann, ist der teil-autonome Roboterbetrieb (Abbildung 1.1a). Die Idee dahinter ist, dass der Roboter seine Aufgaben soweit wie möglich autonom erfüllt. Dabei reichen seine Fähigkeiten aus, um im Normalbetrieb zu operieren, solange keine unvorhergesehenen Situationen eintreten. Kann der Roboter aufgrund einer Ausnahme seine Aufgabe nicht erfüllen, ist es möglich, einen menschlichen Bediener zu Hilfe zu rufen, um das Problem zu lösen. Ein solches Vorgehen erfordert zum einen die Anbindung des Menschen an das Robotersystem und zum anderen speziell auf den teil-autonomen Betrieb ausgelegte Verfahren und Datenaufbereitung.

Ein weiterer wichtiger Aspekt bei der Forschung auf dem Gebiet der Servicerobotik ist die Tatsache, dass gerade die Wahrnehmung eine Schlüsselkomponente für die weitere Verbesserung autonomer Robotersysteme darstellt. Die im Jahr 2010 durchgeführte Effirob-Studie stellt fest, dass „die Wahrnehmung [...] die zentrale Basistechnologie in der gewerblichen Servicerobotik“ ist (Hägele u. a., 2011). Dies wird damit begründet, dass sie eine große Auswirkung auf andere Basistechnologien wie Navigation oder Manipulation hat (vgl. Abbildung 1.1b). Insbesondere die fehlerfreie und effiziente Erfassung der unstrukturierten Umgebung eines Roboters verspricht einen signifikanten Anstieg der Leistungsfähigkeit. Ein weiterer Faktor ist die Verfügbarkeit leistungsfähiger und kostengünstiger optischer Sensoren wie 3-D-Tiefenbildkameras. Diese liefern

überhaupt erst die zur Erstellung dreidimensionaler Umgebungsmodelle notwendigen Daten in ausreichender Qualität und Genauigkeit.

Ein Verfahren zur 3-D-Umgebungserfassung für teil-autonome Roboter kann folglich die Servicerobotik einen Schritt voranbringen, indem es die Möglichkeiten eröffnet, komplexe Aufgaben in unstrukturierten Umgebungen unter Einbeziehung des Menschen zu lösen.

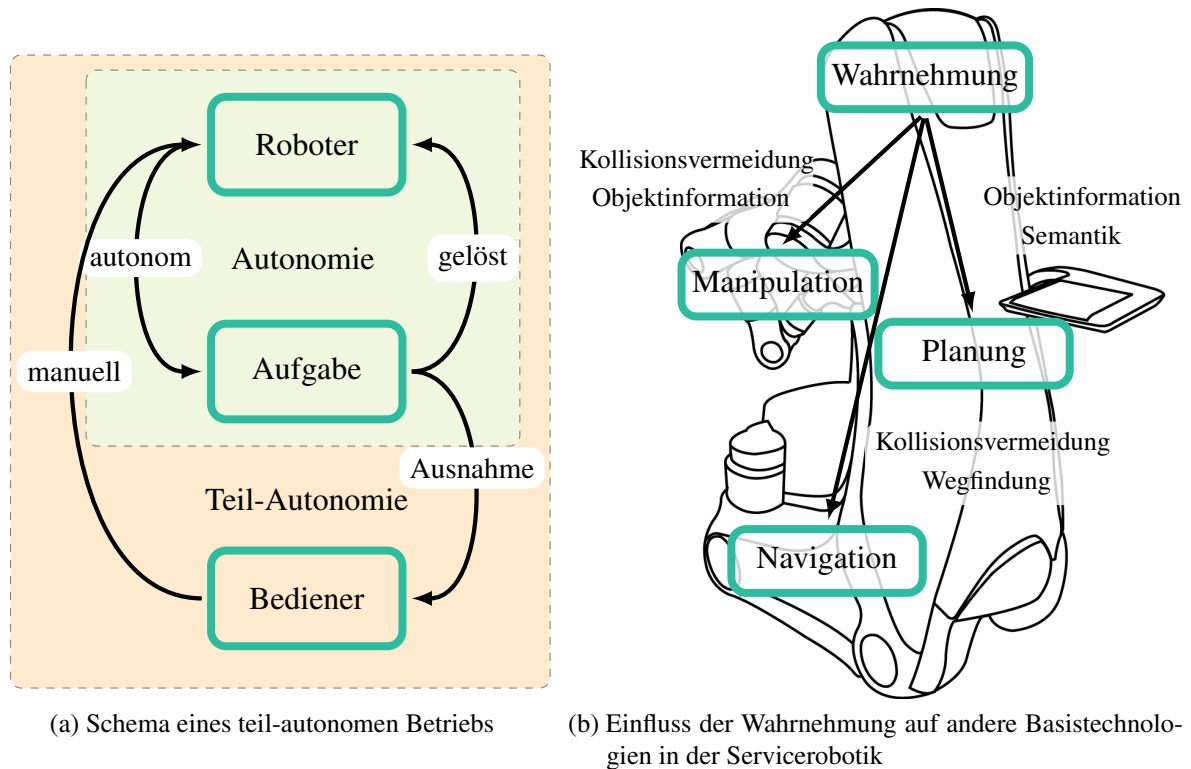


Abbildung 1.1: Einfluss der Wahrnehmung auf andere Basistechnologien und teil-autonomen Betrieb

1.2 Zielsetzung und Vorgehensweise

Ziel dieser Arbeit ist die Konzeption und Umsetzung eines Verfahrens zur 3-D-Umgebungserfassung für teil-autonome mobile Roboter. Das Verfahren soll

- eine dreidimensionale Repräsentation der Umgebung erzeugen, die vom Roboter zur Aufgabenerfüllung genutzt werden kann. Dazu müssen andere Basistechnologien wie Navigation, Manipulation oder (semantische) Planung das Umgebungsmodell interpretieren und darauf zugreifen können.
- Weiterhin soll das Verfahren und das erzeugte Umgebungsmodell einen menschlichen Bediener unterstützen können. Dafür ist es notwendig, die Daten entsprechend aufzubereiten, so dass eine schnelle Übertragung, einfache Visualisierung und intuitive Interaktion möglich sind.

- Durch Unterstützung eines teil-autonomen Betriebs soll dieses Verfahren dabei helfen, Problemsituationen zu meistern, die mit einem autonomen oder einem teleoperierten Roboter auf dem Stand der Technik nicht lösbar wären.
- Dieses Verfahren soll auf gegenwärtig verfügbaren, multifunktionalen, mobilen Servicerobotern einsetzbar sein, muss also vor allem auf deren Anwendungsgebiete und Sensorik ausgelegt sein.

Für die Erarbeitung einer Lösung soll zunächst die Ausgangssituation betrachtet werden. Im Zuge dessen werden typische Anwendungsfelder für Serviceroboter skizziert und untersucht. Dies ist in Hinblick auf die benötigte Funktionalität und Randbedingungen des zu entwickelnden Verfahrens wichtig. Weiterhin wird erörtert, welche Serviceroboterplattformen derzeit verfügbar sind, welche Aktoren und Sensoren sie besitzen und in welchen Modi sie betrieben werden. Darauf aufbauend werden die Randbedingungen des zu entwickelnden Verfahrens abgegrenzt.

Nachfolgend wird die Aufgabenstellung detailliert analysiert und es werden konkrete Anforderungen an das Verfahren abgeleitet, damit die Zielsetzung eingehalten werden kann. Im nächsten Schritt soll dann der Stand der Technik beleuchtet werden. Hier liegt der Fokus auf den Bereichen Umgebungserfassung und -darstellung, sowohl in der Teleoperation als auch bei autonomen Robotern. Ziel ist dabei, die Leistungsfähigkeit bisher entwickelter Verfahren zu bewerten und einen Überblick über aktuell eingesetzte Algorithmen zu vermitteln. Weiterhin soll geprüft werden, inwieweit die Anforderungen bereits durch den Stand der Technik erfüllt sind und wo sich noch Lücken befinden, die durch das in dieser Arbeit entwickelte Verfahren geschlossen werden müssen.

Auf Basis der Anforderungen und des Stands der Technik kann das Verfahren zur 3-D-Umgebungserfassung entworfen werden. Der Kernpunkt ist dabei eine für alle Anwendungen geeignete Kartenrepräsentation für das Umgebungsmodell. Weiterhin werden wichtige Einzelkomponenten identifiziert und deren Entwicklungsaufwand abgeschätzt.

Aus dem Konzept heraus erfolgt die Betrachtung der benötigten Einzelkomponenten. Hier gilt es, bereits bestehende Verfahren gemäß ihrer Einsetzbarkeit zu untersuchen, ggf. neue Methoden zu erarbeiten und diese hinsichtlich der Anforderungen zu evaluieren. Danach werden die Einzelkomponenten verknüpft und die Leistungsfähigkeit des Verfahrens analysiert. Die Anwendbarkeit des neuen Verfahrens in einem teil-autonomen Roboterbetrieb wird schließlich anhand eines Anwendungsbeispiels verdeutlicht.

2 Ausgangssituation

Dieses Kapitel beschreibt die Ausgangssituation für den Betrieb mobiler Serviceroboter. Dabei wird zunächst die Definition und Erklärung wichtiger Begriffe für diese Arbeit vorgenommen. Danach gibt dieses Kapitel einen Überblick über typische Anwendungsfelder für mobile Serviceroboter und die eingesetzte Sensorik.

2.1 Definitionen und Begriffe

Serviceroboter: „Ein Serviceroboter ist eine frei programmierbare Bewegungseinrichtung, die teil- oder vollautomatisch Dienstleistungen verrichtet. Dienstleistungen sind dabei Tätigkeiten, die nicht der direkten industriellen Erzeugung von Sachgütern, sondern der Verrichtung von Leistungen für Menschen und Einrichtungen dienen.“ (Schraft u. Volz, 1996) Als multifunktionaler Serviceroboter wird in dieser Arbeit ein Roboter bezeichnet, der in der Lage ist, mehrere unterschiedliche Aufgaben zu erfüllen und nicht auf eine spezielle Anwendung (z. B. Staubsaugen) beschränkt ist.

Teil-autonomer Betrieb: Wird ein Roboter teil-autonom betrieben, so erfüllt er Aufgaben selbstständig, soweit es ihm aufgrund seiner Sensoren, Aktoren und Software möglich ist. Sollte er an der Ausführung scheitern, so kann ein menschlicher Bediener die Kontrolle über den Roboter übernehmen und die Aufgabe oder einen Teilschritt soweit manuell lösen, bis der Roboter wieder autonom agieren kann. Der durchschnittliche Autonomiegrad ist dabei nicht festgelegt.

Umgebungserfassung: Dieser Begriff beschreibt die Wahrnehmung der Umgebung eines Roboters durch seine Sensoren und die Verarbeitung und Aufbereitung der Daten derart, dass sie zur Erfüllung von Aufgaben genutzt werden können.

Punktwolke: Eine Punktwolke \mathcal{P} ist eine Menge diskreter Punkte in einem dreidimensionalen Koordinatensystem. Die Punkte sind typischerweise durch x -, y - und z -Koordinaten bestimmt und dienen vorrangig dazu, die Oberfläche eines Objekts im Raum zu beschreiben. Sie können auch zusätzliche Informationen wie Farbwerte, Labels oder Punktnormalen besitzen.

Tiefenbild: Ein Tiefenbild bezeichnet eine geordnete Punktwolke, in der die einzelnen Punkte in einer Pixelstruktur angeordnet sind.

2.2 Anwendungsfelder für multifunktionale mobile Serviceroboter

Der folgende Abschnitt gibt einen Überblick über typische Anwendungsfelder multifunktionaler mobiler Serviceroboter. Dabei wird auf häufige Tätigkeiten eingegangen, die Verwendungsmög-

lichkeiten eines Umgebungsmodells aufgezeigt und der Grad der Autonomie in den einzelnen Bereichen beleuchtet. Speziallösungen wie reine Transport- oder Reinigungsroboter, die ausschließlich eine gut funktionierende Navigation benötigen und keine Manipulation von Objekten durchführen, befinden sich außerhalb des Fokus dieser Arbeit, da diese Aufgaben bereits weitgehend autonom gelöst werden. Für jeden Anwendungsbereich werden existierende Roboterplattformen hinsichtlich ihrer Sensorik für die Umgebungserfassung analysiert und daraus Folgerungen für das in dieser Arbeit zu entwickelnde Verfahren abgeleitet.

2.2.1 Erfüllung von Haushaltsaufgaben

Mobile Serviceroboter können für viele alltägliche Aufgaben im Haushalt eingesetzt werden, beispielsweise für Hol- und Bringdienste, zum Kochen, für Reinigungstätigkeiten oder zur Unterhaltung (Mast u. a., 2012). Gemeinsamkeiten bei den Aufgaben sind, dass diese zunächst geplant werden müssen. Weiterhin müssen für eine erfolgreiche Durchführung meistens Objekte identifiziert und nachfolgend manipuliert werden. Schließlich müssen die eingesetzten Roboter in der Lage sein, sicher zwischen Positionen in einer Wohnung zu navigieren.

Den meisten Haushaltsumgebungen ist gemeinsam, dass es sich um unstrukturierte, für den Menschen konzipierte Innenräume handelt, in denen der Roboter parallel zu oder sogar gemeinsam mit Menschen operiert. Der Großteil der Umgebung (Räume, Durchgänge, große Möbelstücke) ist zwar statisch und kann daher als bekannt angesehen werden. Durch die Anwesenheit von Menschen ergibt sich aber eine dynamische Komponente, die sich meist durch Ortsveränderungen von kleineren Objekten oder Umkonfigurationen wie geöffnete oder geschlossene Türen zeigt. Dadurch ergibt sich bereits für scheinbar einfache Tätigkeiten wie Hol- und Bringdienste eine hohe Komplexität, bedingt durch die Vielzahl an möglichen Ausnahmesituationen. Ein Umgebungsmodell kann helfen, Veränderungen zum bekannten Zustand zu erkennen, Objekte entsprechend zu identifizieren und angemessen darauf zu reagieren. In unbekanntem Räumen kann durch die Umgebungserfassung eine dreidimensionale Repräsentation erzeugt werden.

Die meisten Tätigkeiten im Haushalt sind Routineaufgaben und lassen sich zumindest theoretisch leicht autonom lösen. Deswegen liegt der Forschungsschwerpunkt meist auf autonomen Robotersystemen. Ein weiterer Grund dafür ist, dass häufig einzelne Basistechnologien im Rahmen eines Szenarios vorangetrieben und getestet werden, jedoch nicht das Gesamtsystem. Vor diesem Hintergrund wird versucht, die Robustheit und damit die Autonomie für eine Komponente zu maximieren.

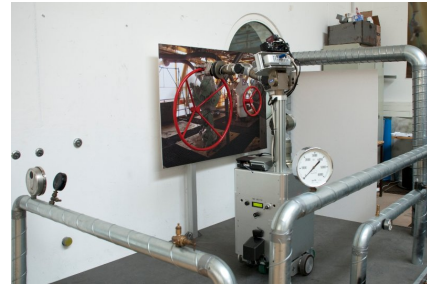
Typische multifunktionale Serviceroboter für Haushaltsaufgaben sind der vom Fraunhofer IPA im Jahr 2009 entwickelte Care-O-bot[®] 3 (Graf u. a., 2009) (siehe Abbildung 2.1a), der PR2 (Abbildung 2.1b) von Willow Garage (Willow Garage, 2012), der von der Universität Bonn entwickelte NimRo (University of Bonn, 2012) oder Rollin' Justin vom DLR (DLR - Institute of Robotics and Mechatronics, 2012). Diese Roboter haben gemeinsam, dass sie etwa so hoch wie ein Mensch sind und neben Navigationsfähigkeiten auch einen oder mehrere Arme zur Manipulation von Objekten besitzen. Die Anzahl verwendeter Sensoren ist typischerweise hoch, fast immer besitzen die Roboter einen beweglichen Sensorkopf.



(a) Care-O-bot® 3 greift eine Flasche (Fraunhofer IPA, 2012)



(b) PR2 liest ein Buch (Straszheim, 2011)



(c) ImRoNet-Demonstrator manipuliert ein Stellrad

Abbildung 2.1: Multifunktionale Serviceroboter für den Haushalt und für den gewerblichen Einsatz

2.2.2 Anwendungen im gewerblichen Umfeld

Bei Anwendungen im gewerblichen Umfeld sind vor allem die Erfüllung von Transportaufgaben, Bodenreinigung und Tätigkeiten im Bereich von Inspektion und Wartung zu nennen, wobei besonders dieser letzte Bereich Gegenstand vieler Forschungsprojekte ist (Connette u. a., 2012). Insgesamt ist eine höhere Spezialisierung auf Einzelaufgaben zu beobachten als bei den Anwendungen im Haushalt. Dies liegt daran, dass gewerblich eingesetzte Robotersysteme zum einen höhere Investitionskosten verursachen dürfen und zum anderen der Roboter häufig bereits durch eine Spezialaufgabe gut ausgelastet ist. Die erfolgreiche Durchführung der meisten Aufgaben erfordert aber wie bei den Haushaltstätigkeiten ein Zusammenspiel von Planung, Navigation, Manipulation und Wahrnehmung der Umgebung.

Die Randbedingungen des Operationsumfelds sind ähnlich wie bei Haushalten: Es handelt sich meist um ursprünglich für den Menschen konzipierte, unstrukturierte und teilweise dynamische Innenraumumgebungen. Daraus ergibt sich das gleiche Anwendungsfeld für ein Umgebungsmodell, allerdings unter anderen Randbedingungen. So ist der Ordnungsgrad deutlich höher als in Haushaltsumgebungen. Dies liegt zum einen an der Anwesenheit anderer Maschinen und Fahrzeuge, für die die Umgebung bereits angepasst ist, und zum anderen daran, dass im gewerblichen Umfeld ein Mindestmaß an Ordnung für einen geregelten Betriebsablauf erforderlich ist. Weitere Unterschiede zu Haushalten liegen darin, dass es sich meist um deutlich weitläufigere Umgebungen handelt, die zudem noch schmutzig oder gefährlich sein können.

Um eine möglichst große Kosteneinsparung gegenüber herkömmlichen Lösungen zu erzielen, ist im gewerblichen Umfeld ein hoher Grad an Autonomie Voraussetzung für einen erfolgreichen Robotereinsatz. In einem teil-autonomen Betrieb können aber bereits Kosteneinsparungen erzielt werden, wenn beispielsweise dadurch die Fernwartung von Anlagen ermöglicht wird und somit der Wartungsspezialist nicht mehr vor Ort sein muss. Dies macht den teil-autonomen Betriebsmodus interessant für gewerbliche Anwendungen.

Roboter für gewerbliche Aufgaben haben funktional ähnliche Anforderungen wie Roboter für den Haushalt. Direkte Benutzerinteraktion spielt jedoch gegenüber Aspekten wie Traglast, La-

der Raum und Robustheit in Bezug auf Schmutz, Gase oder Ähnlichem eine untergeordnete Rolle. Typische Vertreter sind der im von der Landesstiftung Baden-Württemberg getragenen Projekt ATLAS als Demonstrator eingesetzte *rob@work 3* (Connette u. a., 2012). Der Roboter kann zur Inspektion und Wartung von Industrieanlagen, aber auch als Handhabungsassistent oder für Transportaufgaben eingesetzt werden. Ähnlichen Zwecken, beispielsweise der Manipulation von Stellrädern, dient der in (Arbeiter u. a., 2010) vorgestellte *ImRoNet*-Demonstrator (Abbildung 2.1c). *Sensabot* (Showcase, 2012) hingegen dient mangels Manipulator ausschließlich zur Inspektion in schwierigen und gefährlichen Umgebungen.

2.2.3 Erkunden und Bergen bei Katastrophen

Im Umfeld von Katastrophen wie Reaktorunfällen, Erdbeben oder Terroranschlägen können Roboter zu unterschiedlichen Zwecken eingesetzt werden (Sun u. a., 2011). Die Erkundung stark beschädigter Gebäude ist beispielsweise eine unschätzbare Hilfe für Rettungsteams. Die Entfernung von Schutt oder sonstigen Hindernissen durch Roboter ermöglicht den Zugang zu verletzten Personen im Gebiet und dadurch deren Bergung. Die Lösung dieser Aufgaben setzt eine Bewegungsfähigkeit in unwegsamem Gelände wie Schutt, extreme Wendigkeit und Flexibilität in Bezug auf die Umgebungsbedingungen voraus.

Die Bedingungen an Katastrophenorten sind häufig extrem. Durch Beschädigung und Zerstörung von Strukturen ergibt sich zum einen eine völlig unbekannte Umgebung und zum anderen unwegsames Gelände, welches für den Menschen unpassierbar ist. Weiterhin sind sowohl Innen- als auch Außenumgebungen als Operationsgebiet denkbar. Dies führt dazu, dass die autonome Wahrnehmung der Umgebung durch die Vielzahl unbekannter und ungeordneter Objekte eine große Herausforderung ist. Ein Umgebungsmodell kann eingesetzt werden, um räumliches Wissen über das Einsatzgebiet zu sammeln, abzubilden und zu visualisieren. Dieses kann dann zur Pfadplanung und Hinderniserkennung in der Navigation oder für die Manipulation von Objekten verwendet werden.

Durch die Komplexität der Einsatzumgebung und der Vielzahl unbekannter Situationen existieren gegenwärtig hauptsächlich teleoperierte Roboter für den Einsatz bei Katastrophen. Zwar wird dadurch überhaupt erst ein Einsatz möglich, gleichzeitig ergibt sich jedoch das Problem der sicheren Kommunikation zwischen Roboter und Bediener. Insbesondere bei Innenräumen sind nur Verbindungen mit hoher Latenz und geringer Bandbreite möglich. Dadurch können Sensordaten und Steuerkommandos nur eingeschränkt übertragen werden. Die ohnehin komplexe und mühsame Bedienung dieser Roboter wird dadurch noch erschwert (Spectrum, 2011).

Zur Inspektion in Fukushima eingesetzt wurden die beiden von *iRobot* produzierten Roboter *Packbot* (Abbildung 2.2a) und *Warrior* (iRobot, 2012) und der japanische *Quince* (Abbildung 2.2b) (Nagatani u. a., 2011). Weiterhin existiert der vom gleichen Forscherteam bereits früher für den gleichen Einsatzzweck entwickelte Roboter *Kenaf* (Nagatani u. a., 2010).

2.2.4 Erkundung und Probenentnahme bei Weltraummissionen

Bei Weltraummissionen werden Roboter vor allem zur Erkundung und zum Sammeln von Gesteins- oder Bodenproben auf anderen Planeten eingesetzt. Sie müssen dabei häufig größere Ent-

fernungen in unbekanntem Gelände zurücklegen und in der Lage sein, für die Probenentnahme geeignete Stellen zu identifizieren (Showstack, 2012).

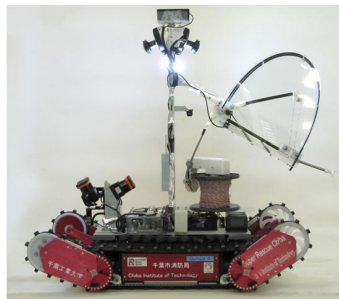
Ähnlich wie bei Katastrophenszenarien handelt es sich bei den Einsatzgebieten für Weltraumroboter um völlig unbekannte Außenumgebungen, die stark unstrukturiert und unwegsam sind. Folglich kann ein Umgebungsmodell grundlegende räumliche Daten über das Umfeld des Roboters liefern. Diese werden hauptsächlich für die Navigation eingesetzt.

Aufgrund der großen Entfernung und damit einhergehenden hohen Latenz bei der Signalübertragung ist eine reine Teleoperation nicht möglich. Vielmehr müssen Roboter für Weltraummissionen meistens völlig autonom agieren, ein teil-autonomer Betrieb ist nur eingeschränkt möglich. Die dafür erforderliche Robustheit wird erkaufte durch einen extrem hohen Entwicklungsaufwand und enorme Kosten. Dies ist unwirtschaftlich und daher kaum auf andere Anwendungsfelder übertragbar.

Fahrzeuge wie der Mars Exploration Rover *Opportunity* (Bell, 2003) und der in Abbildung 2.2c zu sehende *Curiosity* (NASA - Mars Science Laboratory, 2012) sind Beispiele für Roboter bei planetaren Weltraummissionen.



(a) Packbot in Fukushima (TEPCO, 2013)



(b) Erkundungsroboter Quince (FareEastGizmos, 2011)



(c) Mars Rover Curiosity (NASA, 2011)

Abbildung 2.2: Beispiele für Katastrophen- und Weltraumroboter.

2.2.5 Bewertung der Anwendungsfelder und Abgrenzung der Arbeit

Die Gegenüberstellung der Anwendungsfelder ist in Tabelle 2.1 zu sehen. Die Bewertung erfolgt anhand der Randbedingungen der Umgebung und der Eignung für den teil-autonomen Betrieb. Die Bedingungen für Haushalt und gewerbliche Anwendungen sind sehr ähnlich, sie unterscheiden sich lediglich in der Größe und im Ordnungsgrad. Katastrophenorte zeichnen sich hingegen durch einen sehr geringen Ordnungs- und Bekanntheitsgrad aus ebenso wie Weltraumanwendungen. Weiterhin sind bei diesen Gebieten Außenbereiche dominant, während die ersten beiden Anwendungsgebiete überwiegend in Innenraumumgebungen angesiedelt sind.

Ein Verfahren zur Umgebungserfassung kann bei allen Anwendungen dafür eingesetzt werden, Informationen über die Umgebung zu sammeln und für Roboter und Menschen lesbar abzubilden. Dadurch wird der Bekanntheitsgrad der Umgebung erhöht. Ein Umgebungsmodell kann so-

Anwendung	Innen- räume	Bekanntheits- grad	Ordnungs- grad	Größe	Teil- autonomie
Haushalt	✓	hoch	mittel	klein	✓
Gewerbe	✓	hoch	hoch	mittel	✓
Katastrophe	(✓)	niedrig	niedrig	groß	✓
Weltraum	✗	niedrig	niedrig	groß	(✓)
	✗ nicht erfüllt	(✓) teilweise erfüllt	✓ erfüllt		

Tabelle 2.1: Vergleich der Anwendungsfelder

wohl beim Roboter die autonome Navigation, Manipulation oder Planung als auch einen menschlichen Bediener bei der Erfassung der Situation und Steuerung des Roboters unterstützen.

Generell ist für den teil-autonomen Betrieb die Übertragung von Umgebungsdaten über ein Netzwerk erforderlich. Dieses unterliegt je nach Anwendungsfeld unterschiedlich starken Einschränkungen in Bezug auf Bandbreite und Latenz. Relativ geringe Einschränkungen im Haushalt und im gewerblichen Umfeld stehen einer hohen Latenz und geringer Bandbreite in Katastrophengebieten gegenüber. Bei Weltraumanwendungen können Verzögerungen von mehreren Minuten auftreten, weswegen diese nur eine eingeschränkte Eignung für den teil-autonomen Betrieb aufweisen.

Viele der vorgestellten Roboter besitzen ähnliche Komponenten und Sensorkombinationen. Tabelle 2.2 ist zu entnehmen, dass alle Roboter mindestens mit Farbkameras ausgestattet sind. Weiterhin häufig verwendet werden Laserscanner. Bei einigen Robotern sind sie unbeweglich und werden vornehmlich zur 2-D-Navigation eingesetzt. Für eine dreidimensionale Wahrnehmung dominieren drei Ansätze: Stereokameras, schwenkbare Laserscanner und 3-D-Tiefenbildkameras. Alle drei sind in der Lage, Sensordaten in Form von Punktwolken auszugeben. Prinzipiell können also all diese Sensortypen zur Erstellung eines Umgebungsmodells verwendet werden, jedoch mit variierender Auflösung, Dichte und Genauigkeit bei unterschiedlich guter Eignung für bestimmte Umgebungsbedingungen. Zu erkennen ist, dass Roboter für Außenanwendungen vorzugsweise Farb- oder Stereokamerasysteme, teilweise auch Laserscanner zur Umgebungserfassung einsetzen. 3-D-Tiefenbildkameras kommen nur auf den für Innenräume konzipierten Robotersystemen vor.

Die in Tabelle 2.3 aufgelisteten Eigenschaften zeigen, dass bislang kein optimaler Sensor existiert. Schwenkbare Laserscanner zeichnen sich durch eine hohe Genauigkeit, großes Sichtfeld und große Reichweite aus, haben jedoch durch die mechanische Bewegung eine komplexe Handhabung und eine geringe Bildwiederholrate. Zudem muss der Roboter stillstehen, um eine Punktwolke aufnehmen zu können, was für das zu entwickelnde Verfahren nicht akzeptabel ist (Nüchter, 2009).

Stereokameras können ebenfalls sehr genaue Tiefendaten bei hoher Auflösung erzeugen, allerdings nur in einem durch den Basisabstand bestimmten Entfernungsbereich. Der Hauptnachteil ist jedoch, dass bei untexturierten Oberflächen keine dichten Punktwolken erzeugt werden kön-

Roboter	Farb- kamera	Stereo- system	3-D-Tiefen- bildkamera	Laser- scanner statisch	Laser- scanner schwenkbar
Care-O-bot® 3	✓	✓	✓	✓	✗
PR2	✓	✓	✓	✓	✓
NimbRo	✓	✓	✓	✓	✓
Rollin' Justin	✓	✓	✓	✗	✗
ImRoNet	✓	✓	✓	✓	✗
rob@work3	✓	✗	✓	✓	✗
Sensabot	✓	✗	✗	✓	✗
Packbot	✓	✗	✗	✗	✗
Warrior	✓	✗	✗	✗	✗
Quince	✓	✗	✗	✓	✗
Kenaf	✓	✗	✗	✗	✓
Opportunity	✓	✓	✗	✗	✗
Curiosity	✓	✓	✗	✗	✗

✗ nicht vorhanden ✓ vorhanden

Tabelle 2.2: Sensoren zur Umgebungserfassung bei existierenden Robotern

	schwenkbarer Laserscanner	Stereokameras	3-D-Tiefen- bildkamera
Eignung für Außenbereich	✓	✓	(✓)
Vollständigkeit der Daten	✓	✗	✓
hohe Bildwiederholrate	✗	(✓)	✓
hohe Genauigkeit (Tiefe)	✓	✓	✓
hohe Auflösung	(✓)	✓	✓
hohe Reichweite	✓	(✓)	✗
großes Sichtfeld	✓	✗	✗
einfache Handhabung	✗	(✓)	✓

✗ nicht erfüllt (✓) teilweise erfüllt ✓ erfüllt

Tabelle 2.3: Eigenschaften von Sensoren zur Umgebungserfassung

nen. Dadurch ist es nicht möglich, ein vollständiges Umgebungsmodell zu erstellen (Foix u. a., 2011).

3-D-Tiefenbildkameras hingegen sind bei keiner der Eigenschaften herausragend, besitzen jedoch auch keine Nachteile, die ein Ausschlusskriterium darstellen. Weiterhin bieten sie als jüngste Gruppe unter den 3-D-Sensoren das größte Potenzial für eine Steigerung der Leistungsfähigkeit in der Zukunft. Aus diesen Gründen werden in dieser Arbeit Daten von 3-D-Tiefenbildkameras als Ausgangspunkt für die Entwicklung des Verfahrens zugrunde gelegt. Eine Einsetzbarkeit der anderen Sensortypen ist damit nicht ausgeschlossen, wird aber nicht weiter untersucht.

Wegen der Festlegung auf 3-D-Tiefenbildkameras als primäre Sensorik für die 3-D-Umgebungserfassung liegt der Fokus dieser Arbeit auf der Entwicklung eines Verfahrens vornehmlich für Innenraumumgebungen und damit für Anwendungen in Haushalt oder im gewerblichen Umfeld. Dies wird noch gestützt durch den extrem niedrigen Ordnungsgrad in Außengebieten, insbesondere in Katastrophengebieten und der damit einhergehenden Erschwernis einer dreidimensionalen Abbildung der Umgebung. Weiterhin werden auf nahezu allen Robotern für Haushalt und Gewerbe Laserscanner für die 2-D-Navigation eingesetzt. Daher kann davon ausgegangen werden, dass eine für den Aufbau eines Umgebungsmodells hilfreiche Positionsschätzung des Roboters zur Verfügung steht.

3 Analyse der Aufgabenstellung und Ableitung von Anforderungen

In diesem Kapitel wird die Aufgabenstellung der Arbeit detailliert dargestellt, gleichzeitig werden konkrete Anforderungen an das zu entwickelnde Verfahren zur Umgebungserfassung abgeleitet. Aus der Ausgangssituation lässt sich schließen, dass das zu entwickelnde Verfahren in der Lage sein muss, nur teilweise bekannte oder unbekannte, unstrukturierte und teilweise dynamische Umgebungen zu erfassen und zu kartieren. Weiterhin soll das Kartierungsverfahren den teil-autonomen Roboterbetrieb in allen Bereichen unterstützen, um einen erfolgreichen Einsatz in den Anwendungsfeldern Haushalt und Gewerbe zu gewährleisten. Dadurch ergibt sich die Forderung der Eignung zur

- Navigation und Manipulation: Die generierte Karte soll im autonomen Roboterbetrieb für die Kollisionsvermeidung in Navigation und Manipulation verwendet werden,
- Visualisierung und Interaktion: Das Umgebungsmodell soll einfach und für den Menschen lesbar und visualisierbar sein und Möglichkeiten zur Interaktion bieten und
- Planung von Aktionen: Das Umgebungsmodell soll die Grundlage für die semantische und abstrakte Planung von Handlungen durch den Roboter bilden.

Aus diesen drei funktionalen Hauptforderungen lassen sich im Folgenden technische Anforderungen ableiten. Können diese erfüllt werden, gelten auch die Hauptforderungen als sichergestellt.

3.1 Anforderungen für Navigation und Manipulation

Algorithmen für die kollisionsfreie Bahnplanung in Navigation und Manipulation benötigen meist ein spezielles Format, um die Eingangsdaten verarbeiten zu können. Weiterhin geschieht die Bahnplanung oft während der Roboterbewegung. Insbesondere bei der Manipulation wird die Umgebung des Roboters zumeist verändert. Dies führt zu folgenden Anforderungen:

- A1** Die Umgebungskarte muss als Punkterepräsentation vorliegen, da dies das Eingangsformat für gängige Algorithmen zur Kollisionsvermeidung ist (Kunz u. a., 2010).
- A2** Die Genauigkeit des Umgebungsmodells muss ausreichend sein, um eine fehlerfreie Bewegung des Roboters zu ermöglichen. Bei Manipulationsaufgaben muss beispielsweise die Genauigkeit hoch genug sein, um sicher Hindernisse zu vermeiden, aber gleichzeitig das gewünschte Objekt manipulieren zu können. In einer Entfernung von maximal 2 m vom Roboter muss das Umgebungsmodell folglich eine Genauigkeit von 1 cm aufweisen (Succan u. a., 2010).
- A3** Da die Planung häufig während der Roboterbewegung stattfindet, muss die Aktualisierungsgeschwindigkeit für die Punktekarte hoch genug sein, um Kollisionen auch bei beweglichen Hindernissen oder Menschen in der Umgebung sicher verhindern zu können. In der

Praxis bedeutet dies, dass die Aktualisierung der Punkterepräsentation optimalerweise mit voller Sensorbildfrequenz möglich ist (Sucan u. a., 2010).

- A4** Es müssen Möglichkeiten bestehen, Veränderungen in der Umgebung in das Modell einfließen zu lassen oder das Umgebungsmodell im Falle von Veränderungen neu zu generieren.

3.2 Anforderungen für Visualisierung und Interaktion

Für den teil-autonomen Betrieb ist die Anbindung eines menschlichen Bedieners an das Robotersystem unerlässlich. Für das Umgebungsmodell bedeutet dies, dass es zur Bedienerseite übertragen und dort visualisiert werden muss. Auch die Interaktion mit dem Modell muss möglich sein.

- A5** Das Umgebungsmodell muss einfach zu visualisieren sein, d. h. es muss sich ohne aufwändige Berechnungen oder Transformationen anzeigen lassen. Zusätzlich muss ein menschlicher Bediener in der Lage sein, die visualisierten Daten einfach zu interpretieren. Weiterhin muss es möglich sein, die Aufmerksamkeit des Bedieners auf bestimmte Regionen oder Objekte zu lenken. Dies bedeutet, dass die Elemente des Modells als einzelne Instanzen vorliegen müssen.
- A6** Die Aktualisierungsgeschwindigkeit des gesamten Modells muss so hoch sein, dass ein menschlicher Bediener insbesondere bei Navigationsaufgaben noch auf unvorhergesehene Situationen reagieren kann. Wie lang die zur Verfügung stehende Zeit ist, hängt von der Maximalgeschwindigkeit des Roboters ab. Für eine maximale Robotergeschwindigkeit von 0,7 m/s reicht beispielsweise ein Aktualisierungsintervall von 0,5 s aus, den Roboter unter Berücksichtigung des Bremswegs innerhalb von etwa 0,5 m zum Stillstand zu bringen. Dies ist für die meisten Anwendungen ausreichend.
- A7** Das Datenvolumen muss klein genug sein, um auch bei geringer Bandbreite von weniger als 1 Mbit mit der in A6 genannten Rate von 2 Hz über ein Netzwerk übertragen werden zu können.
- A8** Das Umgebungsmodell muss Möglichkeiten zur Interaktion bieten, sei es durch Einblenden von Zusatzinformationen zu Objekten oder Modifikation des Umgebungsmodells.

3.3 Anforderungen für die Planung von Aktionen

Handlungsplanung verwendet häufig semantische Informationen als Eingangsdaten (Pronobis, 2011). So ist es z. B. möglich, dem Roboter abstrakte Anweisungen wie „Hole Objekt A vom Tisch in der Küche“ zu geben. Diese werden dann mit Hilfe des Umgebungsmodells in metrische Informationen übersetzt. Die Planung selbst steht nicht im Fokus dieser Arbeit, auch nicht das Erstellen von semantischen Karten oder Ontologien. Durch eine geometrische Beschreibung der Umgebung kann jedoch eine Basis für diese Anwendungen geschaffen werden. Deswegen gelten hier die Forderungen:

- A9** Das Umgebungsmodell muss geometrische Informationen enthalten,

- A10** die die tatsächliche Umgebung möglichst genau und vollständig wiedergeben. Dabei ist entscheidend, dass die geometrische Information vom Roboter korrekt klassifiziert werden kann. Die Einhaltung dieser Forderung kann durch Evaluierung gegenüber bisher eingesetzten Verfahren geprüft werden.
- A11** Die geometrischen Informationen müssen dazu geeignet sein, semantische Informationen abzuleiten. Sie müssen also bereits höherwertige Information wie Modellparameter oder eine Aussage über die geometrische Grundform enthalten.

3.4 Weitere Anforderungen

Zusätzlich zu den bisher genannten gibt es noch generelle Anforderungen, die für alle Funktionalitäten gelten. Dies sind zum einen die Forderung nach

A12 Robustheit gegenüber ungenauer Roboterposition und

A13 Robustheit gegenüber Messungenauigkeiten der Sensoren und zum anderen

A14 die Forderung nach Modularität. Dies ist besonders für eine anwendungsbezogene Weiterentwicklung des Verfahrens zur Umgebungserfassung von großer Wichtigkeit. Auch für den Austausch von Einzelkomponenten ist es hilfreich, ein modulares Verfahren zu besitzen.

Die Anforderungen A12 und A13 werden durch Evaluierung gegenüber dem Stand der Technik sicher gestellt.

4 Stand der Technik

Basierend auf den in Kapitel 2 beschriebenen Anwendungsfeldern und der Abgrenzung der vorliegenden Arbeit wird der Stand der Technik erfasst. Dabei werden zunächst Messprinzipien und Eigenschaften von existierenden 3-D-Tiefenbildkameras untersucht. Anschließend erfolgt die Analyse von Verfahren zur Umgebungserfassung und -darstellung. Danach werden Ansätze zum teil-autonomen Roboterbetrieb betrachtet. Das Kapitel schließt mit Folgerungen aus dem Stand der Technik unter Berücksichtigung der Anforderungen.

4.1 3-D-Tiefenbildkameras

Da die Ausgangsdaten der verwendeten Sensorik eine wichtige Grundlage für das zu entwickelnde Verfahren zur Umgebungserfassung darstellen, sollen zunächst 3-D-Tiefenbildkameras detailliert betrachtet werden. Die Erzeugung von räumlichen Daten bei solchen Sensoren beruht meist auf einem von zwei physikalischen Messprinzipien, nämlich Messung der Phasenverschiebung und Triangulation. Diese werden im Folgenden erläutert und verfügbare Kameras vorgestellt. Dabei sind insbesondere auftretende Messfehler von besonderer Bedeutung für das zu entwickelnde Verfahren, da diese kompensiert werden müssen.

4.1.1 Kameras nach dem Prinzip der Phasenverschiebung

Die Messung der Phasenverschiebung ist eine besondere Ausprägung der Laufzeitmessung (Siegwart, 2004). Es wird dabei nicht die Signallaufzeit direkt gemessen, sondern der Phasenversatz von ausgesandtem und empfangenem Signal ermittelt. Aus diesem Grund wird ein amplitudenmoduliertes Signal als kontinuierliche harmonische Schwingung $g(t)$ mit der Frequenz f_m emittiert. Bei der Rückkehr des Echos $s(t)$ wird der Phasenversatz φ zwischen beiden Signalen über die Kreuzkorrelation

$$c(\tau) = s(t) \otimes g(t) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} s(t) \cdot g(t + \tau) dt \quad (4.1)$$

mit den Abtastzeitpunkten τ_i zu

$$\varphi = \arctan \left(\frac{c(\tau_3) - c(\tau_1)}{c(\tau_0) - c(\tau_2)} \right) \quad (4.2)$$

berechnet (Lange, 2000). Abbildung 4.1 zeigt einen typischen Signalverlauf. Die Messdistanz ergibt sich aus der Phasenverschiebung zu

$$D = \frac{v\varphi}{4\pi f_m}. \quad (4.3)$$

Durch die Korrelation von Sende- und Empfangssignal kann eine Distanzmessung über die

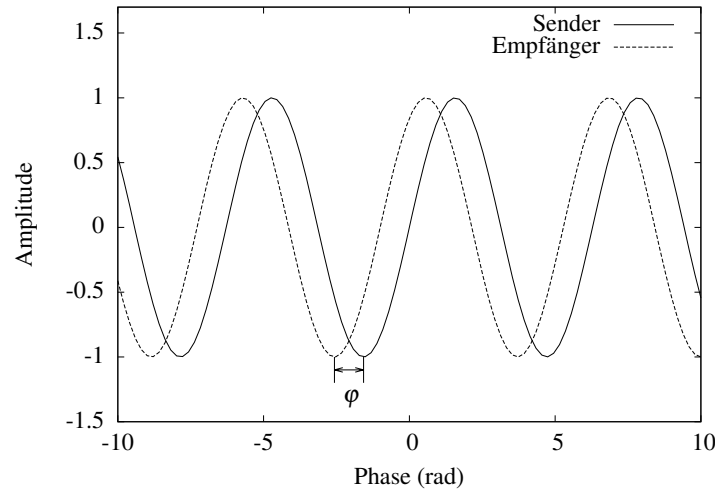


Abbildung 4.1: Phasenverschiebung zwischen ausgesandtem und empfangenem Signal

Phasenverschiebung im Gegensatz zur direkten Laufzeitmessung ohne hochpräzise Messkomponenten erfolgen. Damit ist ein Potenzial für kostengünstige Sensoren vorhanden.

Ein Hauptnachteil dieses Messprinzips ist die Uneindeutigkeit der Phasenverschiebung (*non-ambiguity range*) bei $|\varphi| > \pi$. Damit hängt die Maximalreichweite D_{max} von Sensoren unmittelbar mit der Modulationsfrequenz f_m zusammen und ergibt sich zu

$$D_{max} = \frac{c}{2f_m}. \quad (4.4)$$

Zwar gibt es einen Ansatz zur Rekonstruktion weiter entfernter Messwerte (*phase unwrapping*) (Droeschel u. a., 2010), der Rechenaufwand ist jedoch hoch und die Unterscheidung von Phasen- und geometrischen Sprüngen nicht trivial.

Kameras nach dem Prinzip der Phasenverschiebung (TOF-Kameras) besitzen einen Sender, der aus einem Feld von Infrarot-LEDs besteht. Diese senden modulierte Licht aus, typischerweise in einem Frequenzbereich um 20 MHz. Der Empfänger besteht aus einer CMOS/CCD-Kamera mit einer Auflösung von einigen tausend Pixeln. Jeder Pixel wertet das empfangene Signal unter Verwendung von (4.2) an den Abtastzeitpunkten aus. Durch Korrelation mit dem Sendesignal wird die Phasenverschiebung und schließlich der Messabstand ermittelt. Die Kamera gibt die erzeugten Daten als Tiefenbild aus, in dem jeder Pixel eine Tiefeninformation beinhaltet. Zusätzlich können noch Intensitäts- oder Amplitudenbilder ausgegeben werden. In Abbildung 4.2 sind das Intensitäts- und das Tiefenbild einer Kamera zu sehen. Diese ähneln Graubildern von nor-

malen Kameras und entstehen dadurch, dass weiter von der Kamera entfernte Objekte weniger Licht reflektieren als nähere. (Oggier u. a., 2004)

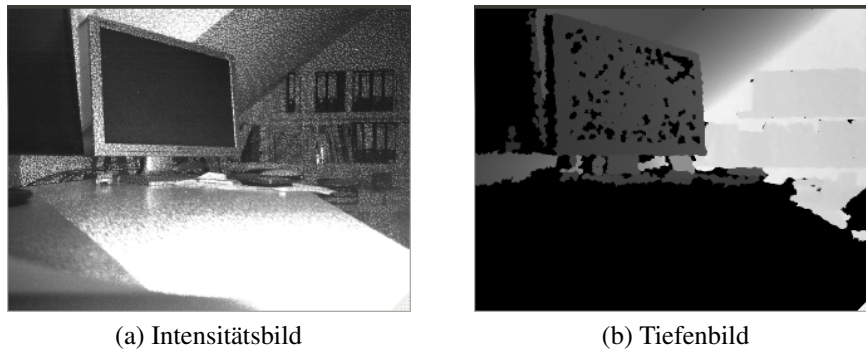


Abbildung 4.2: Beispiel für von den TOF-Kameras erzeugte Daten

Trotz großer Entwicklungsfortschritte in den letzten Jahren hinsichtlich Genauigkeit und Störanfälligkeit der Sensoren unterliegen die TOF-Kameras diversen Fehlereinflüssen, welche die Messdaten verfälschen. Untersuchung der Messfehler und deren Kompensation ist Gegenstand vieler Veröffentlichungen (Lange, 2000; Kahlmann u. Ingensand, 2008; Fuchs u. Hirzinger, 2008; Frank u. a., 2009; Gudmundsson u. a., 2007; Chiabrand, 2010; Oggier u. a., 2004). Fuchs klassifiziert in (Fuchs u. Hirzinger, 2008) zufällige und systematische Fehler.

Die wichtigsten zufälligen Fehler sind

unzureichende Illumination Eine geringe Illumination der Szene durch den Sender führt zu einem schlechten Signal-Rausch-Verhältnis und damit zu Distanzfehlern,

Abrißkanten An Kanten entstehen die Entfernungswerte wegen der diskreten Abtastzeitpunkte durch eine Mittelung von Vorder- und Hintergrundwerten und

Schrotrauschen (*Photon Shot Noise*) Durch zufällig im Empfänger eintreffende Photonen wird die Maximalreichweite eingeschränkt und die Messgenauigkeit herabgesetzt (Lange, 2000; Oggier u. a., 2004).

Diese Fehler können durch Filterverfahren reduziert werden.

Systematische Fehler hingegen können durch Kamerakalibrierung minimiert werden, da ein mathematisches Modell herleitbar ist und diese Fehler nicht von der Umgebung abhängen. Systematische Fehler sind

Verzerrtes Sendesignal Die Sende-LEDs emittieren ein Sinussignal, welches nicht perfekt ist. Dadurch entstehen Fehler bei der Berechnung der Phasenverschiebung,

Latenz Signalwege im Chip führen zu einem Offset auf der Distanzmessung und

Temperatur Nach (Kahlmann u. Ingensand, 2008) führt höhere Temperatur zu einer Messabweichung.

Erste kommerziell erhältliche Sensoren stammten von *MESA Imaging* (Mesa Imaging AG, 2012). Nach dem *Swissranger SR2* (2003) und dem *SR3000* (2005) ist das aktuelle Modell der *Swissranger SR4000* (Abbildung 4.3a) mit einer Auflösung von 176 x 144 Pixeln und Maximalreichweiten von 5 m oder 8 m.

Ein weiterer Hersteller von 3-D-Kameras ist *PMD Technologies* (PMDTec, 2012). Das aktuelle Modell ist der *PMD[vision][®] CamCube 3.0* (Abbildung 4.3b) mit einer Auflösung von 204 x 204 Pixeln. Diese Kamera ist durch aktive Rauschunterdrückung bedingt für den Außeneinsatz geeignet. Eine miniaturisierte Version mit der selben Chiptechnologie ist das in Abbildung 4.3c abgebildete *PMD[vision][®] CamBoard*, welches keine externe Stromversorgung benötigt. Allerdings hat diese Kamera wegen der eingeschränkten Möglichkeit Illumination der Umgebung schlechtere Reichweiten- und Genauigkeitseigenschaften als der *CamCube*.

Schließlich gibt es mit *ifm electronic GmbH* noch einen Hersteller von Tiefenbildkameras für den industriellen Einsatz (ifm electronic, 2012). Der *efector pmd 3d* ist mit einem Industriegehäuse ausgestattet, kann jedoch nur 64 x 48 Bildpunkte erfassen.



Abbildung 4.3: Tiefenbildkameras mit Messung der Phasenverschiebung, von verschiedenen Herstellern

4.1.2 Kameras nach dem Prinzip der Triangulation

Das zweite wichtige Messprinzip zur Entfernungsmessung bei 3-D-Tiefenbildkameras ist Triangulation. Es verwendet trigonometrische Beziehungen, um eine Distanz zum Objekt bei bekanntem Basisabstand L von Sender und Empfänger zu berechnen. Lange unterscheidet zwischen aktiver und passiver Triangulation (Lange, 2000). Im Gegensatz zu auf dem passiven Prinzip basierenden Stereokamerasystemen verwenden 3-D-Tiefenbildkameras die aktive Triangulation um die Distanz gemäß

$$D = f \frac{L}{x} \quad (4.5)$$

zu bestimmen. Hierbei stellt f die Brennweite des Empfängers und x die Position der Punktprojektion auf dem Empfänger dar. Je größer L , desto höher ist die Messgenauigkeit vor allem bei größeren Entfernungen.

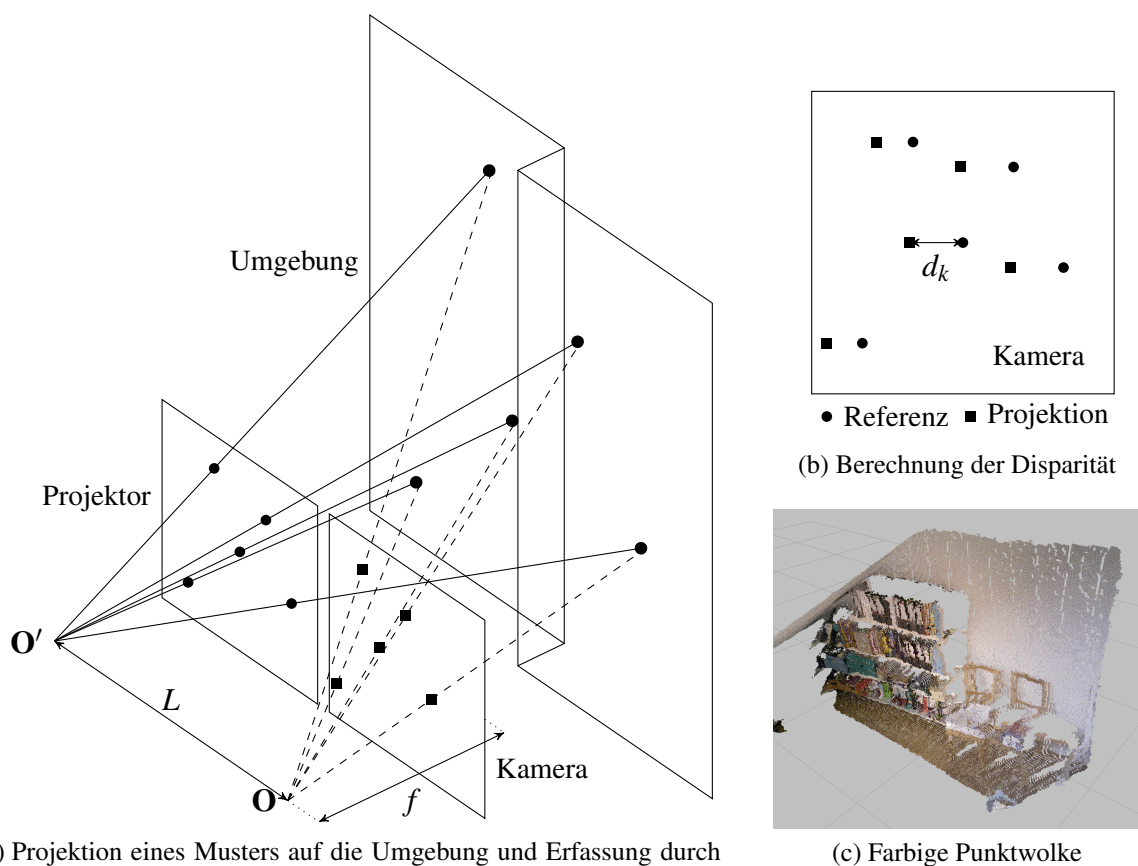


Abbildung 4.4: Messprinzip der Musterprojektion

Einen Sonderfall der aktiven Triangulationsmessung stellt die Projektion von Mustern dar (siehe Abbildung 4.4). Über einen Projektor können eindeutige oder wiederkehrende Muster auf die Umgebung geworfen und die Verzerrung des Musters beim Empfänger gemessen werden (Freedman u. a.). Durch die Musterprojektion ist eine effiziente und vor allem eindeutige Berechnung der Tiefendaten möglich. Die Auflösung hängt dabei unmittelbar mit dem Detailgrad des Musters und der Stärke des Projektors zusammen. Das zugrunde liegende Patent beschreibt einen Sensor, der über eine kohärente Lichtquelle ein Muster auf die Umgebung projiziert. Eine Infrarotkamera erfasst das Muster und kann über Verzerrungen ein Tiefenbild der Umgebung erzeugen. Über Korrelation mit einem Referenzmuster der Tiefe z_0 wird zunächst eine Disparitätskarte erzeugt (vgl. Abbildung 4.4b). Nach (Khoshelham, 2011) können daraus die Tiefenwerte

$$D_k = \frac{z_0}{1 + \frac{z_0}{fL} d_k} \quad (4.6)$$

berechnet werden. Dabei ist f die Brennweite, L der Basisabstand und d_k die Disparität für Punkt k .

Auch bei diesen Kameras können Messfehler beobachtet werden. Wie bei den TOF-Kameras gibt es systematische und zufällige Messfehler, wobei die systematischen Fehler durch eine entsprechende Kalibrierung eliminiert werden können. Wichtige Fehlereinflüsse sind

- Oberflächen unterschiedlicher Reflektivität,
- Abschattungen und
- der Quantisierungsfehler, der durch abnehmende Auflösung ρ der Disparitätskarte

$$\rho \propto \frac{1}{D^2} \quad (4.7)$$

bei zunehmender Entfernung D entsteht. Also gilt für den Quantisierungsfehler

$$e_q \propto D^2. \quad (4.8)$$

Die auf dem Markt befindlichen Kameramodelle besitzen die selbe Sensorik und unterscheiden sich nur in der zusätzlichen Ausstattung. Zu nennen sind hier *Microsoft Kinect* (Microsoft, 2012), *Asus XTion Pro* und *Asus XTion Pro Live* (ASUSTek, 2012), welche in Abbildung 4.5 dargestellt sind. Alle Kameras haben eine Auflösung von 640x480 Bildpunkten und einen Reichweitenbereich von 0,8 m bis 5 m. Einige der Modelle besitzen zusätzlich eine integrierte Farbkamera und können eine farbige Punktwolke erzeugen (Abbildung 4.4c).



(a) Microsoft Kinect (Evan-Amos, 2011)



(b) Asus XTion Pro Live (ASUSTek, 2012)

Abbildung 4.5: Tiefenbildkameras mit Triangulation von verschiedenen Herstellern

4.1.3 Bewertung der 3-D-Tiefenbildkameras

Die auf unterschiedlichen Messprinzipien beruhenden 3-D-Tiefenbildkameras sind aufgrund ihrer Eigenschaften für verschiedene Anwendungen unterschiedlich gut geeignet. So sind die Kameras nach dem Triangulationsprinzip kostengünstiger, haben eine höhere Auflösung, geringeres Messrauschen und eine bessere Tiefengenauigkeit. Dagegen eignen sich TOF-Kameras besser für Außenanwendungen, da die Anfälligkeit gegenüber Umgebungslicht geringer ist. Weiterhin ist sowohl die Minimalreichweite geringer, als auch die Maximalreichweite über die Modulationsfrequenz anpassbar, was maximale Messdistanzen von bis zu 200 m ermöglicht, allerdings bei geringerer Tiefenauflösung. Einige der 3-D-Tiefenbildkameras besitzen eine integrierte Farbkamera und können somit farbige Punktwolken ausgeben. Sollen mit anderen Sensoren ebenfalls farbige Punktwolken erzeugt werden, kann über Sensorfusion eine externe Farbkamera eingebunden werden. Im Rahmen dieser Arbeit kommen vorwiegend Kameras mit Musterprojektion zum Einsatz, da die Qualität der 3-D-Daten ein entscheidendes Kriterium für ein robustes Verfahren zur 3-D-Umgebungserfassung ist und der Fokus auf Innenanwendungen liegt. Alle Kameras

geben Punktwolken in einer geordneten Struktur, also Tiefenbilder aus. Da diese Struktur besonders bei der Betrachtung von Nachbarschaftsbeziehungen zwischen den Punkten Vorteile hinsichtlich der Recheneffizienz bietet, werden im Folgenden existierende Verfahren in Bezug auf diesen Aspekt untersucht und Neuentwicklungen speziell für die Verarbeitung von Tiefenbildern ausgelegt und optimiert.

4.2 Umgebungserfassung in der Servicerobotik

In diesem Abschnitt werden bestehende Verfahren zur Erfassung der Umgebung analysiert. Obwohl 3-D-Tiefenbildkameras als hier bevorzugter Sensortyp identifiziert wurden, erfolgt die Betrachtung auch unter Einbeziehung von Laserscannern und Stereokamerasystemen, da so auch sensortypunabhängige Ansätze erfasst werden können. Weiterhin erfolgt eine getrennte Betrachtung von Verfahren bei teleoperierten und autonomen Robotern, da es sich um weitgehend getrennte Forschungsdomänen handelt. Ziel dieses Abschnitts ist es, einerseits essentielle Teilschritte und Einzeltechnologien der Umgebungserfassung zu identifizieren und andererseits Lücken bei der Verarbeitung und Darstellung der Umgebung, insbesondere im Hinblick auf einen teil-autonomen Betrieb, aufzuzeigen.

4.2.1 3-D-Umgebungserfassung in der Teleoperation

Im Bereich der Teleoperation und Telepräsenz spielt die Anbindung eines menschlichen Bedieners an das Robotersystem eine zentrale Rolle. Viele Arbeiten befassen sich mit der Erzeugung von haptischen Effekten sowohl im Ein- als auch im Ausgang und deren Wahrnehmung durch den Bediener (Hasunuma u. a., 1999; Bauernschmitt u. a., 2009; Arbeiter u. a., 2010). Häufig werden die haptischen Signale mit optischen oder akustischen Signalen verknüpft, um ein multimodales Feedback zu erzeugen (Buss u. a., 2009; Aracil u. a., 2007) und somit die Telepräsenz zu verbessern. Da jedoch in dieser Arbeit die Verarbeitung von optischen Sensordaten im Fokus steht, werden haptische Ein- und Ausgabemodalitäten nicht weiter betrachtet und stattdessen der Einsatz optischer Sensoren in der Teleoperation untersucht. Im Speziellen werden stereoskopische Bildschirme und die Anwendung von Techniken zur virtuellen Realität und 3-D-Darstellung der Umgebung vorgestellt.

Stereoskopische Bildschirme ermöglichen dem Betrachter eine 3-D-Sicht auf die dargestellte Szene. Nach (Aracil u. a., 2007) gibt es drei Arten von Bildschirmen. *Immersive* Bildschirme decken das komplette Sichtfeld des Betrachters ab wie z. B. 360°-Bildschirme. *Autostereoskopische* Anzeigen hingegen liefern jeweils ein Bild für jedes Betrachterauge. Gleiches gilt für *binokulare* Bildschirme. Diese benötigen jedoch zusätzliche Geräte wie eine Shutterbrille oder ein *Head-mounted Display* zur Anzeige von 3-D-Information. Die Gruppe um Tachi arbeitet bereits seit 1996 an dem Telepräsenzsystem *Telesar* (Hasunuma u. a., 1999; Tachi u. a., 2011). *Telesar 4* besitzt einen immersiven stereoskopischen Bildschirm zur Telekommunikation. Jede der TWISTER genannten Einheiten besteht aus einem beweglichen Stereokamerasystem zur Aufnahme und einem LED-Bildschirmring zur Wiedergabe von stereoskopischen Bildern. In (Bell, 2003) wird ein multispektrales Kamerasystem für den Mars Rover vorgestellt, welches für Navigation und Datensammlung verwendet wird.

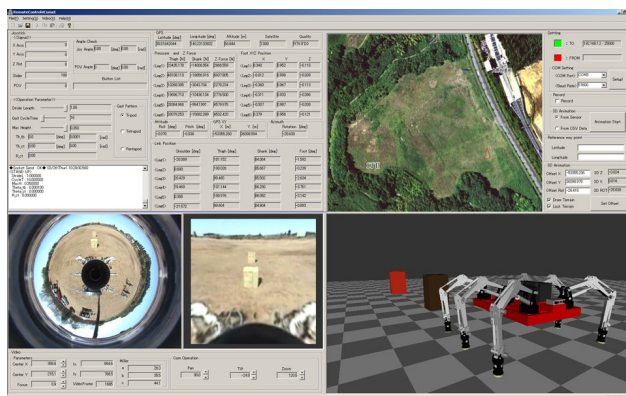
Allen betrachteten Ansätzen ist gemeinsam, dass der Fokus auf der Erzeugung von 3-D-Daten mit Stereokameras und auf deren unmittelbarer Anzeige auf verschiedenen Bildschirmen liegt. Eine Aggregation der Daten zu einem Umgebungsmodell findet bei den genannten Arbeiten nicht statt. Auch Tiefenbildkameras werden nicht eingesetzt. Somit ergibt sich eine geringe Anwendbarkeit der bisherigen Ergebnisse für die vorliegende Arbeit.

Ebenfalls ein großes Teilgebiet in der Teleoperation ist die Verwendung von *erweiterter* (AR) und *virtueller Realität* (VR). Kheddar u. a. unterscheiden zwischen *Predictive Displays* und *Teleprogramming* (Kheddar u. a., 2007). Die AR eines *Predictive Displays* ermöglicht die virtuelle Planung von Roboterbewegungen und die nachfolgende Ausführung auf dem echten Roboter. *Teleprogramming* hingegen hält eine volle virtuelle Repräsentation des Roboters und des Arbeitsraums vor, also ein Umgebungsmodell. Dadurch wird eine kontinuierliche Robotersteuerung ermöglicht. Auf Methoden zur Generierung dieser Repräsentationen geht Kheddar aber nicht ein.

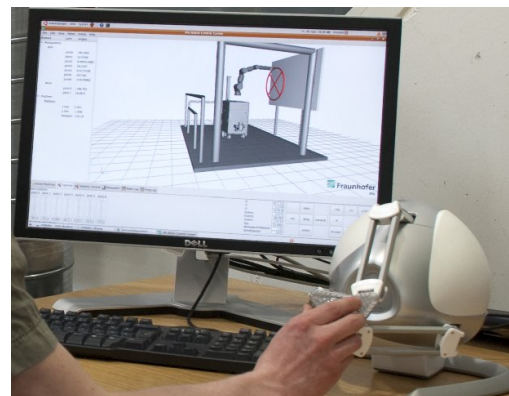
Generell liegt bei vielen Arbeiten eine völlige Entkopplung von VR und Sensoren vor. So beschreiben Bo You u. a. die virtuelle Regelung in VR unter Verwendung eines Umgebungsmodells (Bo You u. a., 2010). Dieses wird jedoch manuell ohne Sensorik generiert. Gleiches gilt für die Ansätze in (Groenda u. a., 2005), die sich mit VR und Telepräsenz in Computerpielen beschäftigen. In (Jiang u. a., 2006) wird die Berechnung virtueller Kräfte in VR beschrieben, jedoch auch keine Sensorik zur Erzeugung der benötigten Daten verwendet.

Im Gegensatz dazu ist beim Einsatz von mobilen Robotern in der Teleoperation häufig eine Visualisierung von Sensordaten zu finden. Ein mobiler Manipulator nutzt in (Notheis u. a., 2010) eine AR für die Nutzerunterstützung, in der neben CAD-Modellen auch Punktwolken einer 3-D-Tiefenbildkamera als Überlagerung im Farbkamerastream angezeigt werden. Der Roboter in (Ohroku u. Nonami, 2009) hingegen nutzt eine omnidirektionale Farbkamera, ein Stereokamerasystem und einen Laserscanner zur Umgebungserfassung. In der in Abbildung 4.6a zu sehenden Visualisierung kann das Farbkamerabild und eine 3-D-Ansicht des Roboters mit Hindernissen angezeigt werden. Diese kommen jedoch nicht aus einem Umgebungsmodell, sondern werden manuell in VR generiert. Einen Schritt weiter gehen Sugimoto u. a. in (Sugimoto u. a., 2005), die während der Roboterbewegung alle zurückliegenden Bilder einer Farbkamera abspeichern, um später eine exozentrische Ansicht erzeugen zu können. Zur Erstellung einer 3-D-Karte werden die Bilder aber nicht verwendet. In (Arbeiter u. a., 2010) werden durch Sensorik erkannte Stellräder in die 3-D-Ansicht einer manuell generierten Umgebungskarte eingeblendet (Abbildung 4.6b). Diese können durch Anklicken vom Benutzer ausgewählt werden. Verwendet wird aber nur das aktuelle Sensorbild und keine in der Vergangenheit liegenden Daten.

Einige Ansätze in der Literatur nutzen bereits Sensordaten von teleoperierten Robotern, um Umgebungsmodelle zu generieren und diese anzuzeigen. So erzeugen Dupuis u. a. *Digital Elevation Maps* (DEM) für den *Mars Exploration Rover* (Dupuis u. a., 2005). Eine Kartenaktualisierung unter Benutzung von LIDAR Sensoren dauert hier 1-2 Minuten. Der Ansatz in (Saitoh u. a., 2006) erzeugt ein Umgebungsmodell aus Farbkamera- und Laserdaten mit einer Updaterate von 18 s. Die entstehende Karte wird in das Kamerabild augmentiert und kann für die Navigation verwendet werden. Eine Tiefenbildkamera wird in (Sheh u. a., 2006) eingesetzt, um Terrainkarten für einen Bergungsroboter zu erstellen. Eine Interaktion des Bedieners mit der Karte ist aber nicht möglich. Ein höher entwickelter Ansatz mit kontinuierlichem Kartenupdate findet sich in (Naga-



(a) Benutzerinterface zur Steuerung eines sechsbeinigen Laufroboters. (Ohroku u. Nonami, 2009)



(b) GUI für die teleoperierte Inspektion und Wartung. (Arbeiter u. a., 2010)

Abbildung 4.6: Beispiele für Visualisierung in der Teleoperation

tani u. a., 2008, 2009, 2010). Ein schwenkbarer Laserscanner wird zur Datenakquisition genutzt. Aus den Sensordaten wird während der Roboterbewegung eine DEM gebaut. Diese kann dann über Netzwerk zur Bedienerseite transferiert und dort in 3-D visualisiert werden. All diesen Ansätzen ist gemein, dass es sich um kleine Roboter für den Außeneinsatz, häufig für die Bergung handelt. Keiner der Roboter besitzt einen Manipulator, der Fokus liegt im Allgemeinen auf der Navigation der Roboter.

Insgesamt ist zu beobachten, dass nur in wenigen der vorgestellten Arbeiten die kontinuierliche Erzeugung konsistenter 3-D-Karten stattfindet. Häufig werden nur die Sensorrohdaten beim Bediener angezeigt oder eine manuell erstellte VR benutzt. Falls ein Umgebungsmodell generiert wird, dann meistens nicht in Echtzeit, oft vergehen Sekunden oder Minuten zwischen Kartenaktualisierungen. Die Benutzerschnittstellen für die mobilen Roboter sind minimalistisch. Eine intuitive Interaktion mit den Umgebungsdaten ist meist nicht möglich. Daraus kann abgeleitet werden, dass die Anforderungen A6 und A11 durch die genannten Ansätze nicht erfüllt werden können.

4.2.2 3-D-Umgebungserfassung in der autonomen Robotik

Gerade in den letzten Jahren wurde eine Vielzahl von Methoden für die 3-D-Umgebungserfassung mit autonomen Systemen entwickelt. Im Folgenden werden Arbeiten, gegliedert nach verwendeten Sensortypen, vorgestellt, der Funktionsumfang analysiert und gemeinsame Verarbeitungsschritte und Einzeltechnologien identifiziert.

In den 90er Jahren des vorigen Jahrhunderts wurden erste Arbeiten zur Erstellung von metrischen 3-D-Karten publiziert. Mit der Verfügbarkeit von Laserscannern war es erstmals möglich, genaue Repräsentationen der Umgebung zu erstellen. Sequeira u. a. stellen in (Sequeira u. a., 1999) einen solchen Ansatz vor. Der Roboter fährt dabei durch die Szene und nimmt an bestimmten Stellen Laserscans auf. Diese werden über einen merkmalsbasierten, iterativen Algorithmus registriert. Schließlich wird über *Meshing* ein geometrisches Umgebungsmodell erzeugt und die Texturen

von Farbkamerabildern hinzugefügt. Eine SLAM-basierte Herangehensweise zeigen Thrun u. a. in (Thrun u. a., 2003). Das Verfahren erzeugt zunächst eine konsistente 2-D-Karte aus Laserscans und erweitert diese dann durch 3-D-Information.

Aufbauend auf diesen frühen Arbeiten entwickelten sich mehrere Kartierungssysteme unter Verwendung von schwenkbaren Laserscannern. In (Grisetti u. a., 2010) wird ein graph-basiertes SLAM-Verfahren zur Erstellung von 3-D-Punktekarten gezeigt. Diese werden als Voxelkarte in einer so genannten *Octomap* speichereffizient abgelegt. Zeitgleich entwickelte Nüchter ein ähnliches System zum Aufbau von Punktekarten. Über iterative Registrierung und einen Graphen werden konsistente Umgebungskarten, auch von größeren Bereichen, erzeugt. In einem weiteren Schritt werden Strukturen wie Böden oder Wänden semantische Labels zugewiesen (Nüchter, 2009). In (Rusu, 2009) stellt Rusu eine Kartierungspipeline für die Erstellung semantischer Karten für einzelne Szenen vor. In dieser Arbeit werden vor allem neue Einzeltechnologien für die Umgebungserfassung beleuchtet. Die Zusammenschaltung der Einzelkomponenten zu einer Pipeline ist laut Autor extrem anwendungsabhängig.

Ein Kartierungssystem unter Verwendung von Stereokameras, IMU und 2-D-Laserscanner wird in (Iocchi u. Pellegrini, 2007) beschrieben. Es handelt sich dabei um ein Offline-Verfahren, welches Sensordaten während der Roboterbewegung aufzeichnet und erst später verarbeitet. Das System erzeugt zunächst mittels 3-D-SLAM eine Punktekarte. Aus dieser werden mittels probabilistischer Hough-Transformation und neuronalen Netzen geometrische Strukturen wie Wände oder Treppen extrahiert und anschließend mit Textur belegt. Für einen Bediener ist es möglich, Objekte zu modifizieren.

Seit dem Markteintritt von TOF-Kameras wurden auch für diese Sensoren Verfahren zur Umgebungserfassung entwickelt. In (May u. a., 2009) werden die in (Nüchter, 2009) beschriebenen Verfahren auf die besonderen Eigenschaften der Tiefenbildkameras optimiert. Dadurch ist es möglich, auch mit dieser Sensorart 3-D-Punktekarten zu erstellen. Auf eine Extraktion von Geometrien wird jedoch verzichtet. Holz u. a. zeigen die anwendungsorientierte Segmentierung von Objekten auf Tischen unter Verwendung von TOF-Kameras (Holz u. a., 2011). In (Biswas u. Veloso, 2012) und (Poppinga u. a., 2008) wird die Erstellung von Ebenenkarten aus Tiefenbildern gezeigt. Die Arbeiten beinhalten eine schnelle Ebenenextraktion und die Erzeugung von Hüllpolygonen für die Ebenensegmente. Biswas zeigt zusätzlich, wie Polygone über mehrere Ansichten fusioniert werden können.

Mit der deutlichen Verbesserung der Sensordatenqualität durch das Erscheinen von *Microsoft Kinect* und ähnlichen Sensoren eröffneten sich schließlich nochmals neue Möglichkeiten für die Umgebungserfassung. Teilweise wurden für Laserscanner entwickelte Verfahren für die neuen Sensoren modifiziert, wie z. B. in (Sturm u. a., 2011; Engelhard u. a., 2011). Die Autoren nutzen die Tatsache aus, dass 3-D-Tiefenbildkameras auch Farbinformation ausgeben, und modifizieren das in (Grisetti u. a., 2010) vorgestellte Verfahren, indem sie texturbasierte SURF-Merkmale für die Registrierung von Punktwolken verwenden. Teilweise gibt es auch völlig neue Entwicklungen wie in (Henry u. a., 2012). Bei diesem Verfahren werden Karten aus Oberflächenelementen (SURFELS) aufgebaut. Diese beschreiben einen kleinen Oberflächenausschnitt sowohl geometrisch als auch durch Textur. Der Fokus liegt auf der Registrierung von Punktwolken mit einer Kombination aus RANSAC und ICP. Eine weitergehende geometrische Interpretation findet nicht statt. In der Anwendung wird die Kartierung eines ganzen Stockwerks gezeigt. Izadi

u. a. zeigen in (Izadi u. a., 2011) mit *KinectFusion* die Erstellung von detaillierten Oberflächenkarten bei freier Kamerabewegung. Es werden keine neuen Algorithmen vorgestellt, sondern die effiziente und optimierte Anwendung von Methoden aus dem Stand der Technik mit einer GPU-Implementierung verbunden. Dies ermöglicht eine extrem schnelle Verarbeitung der Sensordaten, Erstellung von Punktekarten und Ableitung von Oberflächenmodellen für kleine Umgebungen.

Weiterführende Arbeiten beschreiben die semantische Klassifizierung von Umgebungen. So findet sich in (Pronobis, 2011) ein Ansatz zur topologischen Beschreibung. Es werden lokale semantische Beziehungen von Objekten und Räumen ausgenutzt, um eine topologische Karte zu erstellen. Mit Hilfe dieser kann der Roboter dann abstrakte Aktionsbefehle umsetzen. Mozos stellt ebenfalls Ergebnisse zur semantischen Klassifizierung vor, wobei der Fokus auf größeren Strukturen wie Räumen liegt (Mozos, 2010). So ist das System z. B. in der Lage, Büroräume von Fluren zu unterscheiden und diese durch Türdurchgänge topologisch zu verknüpfen.

Es ist zu erkennen, dass es im Stand der Technik sehr viele verschiedene Ansätze zur Umgebungserfassung gibt. Dies liegt zum einen am Einsatz unterschiedlicher Sensorsysteme für die selben Anwendungen und zum anderen daran, dass bisher kein optimales Verfahren entwickelt wurde, das für eine Vielzahl von Anwendungsfällen gleichermaßen gut eingesetzt werden kann. Zunächst kann aber trotzdem der Eindruck entstehen, dass die Anforderungen dieser Arbeit durch eines der bestehenden Ansätze bereits erfüllt werden können. Aus diesem Grund ist eine detaillierte Betrachtung der verwendeten Einzeltechnologien notwendig. Nur diese kann die tatsächliche Leistungsfähigkeit der Verfahren beurteilen und Lücken deutlich machen.

Aus den vorgestellten Ansätzen lassen sich essentielle Einzeltechnologien für ein System zur Umgebungserfassung ableiten. Häufig ist der erste Schritt die

- **Vorverarbeitung** der Sensordaten, um die Qualität zu erhöhen oder weitere Verarbeitungsschritte zu ermöglichen.
- Weitere grundlegende Schritte bei der Punktwolkenverarbeitung sind die **Nachbarschaftssuche** und
- die Berechnung von **Normalenvektoren**.
- Besonders, aber nicht ausschließlich für TOF-Kameras sind **Rauschfilter** wichtig, um Messfehler zu kompensieren.
- Weiterhin führen fast alle Systeme eine **Registrierung** der Sensordaten durch, damit diese in einem gemeinsamen Koordinatensystem vorliegen.
- Bei einigen Verfahren werden **Merkmale** zur Verbesserung der Leistungsfähigkeit eingesetzt.
- Zur Ableitung höherwertiger Information wie Geometrien gibt es keine einheitliche Vorgehensweise. Häufig wird eine **Segmentierung** zur Unterteilung von Punktwolken in zusammengehörige Regionen durchgeführt. Auch hier können Merkmale die Segmentierung unterstützen oder zur Klassifikation der Regionen eingesetzt werden.
- Ebenfalls häufig kommen Verfahren zur **Oberflächenrekonstruktion** zum Einsatz.
- Allen vorgestellten Systemen ist gemeinsam, dass sie die verarbeiteten Daten in einer **Kartenrepräsentation** ablegen.

	Kantenlänge Voxel (m)			
	original	0,01	0,03	0,05
Anzahl Punkte	307 200	51 336	6 501	2 425
Reduktion auf (%)	100	16,7	2,1	0,8

Tabelle 4.1: Reduktion der Punktdichte im Voxelfilter bei unterschiedlichen Auflösungen

4.3 Einzeltechnologien für 3-D-Umgebungserfassung

In diesem Abschnitt werden die im vorherigen Abschnitt gefundenen benötigten Einzeltechnologien für ein System zur Umgebungserfassung genauer beleuchtet und deren Eignung für die vorliegende Arbeit analysiert.

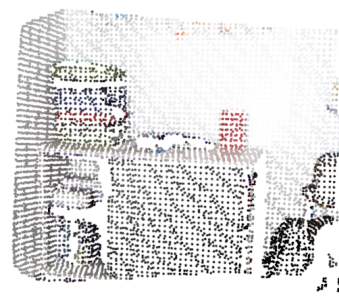
4.3.1 Vorverarbeitung

Die Vorverarbeitung dient dazu, die Sensordaten in eine für weitere Verarbeitungsschritte geeignete Form zu bringen, sei es durch Resampling, Glättung oder Fusion.

Der Voxelfilter (Rusu u. Cousins, 2011) ist ein wirkungsvolles Werkzeug, um über Resampling Punktwolken mit gleichmäßiger Punktdichte zu erzeugen. Der Filter baut eine 3-D-Gitterstruktur auf, sortiert die Punkte ein und gibt die Mittelpunkte belegter Gitterzellen aus. Die gewünschte Auflösung wird vorgegeben, der Filter entfernt dann alle Punkte, die enger beieinander liegen. Somit kann das Datenvolumen einer Punktwolke bei steuerbarem Informationsverlust reduziert werden. In Abbildung 4.7 ist eine Punktwolke vor und nach Anwendung des Voxelfilters zu sehen. Die Auflösung der heruntergerechneten Punktwolke beträgt 0,03 m. Tabelle 4.1 zeigt, zu



(a) Ungefiltert



(b) Auflösung reduziert auf 0,03 m

Abbildung 4.7: Anwendung des Voxelfilters

welchem Grad die Punktdichte bei unterschiedlicher Auflösung des Voxelfilters reduziert wird. Bei 0,03 m Kantenlänge ergibt sich beispielsweise eine Reduktion auf 2,1% des ursprünglichen Datenvolumens.

Für die Glättung von Punktwolken existiert mit *Moving Least Squares* (MLS) eine Methode, die lokale Polygone in die Oberfläche einer Punktwolke einpasst (Marton u. a., 2009). Dadurch kann einerseits Rauschen reduziert, andererseits können kleine Löcher geschlossen werden. Das Verfahren ist jedoch nicht echtzeitfähig.

Sollen die Daten mehrerer Sensoren in Beziehung zueinander gesetzt werden, ist eine Fusion der Daten notwendig. Dies ist beispielsweise dann sinnvoll, wenn die Punktwolke einer TOF-Kamera und das Bild einer Farbkamera zu einer farbigen Punktwolke zusammengesetzt werden, um die Information beider Datentypen gemeinsam verwenden zu können. Im Rahmen dieser Arbeit kann die Farbinformation beispielsweise dazu genutzt werden, eine besser verständliche Visualisierung zu erzeugen. Einige 3-D-Kameras wie z.B. *Microsoft Kinect* erzeugen bereits selbst eine farbige Punktwolke und benötigen deshalb keine Sensorfusion.

Um eine Sensorfusion durchführen zu können, müssen die Sensoren zunächst zueinander kalibriert werden (Forsyth u. Ponce, 2003). Dazu werden mit beiden Kameras Bilder eines bekannten Objekts (beispielsweise eines Schachbretts) aufgenommen und dann miteinander in Beziehung gesetzt. Die Kalibrierung liefert zum einen die intrinsischen Parameter beider Kameras und zum anderen die Transformation zwischen den Sensoren. Die Parameter und die Transformation können dann verwendet werden, um die 3-D-Daten der Punktwolke in das Farbkamerasystem zu transformieren und zu projizieren. Dadurch kann jedem 3-D-Punkt ein Farbwert zugeordnet werden.

Ein Punkt \mathbf{x}_w in Weltkoordinaten wird zunächst über

$$\mathbf{x}_c = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_w \\ 1 \end{pmatrix} \quad (4.9)$$

in das Kamerakoordinatensystem transformiert. Im nächsten Schritt

$$\mathbf{x}_n = \begin{pmatrix} \frac{X_c}{Z_c} \\ \frac{Y_c}{Z_c} \\ Z_c \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} \quad (4.10)$$

erfolgt gemäß dem Lochkameramodell die Projektion des Punktes auf die Bildebene. Anschließend wird die Verzerrung der Kameralinse mit Hilfe der radialen Verzerrungskoeffizienten \mathbf{k}_c über die Beziehung

$$\mathbf{x}_d = \left(1 + k_{c,0}r^2 + k_{c,1}r^4 + k_{c,2}r^6\right) \mathbf{x}_n \quad (4.11)$$

korrigiert. Dabei gilt $r^2 = x^2 + y^2$. Schließlich werden die Pixelkoordinaten \mathbf{x}_p des Punktes über die intrinsische Kameramatrix zu

$$\begin{pmatrix} \mathbf{x}_p \\ 1 \end{pmatrix} = \begin{pmatrix} f_{c,x} & 0 & u_0 \\ 0 & f_{c,y} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_d \\ 1 \end{pmatrix} \quad (4.12)$$

berechnet. Die Matrixeinträge $f_{c,x}$ und $f_{c,y}$ beschreiben die Brennweite, die Werte u_0 und v_0 den tatsächlichen Bildmittelpunkt.

Die in diesem Abschnitt beschriebenen Methoden stellen nützliche Werkzeuge dar, die im Rahmen dieser Arbeit verwendet werden, um die Sensordaten in die gewünschte Form zu bringen. Neuentwicklungen auf diesem Gebiet sind nicht nötig.

4.3.2 Nachbarschaftssuche in Punktwolken

Die Suche nach benachbarten Punkten in Punktwolken ist ein essentieller Bestandteil vieler Verarbeitungsschritte. Typischerweise wird dabei zu einem Abfragepunkt \mathbf{p}_q eine Menge von Nachbarpunkten \mathcal{P}_k bestimmt. Die Punkte können entweder die k nächsten Punkte zum Abfragepunkt sein (*k search*) oder es werden alle Punkte in einem bestimmten Radius gesucht (*radius search*). Für die meisten Verfahren ist die Radiussuche von größerer Wichtigkeit.

Bei der Suche in ungeordneten Daten muss theoretisch der Abstand zwischen Abfragepunkt \mathbf{p}_q und jedem anderen Punkt der Punktwolke bestimmt werden, um alle Nachbarpunkte \mathbf{q}^k zu finden. Dies ist sehr aufwändig, beansprucht daher viel Rechenzeit und ist kein sinnvoller Ansatz für Echtzeitanwendungen. Eine vielversprechende Herangehensweise ist deswegen die *Approximate Nearest Neighbor* (ANN) Suche. Die meisten Ansätze (Arya u. Mount, 1993; Nüchter u. a., 2007a; Connor u. Kumar, 2010) unterteilen dabei den Suchraum orthogonal und führen eine Schranke $\varepsilon > 0$ ein, die einen maximalen Fehler für die Lösung vorgibt. Für jeden gefundenen Nachbarpunkt \mathbf{p}_i^k gilt das Verhältnis

$$\|\mathbf{p}_i^k - \mathbf{p}_q\| \leq (1 + \varepsilon) \|\mathbf{q}^k - \mathbf{p}_q\| \quad (4.13)$$

zum echten Nachbarpunkt \mathbf{q}^k .

Den genannten Verfahren ist gemeinsam, dass sie die geordnete Struktur von Tiefenbildern nicht ausnutzen, um eine effiziente Suche auf Pixelbasis durchzuführen. Deswegen können sie ohne ein Downsampling der Daten nicht in Echtzeit rechnen. Aus diesem Grund wird im Rahmen dieser Arbeit eine neue Methode zur Nachbarschaftssuche auf geordneten Punktwolken entwickelt.

4.3.3 Normalenvektoren

Punktnormalen sind ein relativ einfaches Mittel, um lokale Geometrien in Punktwolken zu beschreiben. Die Berechnung ist unkompliziert, jedoch ist auch die Beschreibungskraft gering. Weiterhin sind Normalen nur bedingt robust gegenüber Rauschen. Dennoch sind Punktnormalen in dieser Arbeit relevant, da sie häufig als Grundlage für nachfolgende Berechnungen dienen. Hoppe u. a. erzeugen in (Hoppe u. a., 1992) Normalen durch Einpassen von Tangentialebenen in die Nachbarschaft jedes Punkts. Diese Methode ist gut geeignet für stetige Oberflächen, die Komplexität der Berechnungen ist jedoch hoch (Rusu, 2009). Darauf aufbauend stellt Rusu in (Rusu u. a., 2008a) eine Methode vor, die durch einen kombinierten Ansatz aus *Principal Component Analysis* (PCA) und RMSAC auch in der Nähe von Kanten korrekte Punktnormalen erzeugt.

Zunächst wird eine Nachbarschaftssuche in der Punktwolke durchgeführt. Für jeden Nachbarschaftsbereich \mathcal{P}_k mit k Punkten wird nachfolgend der Schwerpunkt

$$\bar{\mathbf{p}} = \frac{1}{k} \sum_{i=1}^k \mathbf{p}_i \quad (4.14)$$

berechnet. Im nächsten Schritt erfolgt die Ableitung der Kovarianzmatrix gemäß

$$\mathbf{C} = \frac{1}{k} \sum_{i=1}^k (\mathbf{p}_i - \bar{\mathbf{p}}) (\mathbf{p}_i - \bar{\mathbf{p}})^T. \quad (4.15)$$

Für die Berechnung der Eigenwerte λ_j und Eigenvektoren \mathbf{v}_j von \mathbf{C} gilt $\lambda_1 > \lambda_2 > \lambda_3$ (vgl. Anhang C). Der zum kleinsten Eigenwert gehörige Eigenvektor ist identisch mit dem Normalenvektor

$$\mathbf{n}_q = \mathbf{v}_3. \quad (4.16)$$

Die zu den anderen beiden Eigenwerten gehörigen Eigenvektoren stellen Tangenten an die lokale Oberfläche dar. Ein Nebenprodukt dieses Verfahrens ist daher die Krümmung c_i der lokalen Oberfläche, welche sich zu

$$c_i = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \quad (4.17)$$

ergibt. Das Verfahren rechnet auch an Kanten genau, benötigt aber eine große Rechenzeit.

Eine besonders schnelle Berechnung von Punktnormalen auf Tiefenbildern hingegen zeigen Holz u. a. in (Holz u. a., 2012). Das Verfahren nutzt das Kreuzprodukt von zu einer Nachbarschaft tangentialen Vektoren, um die Normale zu bestimmen. Um das Rauschverhalten zu verbessern, werden die Tangentialvektoren über Integralbilder geglättet. Probleme hat das Verfahren an Kanten, da die Berechnung von Integralbildern einen geschlossenen Bereich aus gültigen Tiefenwerten um den Abfragepunkt benötigt. Bilder von 3-D-Tiefenbildkameras liefern aber gerade an Kanten vereinzelt ungültige Tiefenwerte.

Für die vorliegende Arbeit ist die Berechnung von Normalenvektoren als Basis für komplexere Merkmale oder die Segmentierung relevant. Da die genannten Ansätze entweder zu langsam rechnen oder nicht an allen Punkten Normalen liefern, muss ein neues Verfahren zur Berechnung von Normalenvektoren auf Tiefenbildern entwickelt werden.

4.3.4 Rauschfilter

Vor allem um die Messfehler von TOF-Kameras (vgl. Abschnitt 4.1.1) zu kompensieren, aber auch um allgemein Gaußsches Rauschen zu reduzieren, gibt es in der Literatur Ansätze für Rauschfilter. In (Fuchs u. May, 2008) finden sich der Amplitudenfilter zum Ausgleich von Fehlmessungen aufgrund schwacher oder Mehrfachreflexion und der Abrisskantenfilter.

Es gibt aber auch generelle Filterverfahren unabhängig vom Sensortyp. In (Rusu u. a., 2007; May u. a., 2008) wird der statistische Ausreißerfilter (SAF) vorgestellt, der Ausreißerpunkte in einer Punktwolke identifizieren und eliminieren kann. Zunächst wird der mittlere Abstand

$$d(\mathbf{p}_q) = \frac{1}{k} \sum_{n=1}^k \|(\mathbf{p}_q - \mathbf{q}_k)\| \quad (4.18)$$

der k nächsten Nachbarn um den Punkt \mathbf{p}_q für jeden Punkt der Punktwolke berechnet. Nachfolgend wird ein Punkt als Ausreißer markiert, wenn

$$d(\mathbf{p}_q) > d_{th} \quad (4.19)$$

erfüllt ist.

Zu den TOF-Filtern findet sich in der Literatur wenig Auswertung hinsichtlich der Parametereinstellung. Weiterhin ist unklar, wie die Filter hintereinander geschaltet werden können, um ein besseres Filterergebnis zu erzielen. Der statistische Ausreißerfilter kann die Vorteile von Tiefenbildern nicht ausnutzen, deswegen rechnet er langsam. Hier muss im Rahmen dieser Arbeit ein neuer Ansatz zur effizienten Rauschreduktion gefunden werden.

4.3.5 3-D-Merkmale

Merkmale sind immer dann wichtig, wenn Sensordaten klassifiziert oder bestimmte markante Punkte im Raum wiedererkannt werden sollen. Aus den Punktwolken von 3-D-Tiefenbildkameras werden häufig 3-D-Merkmale extrahiert, die geometrische Eigenschaften beschreiben.

Bei den lokalen Punktmerkmalen gibt es zwei Kategorien, solche, die vor allen die Oberflächeneigenschaften um den Punkt beschreiben und solche, die einen robusten Deskriptor berechnen, der gut assoziiert werden kann. Die in (Sadjadi u. Hall, 1980) vorgestellten *Moment Invariants* beschreiben Punkte über ihre dreidimensionalen Momente. Deskriptoren, die Intensitätsinformationen in der Punktwolke benötigen, sind beispielsweise *Intensity Gradients* (Rusu u. Cousins, 2011) oder *Intensity Spin Images* (Lazebnik u. a., 2005; Johnson u. Hebert, 1999). Bei beiden Verfahren wird ein Deskriptor aus den Intensitätsdaten abgeleitet, die indirekt eine Aussage über die Oberflächengeometrie machen. Steder zeigt in (Steder u. a., 2011) NARF-Merkmale auf Tiefenbildern. Diese können besonders gut Kanteneigenschaften beschreiben. Mit RIFT (Skelly u. Sclaroff, 2007) existiert eine dreidimensionale Variante von SIFT (Lowe, 2004) mit ähnlichen Beschreibungseigenschaften. Der SHOT Deskriptor erreicht eine höhere Genauigkeit bei der Datenassoziation durch Kombination von Farb- und Tiefendaten (Tombari u. a., 2011).

Merkmale, die sowohl zur Assoziation als auch zur Oberflächenbeschreibung eingesetzt werden können, sind *Point Feature Histograms* bzw. *Fast Point Feature Histograms* (FPFH) (Rusu u. a., 2008c, 2009b). Aus geometrischen Relationen zwischen Nachbarpunkten werden Histogramme erzeugt, die zum einen Deskriptoren darstellen und zum anderen durch ein maschinelles Lernverfahren Punkte Oberflächenklassen zuordnen. Zwei ähnliche Verfahren, die ausschließlich zur Oberflächenbeschreibung geeignet sind, sind *Radius-based Surface Descriptor* (RSD) (Marton u. a., 2010) und *Principal Curvatures* (PC) (Rusu u. Cousins, 2011). Während RSD die Krümmungsradien von lokalen Oberflächen betrachtet, berechnet PC die Krümmungen selbst.

Die Vielzahl der existierenden Merkmalstypen lässt darauf schließen, dass bislang kein optimaler Deskriptor existiert. Weiterhin gibt es kaum vergleichende Evaluierungen. Jedoch finden sich in der Literatur vielversprechende Ansätze, so dass im Rahmen dieser Arbeit kein neuer Merkmalspunkt entwickelt werden muss. Vielmehr sollen ausgewählte Methoden gegeneinander evaluiert werden.

4.3.6 Registrierung

Eine Registrierung wird benötigt, um Punktwolken von verschiedenen Aufnahmen in einem gemeinsamen Koordinatensystem vereinen zu können. In den letzten 20 Jahren wurden enorm viele Ergebnisse zur Registrierung für Punktwolken publiziert. Der wohl am häufigsten verwendete Algorithmus ist *Iterative Closest Point* (ICP). Dieses Verfahren assoziiert zunächst die Punkte \mathbf{a}_i , \mathbf{b}_j von zwei Punktwolken mit n Punkten aufgrund der euklidischen Distanz, schätzt danach die Transformation $\mathbf{T} = (\mathbf{R}, \mathbf{t})$ zwischen den Punktpaaren, in dem es die Fehlerfunktion

$$E(\mathbf{T}) = \sum_{i=1}^n \sum_{j=1}^n w_{i,j} \left\| \mathbf{a}_i - (\mathbf{R}\mathbf{b}_j + \mathbf{t}) \right\|^2 \quad (4.20)$$

minimiert, wobei $w_{i,j} = 1$, falls \mathbf{a}_i und \mathbf{b}_j korrespondieren, andernfalls $w_{i,j} = 0$. Anschließend werden alle Punkte mit \mathbf{T} transformiert. Dies wird solange wiederholt, bis ein lokales Minimum erreicht ist. Ursprünglich von Besl u. a. in (Besl u. McKay, 1992) veröffentlicht, gibt es inzwischen eine Vielzahl von Varianten. Diese reichen von unterschiedlicher Selektion von Untermengen über modifizierte Fehlerfunktionen bis hin zur Verwendung von Gewichts- oder Kontrollfunktionen (Rusinkiewicz u. Levoy, 2001). Auch für die Berechnung der Transformation gibt es unterschiedliche Möglichkeiten, z. B. die direkte oder approximierte Lösung. In (Nüchter, 2009) wird gezeigt, dass es kaum Unterschiede zwischen den Transformationsberechnungen gibt. Die meisten ICP-Varianten setzen *Singular Value Decomposition* (SVD) (Arun u. a., 1987) ein.

Ein Nachteil des ursprünglichen ICP-Algorithmus sind schwache Korrespondenzen durch Assoziation der Rohpunkte der Punktwolken. Deswegen gibt es viele ICP-Varianten, die Merkmale einsetzen, um die Korrespondenzen zwischen den Punktwolken robuster zu machen. Die Registrierung, häufig auch nur der initiale Schritt, wird dann nur noch auf den Merkmalspunkten durchgeführt, um das globale Minimum zu finden. Johnson schlägt in (Johnson u. Kang, 1997) vor, Farbkorrespondenzen aus Kamerabildern hinzuzuziehen. Andere Veröffentlichungen verwenden Punktnormalen (Alexa u. a., 2003; Rusu u. a., 2007; Izadi u. a., 2011) oder Intensitätswerte (Akca, 2007). Komplexere Merkmalstypen werden z. B. in (Rusu u. a., 2008b) betrachtet. Rusu zeigt, dass unter Verwendung von *Integral Volume Descriptors* die initiale Transformation robuster wird. In (Engelhard u. a., 2011) und (Henry u. a., 2012) werden hingegen Texturmerkmale aus Farbbildern für die Registrierung verwendet. Diese Variante wird im Folgenden als *Feature ICP* (FICP) bezeichnet.

Segal u. a. zeigen in (Segal u. a., 2009) eine generalisierte Variante von ICP. Diese ist in der Lage, verschiedene Fehlermetriken abzubilden und damit ebenfalls stärkere Korrespondenzen

zwischen den Datensätzen zu ermöglichen. Die Fehlerfunktion

$$E(\mathbf{T}) = \sum_{i=1}^n \sum_{j=1}^n w_{i,j} \left\| \mathbf{n}_i (\mathbf{a}_i - (\mathbf{R}\mathbf{b}_j + \mathbf{t})) \right\|^2 \quad (4.21)$$

wird dabei abhängig von der Umgebung der Punkte (Normalenvektoren \mathbf{n}_i) gewichtet. Das Verfahren ist robuster als ICP, da mehr Information über die Punkte zur Korrespondenzfindung genutzt wird.

Eine Erweiterung von ICP ist *frustum ICP* (May u. a., 2009; Luck u. a., 2000). Durch Berücksichtigung des Sichtfelds des Sensors wird der Überlappungsbereich der zu registrierenden Punktwolken erkannt. Dies kann in bestimmten Situationen sowohl die Registrierungsgenauigkeit als auch die Geschwindigkeit erhöhen (Arbeiter u. a., 2011).

Neben ICP gibt es auch noch andere Ansätze. Bei *Normal Distributions Transform* werden beispielsweise die Punkte über eine Gitterstruktur in Normalverteilungen umgerechnet. Über das Newton-Verfahren wird dann der Fehler zwischen den Verteilungen zweier Punktwolken minimiert und somit die Transformation bestimmt. Das Verfahren weist eine hohe Datenreduktion auf und ist deswegen gut für große Karten geeignet, wurde aber für Sensoren mit großem Sichtfeld wie z. B. Laserscanner konzipiert und wird meist in Außenumgebungen eingesetzt (Magnusson u. a., 2007).

Obwohl es viele Veröffentlichungen zu und Varianten für die Registrierung gibt, sind die Kernprobleme noch nicht gelöst. Zum einen lastet allen ICP-Varianten, auch den merkmalsbasierten, der Makel an, dass sie nicht garantiert im richtigen Minimum konvergieren. Zum anderen hängt die Qualität der Registrierung extrem von den verwendeten Merkmalen ab, deren Robustheit wiederum szenenbedingt skaliert. Zudem ist die Korrespondenzfindung in ungeordneten Punktwolken bei den meisten Verfahren ein rechenintensiver Prozess. Weiterhin wurden viele Varianten für schwenkbare Laserscanner entwickelt, die durch das große Sichtfeld eine bessere Korrespondenzfindung ermöglichen. Für 3-D-Tiefenbildkameras ergibt sich hingegen durch die eingeschränkte Sicht, aber hohe Bildrate die Forderung nach größerer Robustheit bei kürzerer Rechenzeit.

4.3.7 Segmentierung

Segmentierung beinhaltet zum einen die Unterteilung einer Punktwolke in sinnvoll zusammengehörige Teile und zum anderen die geometrische Beschreibung der Segmente. Segmentierung kann sowohl auf Farbbildern als auch auf den Punktwolken selbst durchgeführt werden. Generell kann zwischen kanten- und regionsbasierter Segmentierung unterschieden werden, wobei auch die Kombination beider Methoden möglich ist.

Kantenbasierte Segmentierung beruht darauf, dass durchgehende Kanten (Konturen) in den Sensordaten gefunden werden. In einem nachfolgenden Schritt werden dann alle Punkte, die innerhalb einer Kontur liegen, gruppiert. Wird die Kantendetektion auf Farb- oder Intensitätsbildern durchgeführt, können beispielsweise Sobel- oder LaPlace-Operatoren eingesetzt werden. Beide definieren einen Kernel, mit dessen Hilfe der Gradient der Bildintensität berechnet werden kann. Kanten haben dabei typischerweise eine höhere Intensität (Gonzalez u. Woods, 2002).

Eine mehrstufige Methode, die auf die bereits erwähnten zurückgreift, ist *Canny*. Diese liefert bereits zusammenhängende Kantenstücke und nicht nur Kantenpunkte (Canny, 1986).

Um Kanten in Punktwolken zu finden, können z. B. Punktnormalen und Distanzsprünge zwischen Punkten ausgewertet werden (Rusu u. a., 2007). In (Georgiev u. a., 2011) wird hingegen die Kantenextraktion durch Zerlegung der Punktwolke in 2-D-Scans beschrieben (*scan-line*). Eine Ableitung von Kantenpunkten über die lokale Krümmung einer Punktwolke findet sich in (Yang u. Lee, 1999).

Deutlich komplexer ist das Problem, aus den Kanten geschlossene Konturen zu erzeugen. In (Suzuki, 1985) wird eine Methode für die topologische Analyse von Kantenbildern vorgestellt. Damit kann das Bild in einen Konturgraph überführt werden. Hsiao u. a. nutzen morphologische Operatoren, um Lücken in den Kanten zu schließen (Hsiao u. a., 2005). Ebenfalls graph-basierte Strategien werden in (Sappa u. Devy, 2001) und (Bultheel u. Cools, 2008) verwendet, um Kantenpunkte zu verbinden. Sowohl Georgiev u. a. in (Georgiev u. a., 2011) als auch Yang und Lee in (Yang u. Lee, 1999) zeigen die Verknüpfung von Linienstücken über Nachbarschaftsrelationen.

Bei der regionsbasierten Segmentierung ist die Herangehensweise im Vergleich zur kantenbasierten Segmentierung invers. Anstatt geschlossene Konturen als Abgrenzung zu finden, wird versucht, homogene Gebiete in einer Punktwolke zu identifizieren und diese zu beschreiben. Kanten können jedoch als Homogenitätskriterium zum Einsatz kommen. Häufig werden parametrische Modelle in die Sensordaten gefittet (*top-down*). Hierbei stellt der RANSAC-Algorithmus den am weitesten verbreiteten Ansatz dar (Rusu u. a., 2009a; Biswas u. Veloso, 2012; Nüchter u. Hertzberg, 2008; Schnabel u. a., 2007). Bei dieser Herangehensweise wird ein parametrisches Ebenenmodell so in die Punktwolke eingepasst, dass das Verhältnis zwischen zum Modell passenden Punkten und Ausreißern möglichst groß wird. Andere Ansätze benutzen Farbinformation (Taylor u. Cowley, 2011), *Hough Transformation* (Borrmann u. a., 2011) oder Punktnormalen (Holz u. a., 2012).

Als Alternative zum Ebenen-Fitting gibt es viele Arbeiten zu *Region Growing* (RG), auch als *bottom-up* Methode bezeichnet. Hierbei wird ein Ähnlichkeitskriterium definiert. Danach wachsen die Segmente so weit, bis das Kriterium nicht mehr erfüllt ist. In (Rabbani u. a., 2006) und (Zhan u. a., 2010) wird RG anhand von Normalenvektoren gezeigt. Zhan verwendet zusätzlich noch Farbinformation. Rusu u. a. hingegen zeigen RG mit Kantenpunkten als Kriterium (Rusu, 2009). Eine ähnliche Vorgehensweise ist in (Hsiao u. a., 2005) zu finden. Koster und Spann zeigen in (Koster u. Spann, 2000) einen Ansatz, bei dem in einem Vorverarbeitungsschritt Modellparameter geschätzt und danach möglichst viele zu diesem Modell passende Punkte identifiziert werden. Ein RG-Verfahren, das nur für Ebenen funktioniert, aber sehr effizient arbeitet, wird in (Poppinga u. a., 2008) vorgestellt. Ein 2012 in der *Point Cloud Library* (Rusu u. Cousins, 2011) veröffentlichtes Verfahren ist *Multi Plane Segmentation* (MPS). Dieses rechnet durch die Verwendung von *connected components* (Samet u. Tamminen, 1988) sowohl genau als auch effizient. Prinzipiell sind die RG-Varianten genereller einsetzbar als die anderen Ansätze, da theoretisch beliebig geformte Segmente beschrieben werden können.

Kantenbasierte Verfahren haben gemeinsam, dass sie anfällig gegenüber Rauschen sind. Das liegt daran, dass Sensordaten in der Nähe von Kanten größeren Störungen unterworfen sind als solche auf Flächen. Weiterhin müssen aus den Kanten geschlossene Konturen erzeugt werden. Diese beiden Aspekte resultieren in einer erhöhten Rechenzeit gegenüber regionsbasierten

Verfahren und damit eingeschränkter Echtzeitfähigkeit. Aus diesem Grund werden im weiteren Verlauf dieser Arbeit nur noch regionsbasierte Verfahren betrachtet. Aber auch diese stellen bisher keine optimale Lösung des Segmentierungsproblems dar. RANSAC-basierte Methoden sind beispielsweise rechenaufwändig, können die Vorteile geordneter Punktwolken nicht ausnutzen und sind nicht modelltreu, da sie immer das beste Modell finden, aber nicht überprüfen, ob die entsprechende Form überhaupt in der Punktwolke vorkommt. RG-basierte Verfahren neigen zur Über- oder Untersegmentierung. Dies kommt daher, dass es schwierig ist, ein für alle Sensordaten und Szenen gleichermaßen diskriminatives und rauschrobustes Ähnlichkeitskriterium zu finden. Weiterhin beschränken sich die recheneffizienten Verfahren weitgehend auf die Segmentierung planarer Segmente und lassen komplexere Formen außer Acht.

4.3.8 Oberflächenrekonstruktion

Die Oberflächenrekonstruktion ist ein weiterer Schritt, um zum einen die geometrische Beschreibung von Segmenten zu verfeinern und zum anderen Daten zu reduzieren. Ein geläufiger Ansatz ist das Erzeugen von Netzen (*Meshing*). Ein Netz besteht meistens aus einfachen konvexen Polygonen (meist Dreiecke), die über gemeinsame Kanten verfügen und so die Oberfläche eines Objekts beschreiben können. Um aus einer Punktwolke ein Netz zu erzeugen, wird häufig *Delauney Triangulation* eingesetzt (Lee u. Schachter, 1980). Netze werden in (Barber u. a., 1996) und (Bernardini u. Bajaj, 1997) verwendet, um konvexe oder konkave Hüllen für Punktwolken zu berechnen. Rusu zeigt in (Rusu u. a., 2009a), wie Netze verwendet werden können, um Haushaltsobjekte zu beschreiben und zu erkennen. Die Verwendung von Triangulation zur Erzeugung von dichten Oberflächenmodellen wird in (Marton u. a., 2009) präsentiert. Mit dem MLS-Verfahren (vgl. Abschnitt 4.3.1) ist es möglich, kleine Löcher in der Oberfläche zu schließen. Die Berechnung von konvexen Hüllen für Ebenen wird in (Poppinga u. a., 2008; Biswas u. Veloso, 2012) gezeigt, was zu Ebenenpolygonen führt. Wichtig für die Weiterverarbeitung von Ebenenpolygonen (*Merging*) sind Polygonschnittverfahren (*Polygon Clipping*). Hier gibt es mit (Vatti, 1992; Holwerda, 1998; Leonov, 1998; Zalik u. a., 1998) eine Reihe ähnlicher Methoden, die mit konvexen und konkaven Polygonen umgehen können. Manche (z. B. (Vatti, 1992)) sind darüber hinaus auch in der Lage, Polygone mit Löchern zu verarbeiten. Besonders bei der Verarbeitung von Objekten in einer Geometriekarte sind diese Verfahren relevant.

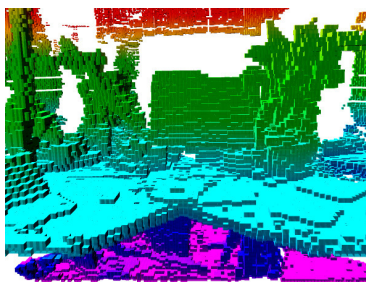
4.4 Kartenrepräsentationen

Für die Speicherung von Kartierungsdaten gibt es in der Literatur mehrere Ansätze. Diese sind häufig anwendungsmotiviert. Generell kann zwischen metrischen und topologischen Karten unterschieden werden. Metrische Karten bilden Regionen und Objekte im Raum durch präzise Koordinaten ab. Topologische Karten speichern Relationen zwischen Objekten, häufig als Graph.

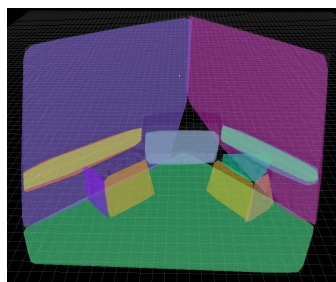
Bei den metrischen Karten gibt es wiederum verschiedene Varianten. Am einfachsten ist sicherlich die Speicherung in Sensorrohdaten, also als Punktwolke, da hierfür keine weitergehende Verarbeitung stattfinden muss. Verwendung finden solche Karten bei SLAM (Nüchter u. a., 2007b; May u. a., 2009; Cole u. Newman, 2006) aber auch für die Szenenrekonstruktion (Rusu, 2009; Newcombe u. a., 2011). Typischerweise werden die einzelnen Punktwolken einfach aufaddiert.

Zusätzliche Schritte können Re- oder Downsampling sein, um den Speicherverbrauch zu optimieren.

Einen anderen Ansatz für metrische Repräsentationen stellen Gitterkarten dar. Ursprünglich wurden diese für die 2-D-Kartierung mit Ultraschallsensoren entwickelt (Moravec, 1988). Die zu kartierende Region wird dabei in regelmäßige Gitterzellen unterteilt. Diese besitzen eine Belegungswahrscheinlichkeit, die durch Sensormessungen beeinflusst wird. Durch diese Vorgehensweise ist es möglich, Freiraum darzustellen und über *Raycasting* Objekte aus der Karte zu löschen. Allerdings haben Gitterkarten durch die Darstellung des Freiraums ohne weitere Optimierung einen hohen durchschnittlichen Speicherbedarf. In (Herbert u. a., 1989; Triebel u. a., 2006; Nagatani u. a., 2009) werden die Karten durch Höheninformationen ergänzt, so dass 2,5-D-Repräsentationen entstehen. Jede Gitterzelle enthält zusätzlich noch einen oder mehrere Höhenwerte, um das Terrain darzustellen. Echte 3-D-Gitterkarten werden in (Meagher, 1982; Fairfield u. a., 2007) eingeführt. Der Raum wird dabei durch einen *Octree* repräsentiert, der den Raum in kubische Volumina (Voxel) unterteilt, welche die Knoten des Baums darstellen (Abbildung 4.8a). Der Baum ist hierarchisch aufgebaut, wodurch ein Volumenelement solange in acht Subelemente aufgeteilt wird, bis die gewünschte räumliche Auflösung erreicht ist (Wurm u. a., 2010). Eine Erweiterung zeigen Wurm u. a. in (Wurm u. a., 2011). Der Raum wird durch einen Baum dargestellt, wobei jeder Knoten eine *Octomap* enthält. Die Transformation zwischen den Knoten wird mitgespeichert, um eine metrische Gesamtdarstellung zu ermöglichen.



(a) Voxelkarte (Wurm u. a., 2010)



(b) Ebenenkarte (Biswas u. Veloso, 2012)



(c) SURFEL-Karte (Henry u. a., 2012)

Abbildung 4.8: Verschiedene Kartenrepräsentationen

Die Ableitung geometrischer Merkmale und deren Speicherung in einer Karte sind ebenfalls in einigen Publikationen zu finden. Häufig werden die Geometriekarten auch als semantische Karten bezeichnet, da sie aufbereitete Informationen enthalten. Der Begriff ist jedoch nicht klar abgegrenzt. So können semantische Karten nur aus Ebenen bestehen (Nüchter u. Hertzberg, 2008) oder detaillierte Information über Objekte im Raum enthalten (Pronobis, 2011). Viele 2-D-SLAM-Verfahren benutzen Merkmale wie Linien zum Aufbau einer Karte (Thrun u. a., 2005). Auch im 3-D-SLAM werden Merkmalskarten erzeugt, die jedoch nur zur Lokalisierung des Roboters verwendet werden, nicht zur geometrischen Interpretation (Engelhard u. a., 2011; Strasdat u. a., 2010). Rusu ersetzt in (Rusu, 2009) Teile der Punktekarte durch Basisgeometrien (Boxen). Diese werden jedoch nicht in einer separaten Karte gespeichert. In (Nüchter u. Hertzberg, 2008;

Biswas u. Veloso, 2012) wird eine Ebenenkarte der Umgebung erzeugt (Abbildung 4.8b). Während Nüchter sich auf die Ableitung der Ebenen aus Punktwolken fokussiert, setzt sich Biswas auch mit dem Management geometrischer Karten auseinander. Pfister führt in (Pfister u. a., 2000) kleine Oberflächenelemente (SURFELS) ein. Diese beschreiben eine lokale Oberfläche und deren Textur. Weiterhin sind sie in der Lage, ihre Auflösung automatisch an die verfügbare Informationsdichte anzupassen. Henry u. a. zeigen in (Henry u. a., 2012) den Einsatz von SURFELS in der Kartierung (Abbildung 4.8c). Sie sind dadurch in der Lage, die Daten um den Faktor 32 zu komprimieren. Allerdings wird die SURFEL-Karte in einem offline Schritt nach der Kartierung erzeugt.

Im Gegensatz zu metrischen Karten bilden topologische Karten Relationen von Objekten und Regionen im Raum ab. Dies ist vor allem dann sinnvoll, wenn ein Roboter in weitläufigen Umgebungen handeln soll. In (Nüchter, 2009) wird der Einsatz einer graph-basierten topologischen Karte für 3-D-SLAM gezeigt. Die einzelnen Sensoraufnahmen werden über die Aufnahmepositionen verknüpft, so dass im Fall des Kreisschluss eine globale Optimierung stattfinden kann. Ein ähnlicher Ansatz wird in (Grisetti u. a., 2010) gezeigt. Pronobis zeigt in (Pronobis, 2011) die semantische Klassifikation von Räumen und Objekten. Diese werden dann in eine topologische Karte integriert, um dem Roboter eine semantische Planung zu ermöglichen.

Während Aufbau und Speicherung von Punktekarten weitgehend gelöst sind, zeigt der Stand der Technik bei geometrischen Karten Lücken. Zum einen werden häufig nur Ebenen kartiert und dadurch keine allgemeingültige geometrische Repräsentation erzeugt, zum anderen spielen die Möglichkeiten der Benutzerinteraktion eine untergeordnete Rolle.

4.5 Übertragung von 3-D-Daten über Netzwerke

Die Übertragung von 3-D-Daten über Netzwerke ist bei teil-autonomen Anwendungen von großer Wichtigkeit, da ständig Daten zur Bedienerseite transferiert werden müssen. Der Stand der Technik weist hier zwei Vorgehensweisen auf, zum einen die Kompression von Punkt- und Meshdaten vor der Übertragung, häufig verbunden mit inkrementellem Transport und zum anderen die Wahl einer komprimierenden Datenrepräsentation für die Karten.

Ein Forschungsgebiet ist die hybride TCP/UDP-Übertragung für das Streaming (Chen u. a., 2003; Tari u. a., 2005; Tian u. AlRegib, 2006). Meist werden wichtige Strukturdaten über das verlässliche TCP gesendet, wohingegen für voluminöse Daten UDP verwendet wird. Andere Ansätze wie (Feisst u. Christ, 2004) nutzen dynamische Kompression von 3-D-Daten je nach Bandbreite. In (Gumhold u. a., 2005) wird die Kompression von Punktwolken durch eine hierarchische Baumstruktur beschrieben, was jedoch sehr langsam ist. Kaess u. a. zeigen in (Kaess u. a., 2003) die echtzeitfähige Kompression von Laserscans mit Hilfe von Graustufenbildern und verlustfreier Bildkompressionsverfahren. In (Waschbüsch u. a., 2004) wird die Datenreduktion bei Punktwolken in mehreren Auflösungsstufen vorgestellt. Yang u. a. nutzen in (Yang u. a., 2004) zusätzlich Informationen über den Blickpunkt des Bedieners, um mehr oder weniger stark zu komprimieren. Die inkrementelle Übertragung von 3-D-Strukturen wird in (Li u. a., 2003; Song u. Lee, 2004) gezeigt. Dabei werden je ein komplettes Modell auf beiden Seiten vorgehalten und nur die Änderungen übertragen.

Durch die Wahl einer speichereffizienten Repräsentation für die Karte kann bereits eine entscheidende Reduktion des Datenvolumens erfolgen, was dann eine gesonderte Behandlung der Übertragung überflüssig macht. Nagatani u. a. zeigen beispielsweise in (Nagatani u. a., 2010) die Verwendung von DEM als Kartenformat. Durch Anpassung der Gitterauflösung können die 3-D-Daten schon sehr stark komprimiert werden, allerdings unter Verlust von Detailinformation. Soll die Punktrepräsentation beibehalten werden, kann Downsampling interessant sein. Über einen Voxelfilter (Rusu u. Cousins, 2011) kann die zu übertragende Punktwolke auf die gewünschte Auflösung heruntergerechnet werden. Weiterhin können auch *Octrees* verwendet werden, um Punktdaten zu komprimieren. In (Wurm u. a., 2010) wird gezeigt, dass der Kompressionsgrad mit zunehmender Kartengröße ansteigt. Allerdings beansprucht die Verwendung von *Octrees* viel Rechenzeit.

4.6 Ansätze zum teil-autonomen Betrieb

Da die vorliegende Arbeit den teil-autonomen Betrieb von Robotern im Fokus hat, sollen in diesem Abschnitt entsprechende Ansätze untersucht werden. Dabei soll vor allem festgestellt werden, wie ein System zur Umgebungserfassung in einen teil-autonomen Betriebsmodus eingebettet werden kann und wie eine optimale Kopplung zwischen Bediener und Roboter aussieht. Dabei kann der Grad der Autonomie von fast reiner Teleoperation (starke Kopplung) bis hin zu sehr großer Autonomie mit geringem Bedienereingriff reichen (schwache Kopplung) (Aracil u. a., 2007).

Starke Kopplung liegt beispielsweise in dem Fall vor, in dem eine *Master-Slave-Architektur* für eine Armsteuerung in fließendem Übergang mit einem autonomen Regler überlagert werden kann (Malysz u. Sirouspour, 2011). Einen ähnlichen Ansatz zeigen (Chipalkatty u. a., 2011). Hier wird die manuelle Fortbewegung eines teleoperierten vierbeinigen Laufroboters durch einen modell-prädiktiven Regler unterstützt, um die Stabilität der Bewegungen zu erhöhen. Im Bereich der Manipulation schlagen Balaguer u. a. in (Balaguer u. a., 2007) vor, den Einfluss der Slave-Loop je nach Fähigkeiten des Bedieners anzupassen und damit den Autonomiegrad zu steuern. Ein Beispiel für starke Kopplung in der Wahrnehmung ist in (Carff u. a., 2009) zu finden. Der menschliche Bediener kann im Kamerabild eines Roboters Hindernisse einzeichnen oder Oberflächen klassifizieren. Eine Vorverarbeitung der Sensordaten findet in diesem Fall nicht statt.

Wird der Autonomiegrad erhöht, liegt eine schwächere Kopplung von Bediener und Roboter vor. Cherubini u. a. beschreiben ein dreistufiges Navigationskonzept (Cherubini u. a., 2007). Im *Single step* Modus reagiert der Roboter direkt auf die Eingaben des Benutzers mit Bewegungen. Im teil-autonomen Modus gibt der Bediener nur noch die Richtung vor, dazwischen fährt der Roboter autonom. Im autonomen Modus schließlich wird nur noch der Zielpunkt vorgegeben, zu dem der Roboter dann selbständig navigiert. Ähnliche Ansätze werden in (Coltin u. a., 2012; Nagatani u. a., 2010; Iocchi u. Pellegrini, 2007) für unterschiedliche Anwendungsfälle vorgestellt.

Im Bereich der Wahrnehmung gibt es deutlich weniger Arbeiten zur Benutzerintegration. Johnson-Roberson u. a. stellen in (Johnson-Roberson u. a., 2011) ein System zur benutzerunterstützten Szeneninterpretation vor. Der Bediener kann dabei eine vorsegmentierte Szene über Sprachkommandos optimieren. Ein ähnlicher Ansatz wird in (Pitzer u. a., 2011) beschrieben. Um die Objekterkennung zu verbessern, kann der Bediener in der Szene eine Region markieren, die das

Objekt enthält. Dieses wird dann autonom segmentiert. Eine ganzheitliche Herangehensweise für Navigation, Umgebungserfassung und Manipulation zeigen Goodfellow u. a. in (Goodfellow u. a., 2010). Der Roboter führt dabei Einzelschritte autonom aus und präsentiert die jeweiligen Ergebnisse dem Bediener. Dieser kann dann entscheiden, ob und wie die nächste Aktion ausgeführt werden soll. Ein anderer Ansatz beschreibt das teil-autonome Greifen von Objekten mit einem Roboterarm unter Verwendung von Kameras (Notheis u. a., 2010). Auch hier wird die Gesamtaufgabe in einzelne Aktionen unterteilt, die teils autonom, teils teleoperiert ausgeführt werden.

Eine weiterführende Methode zur Anbindung des Bedieners, um insbesondere unvorhergesehene Situationen zu meistern, wird in (Qiu u. a., 2012) beschrieben. Die vom Roboter zu lösende Aufgabe wird dabei in elementare Einzelschritte zerlegt, die der Roboter autonom durchläuft. Zwischen jedem Einzelschritt besteht die Möglichkeit des Bedieneringriffs basierend auf den Ergebnissen der Aktion.

Für die vorliegende Arbeit wird ebenfalls eine eher schwache Kopplung von Mensch und Roboter angestrebt. Der Roboter führt Einzelschritte autonom durch und gibt Feedback über die Ergebnisse oder Probleme. Der Bediener kann dann durch manuellen Eingriff reagieren. Für das zu entwickelnde System bedeutet dies, dass eine Aktualisierung des Umgebungsmodells als elementarer Einzelschritt betrachtet wird. Dabei wird eine (ggf. vom Bediener vorgegebene) Region der Umgebung in einem Schritt kartiert. Danach kann der Bediener in der Visualisierung das Ergebnis betrachten und mit dem Modell interagieren, in dem er z. B. die nächste Aktion einleitet. Für Navigationsaufgaben ist auch der Fall relevant, dass während der Roboterbewegung ein kontinuierlicher Kartenaufbau stattfindet. In diesem Fall liegt eine stärkere Kopplung vor, da das Kartierungsergebnis direkt von den Bewegungsbefehlen des Bedieners abhängt. Die Anforderungen an das zu entwickelnde Verfahren zur 3-D-Umgebungserfassung sind so ausgelegt, dass beide Modi zuverlässig betrieben werden können.

4.7 Folgerungen aus dem Stand der Technik

Um konkrete Folgerungen für das in dieser Arbeit zu entwickelnde System zur Umgebungserfassung abzuleiten, soll der Stand der Technik in diesem Abschnitt zunächst kritisch bewertet werden. Details wurden dabei schon in den vorigen Abschnitten genannt, hier sollen die wichtigsten Punkte zusammengefasst werden. Gleichzeitig wird überprüft, inwieweit die Anforderungen aus Kapitel 3 bereits durch den Stand der Technik erfüllt werden und welche durch Neuentwicklungen im Rahmen dieser Arbeit abgedeckt werden müssen.

Die Betrachtung der 3-D-Tiefenkameras lässt erkennen, dass diese durchaus leistungsfähig und für die Anforderungen dieser Arbeit geeignet sind. Allerdings muss besonders bei TOF-Kameras eine Filterung der Sensordaten stattfinden. In der Literatur gibt es einige Ansätze für Rauschfilter, die jedoch teils zu langsam, teils unzureichend hinsichtlich der Parametereinstellungen evaluiert sind. Weiterhin ist unklar, welche Filter wie in einer Filterkaskade zusammengeschaltet werden können. Diese Probleme und Fragestellungen können durch eine gründliche Evaluierung bestehender Verfahren und neue Filterentwicklungen gelöst und beantwortet werden.

Die Analyse von Arbeiten zur 3-D-Umgebungserfassung lässt insbesondere in der autonomen Robotik eine Vielzahl häufig weit fortgeschrittener Lösungen erkennen. Unterschiede liegen in der eingesetzten Sensorik und der Optimierung auf verschiedene Anwendungsfälle. Gemeinsamkeiten sind bei den benötigten Einzelschritten und den resultierenden Kartenrepräsentationen erkennbar. Die meisten Arbeiten vernachlässigen eine Benutzerinteraktion mit dem Umgebungsmodell, da der autonome Betrieb im Vordergrund steht. Teleoperierte Ansätze hingegen weisen einen deutlich niedrigeren Grad an Sensordatenverarbeitung auf. Eine Benutzerinteraktion findet zwar häufig statt, jedoch meist nur auf aktuellen Sensordaten und nicht auf einem auch zurückliegende Information enthaltenden Umgebungsmodell.

Bei den Einzeltechnologien für die Umgebungserfassung zeigt sich ein ähnlich vielfältiges Bild wie bei den Gesamtsystemen. Sowohl bei der Nachbarschaftssuche als auch bei der Normalenberechnung kann durch konsequente Ausnutzung der geordneten Struktur von Tiefenbildern eine deutliche Effizienzsteigerung erzielt werden. Bei den Merkmalen gibt es besonders viele unterschiedliche Verfahren. Fast alle beanspruchen für sich, eine optimale Lösung des Problems bereitzuhalten. Vergleichende Evaluierungen gibt es jedoch nicht. Häufig verwendet und ohne Ausschlusskriterien sind PC, FPFH und RSD. Diese sollen im Rahmen dieser Arbeit genauer untersucht werden. Ähnliches gilt für die Segmentierung, wo es mit dem Einpassen von parametrischen Modellen und *Region Growing* zwei für diese Arbeit relevante Ansätze gibt. Häufig ist ein Fokus auf der Segmentierung von Ebenen zu erkennen, komplexere Formen werden meist nicht betrachtet. Weiterhin ist bei RG die Über- und Untersegmentierung von Szenen ein ungelöstes Problem. Verfahren, die parametrische Modelle einsetzen, sind prinzipbedingt nicht modelltreu, was besonders in komplexen Szenen falsche Ergebnisse produzieren kann. Diese Tatsachen zeigen deutlich den Bedarf für neue Ansätze. Bei der Registrierung wird ICP in allen seinen Varianten am häufigsten eingesetzt. Anders als bei den vorhergegangenen Komponenten liegen hier jedoch gute Vergleiche vor. Allen Varianten sind jedoch die Schwächen gemeinsam, dass das Verfahren in lokalen Minima konvergiert und viel Rechenzeit beansprucht. Hier ist noch deutlicher Entwicklungsbedarf zu sehen.

Für alle Einzelschritte gilt, dass sie maßgeblich Auswirkung auf die Erfüllung der Anforderungen nach Genauigkeit (A2, A10), Aktualisierungsgeschwindigkeit (A3, A6) und Robustheit (A12, A13) haben. Ob und welche Verfahren die Anforderungen erfüllen, muss im Rahmen dieser Arbeit evaluiert und die Einhaltung ggf. durch Neuentwicklungen sichergestellt werden.

Bei den Kartenrepräsentationen gibt es zumindest für Punktekarten (A1) mit der Darstellung als Punktwolke oder als 3-D-Gitter zwei funktionale und robuste Lösungen, von denen das 3-D-Gitter auch dynamische Änderungen erfassen kann (A4). Für die Behandlung von Regelgeometrien in einer Karte gibt es Ansätze, die Ebenen betrachten. Dies stellt aber keine allgemeingültige Lösung dar. Damit ist die Anforderung A9 nur teilweise erfüllt. Die Anforderungen A5, A8 und A11 können durch entsprechende Darstellung und Speicherung der geometrischen Information erreicht werden. Wie diese konzipiert sein müssen, ist durch den Stand der Technik nicht eindeutig ersichtlich, da geometrische Repräsentationen hauptsächlich in der autonomen Robotik verwendet werden und teleoperierte Ansätze zumeist auf die Erstellung eines Umgebungsmodells verzichten.

Bei der Übertragung von 3-D-Daten über Netzwerke gibt es viele Arbeiten zum Streaming und zur Kompression von Punktwolken. Für diese Arbeit besser geeignete Ansätze lösen das Problem

über die Wahl einer passenden Kartenrepräsentation. Dadurch kann Anforderung A7 abgedeckt werden.

Die Einbettung der Umgebungserfassung in einen teil-autonomen Betriebsmodus wird in dieser Arbeit über eine schwache Kopplung von Roboter und Bediener realisiert. Dies hat den Vorteil, dass die Kartenaktualisierung als elementarer Einzelschritt betrachtet werden kann. Dadurch wird auch die Forderung nach Modularität (A14) erleichtert. Diese muss aber zusätzlich durch Unterteilung des Verfahrens in sinnvolle Einzelschritte und die verwendeten Entwicklungs- und Implementierungsmethoden sichergestellt werden.

5 Entwurf des Verfahrens zur 3-D-Umgebungserfassung

Die in Kapitel 3 hergeleiteten Anforderungen werden nun als Basis für den Entwurf eines Verfahrens für die 3-D-Umgebungserfassung verwendet. Die Anforderungen werden dabei im Einzelnen betrachtet und Möglichkeiten erarbeitet, diese zu erfüllen. Dabei können sowohl die Wahl von Algorithmen und Verarbeitungsschritten eine Rolle spielen als auch deren Reihenfolge in der Verarbeitungskette. Eine Herausforderung stellt die Vielzahl der bereits verfügbaren Verfahren für Einzelschritte in der Kartierung dar. Deswegen kann die Konzeption in diesem Kapitel nur bis zu einem gewissen Detailgrad erfolgen.

5.1 Anwendungsszenario und Zielrobotersystem

Da in dieser Arbeit hauptsächlich Daten von 3-D-Tiefenbildkameras verwendet werden, liegt der Schwerpunkt bei den Anwendungen auf solchen, die größtenteils in Innenraumumgebungen ablaufen. Damit kommen nach Tabelle 2.1 die Erfüllung von Haushaltsaufgaben oder Anwendungen im gewerblichen Umfeld in Betracht. Durch die eingeschränkte Reichweite von 3-D-Tiefenbildkameras und die Weitläufigkeit vieler gewerblicher Umgebungen sind diese weniger gut für die Erfassung mit solchen Sensoren geeignet als Haushaltsumgebungen. Weiterhin sind Haushaltsumgebungen durch den allgemein geringeren Ordnungsgrad anspruchsvoller, was die Kartierung angeht. Deswegen liegt der Fokus in dieser Arbeit auf der Erfassung von Szenen im Haushalt (siehe Abbildung 5.1c), ohne gewerblich geprägte Umgebungen als Anwendungsfeld auszuschließen.

Als Zielrobotersystem wird der vom Fraunhofer IPA entwickelte Care-O-bot[®] 3 (Abbildung 5.1a) gewählt, da er die benötigte Sensorausstattung besitzt und für die Erfüllung von Haushaltsaufgaben konzipiert wurde. Der in Abbildung 5.1b gezeigte Sensorkopf des Roboters ist auf einer Schwenk-Neige-Einheit montiert und kann somit durch seine Bewegung des Kopfes einen großen Sichtbereich abdecken. Auf dem Sensorkopf befinden sich ein Stereokamerasystem und eine 3-D-Tiefenbildkamera vom Typ *Microsoft Kinect*, welche die Eingangsdaten für das zu entwickelnde Verfahren liefert. Die an der Basis angebrachten Laserscanner werden zur Lokalisierung benutzt, so dass bereits eine relativ genaue Positionsschätzung zur Verfügung steht, was den Aufbau eines konsistenten Umgebungsmodells unterstützt.

5.2 Konzeption des Verfahrens zur 3-D-Umgebungserfassung

Der Entwurf des Verfahrens kann aufgeteilt werden in die Konzeption der Repräsentation für das Umgebungsmodell und den Entwurf der Verarbeitungsschritte, um diese Repräsentation zu erstellen.

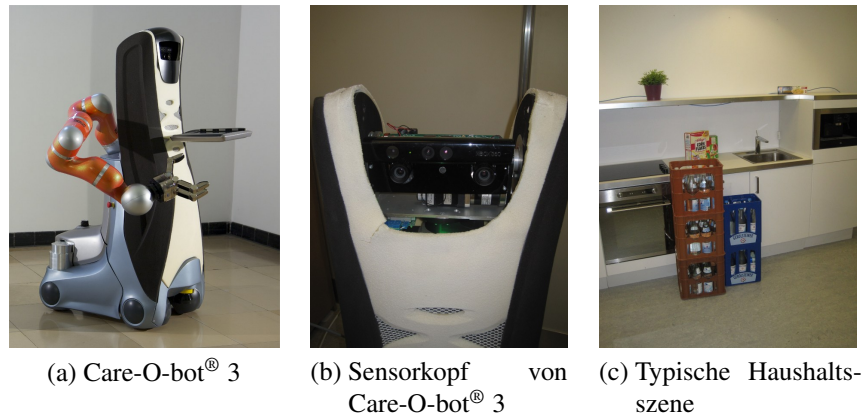


Abbildung 5.1: Zielsystem und Anwendungsszenario

5.2.1 Entwurf der Repräsentation für das Umgebungsmodell

Die Repräsentation des Umgebungsmodells stellt einen zentralen Aspekt des zu entwickelnden Verfahrens dar. Einerseits können nur durch sie einige der Anforderungen erfüllt werden, andererseits bestimmt sie auch die benötigten Verarbeitungsschritte zur Erzeugung der Repräsentation (Abbildung 5.2). A1 und A9 fordern sowohl eine Punktekarte (PK) als auch eine Geometriekarte (GK) des Umgebungsmodells. Aus dem Stand der Technik geht hervor, dass bei den Punktekarten nur die Darstellung als Punktwolke und das 3-D-Gitter in Frage kommen und für die GK die Darstellung durch Regelgeometrien. Ein Vergleich der Repräsentationen ist in Tabelle 5.1 dargestellt. Ein (✓) bedeutet dabei, dass das Kriterium mit Einschränkungen erfüllbar ist. Die Regelgeometriekarte ist der einzige Typ, der sowohl Punkte als auch Geometrien darstellen kann. Allerdings ist es wenig effizient (sowohl vom Speicherverbrauch als auch von der Zugriffsgeschwindigkeit), Punkte als Geometrien zu behandeln. Sowohl 3-D-Gitter als auch Regelgeometriekarte können den vollen Informationsgehalt beinhalten, verlieren dadurch jedoch ihre Vorteile bei der Speichereffizienz, den sie normalerweise durch sinnvolle Reduktion der Daten erhalten (A7). Die Punktwolke hingegen hat immer vollen Informationsgehalt bei hohem Speicherverbrauch. Schließlich ist bei der Punktwolke kein dynamisches Update (A4) möglich, da keine Daten gelöscht werden können. Das 3-D-Gitter wiederum ist kein durch den Menschen lesbares Format und lässt sich nicht einfach visualisieren (A5). Hier bietet die Regelgeometriekarte die besten Voraussetzungen. Einfache Geometrien lassen sich durch den Menschen einfach deuten (A5) und sind auch mit einfachen Mitteln visualisierbar. Die Punktwolke lässt sich zwar einfach darstellen. Insbesondere ungeübte Personen tun sich jedoch schwer, sie richtig zu interpretieren.

Dies lässt den Schluss zu, dass keine der drei Repräsentationen für sich allein als Umgebungsmodell geeignet ist. Aus diesem Grund wird in dieser Arbeit eine hybride Repräsentation aus 3-D-Gitter und Regelgeometrien gewählt, da sich nur so alle Anforderungen erfüllen lassen und die Punktwolke gegenüber dem 3-D-Gitter deutliche Nachteile besitzt. Dass sich das 3-D-Gitter schlecht visualisieren lässt, kann durch eine temporäre Umwandlung in eine Punktwolke behoben werden. Durch das kompakte Datenvolumen der GK ist es auch ohne Probleme möglich,

Anforderung	Punktwolke	3-D-Gitter	Regelgeometrien
Punkte abbildbar (A1)	✓	✓	(✓)
Geometrien abbildbar (A9)	✗	✗	✓
voller Informationsgehalt (A10)	✓	(✓)	(✓)
speichereffizient (A7)	✗	✓	✓
dynamisches Update (A4)	✗	✓	✓
Visualisierbarkeit (A5)	(✓)	✗	✓

✗ nicht erfüllt (✓) teilweise erfüllt ✓ erfüllt

Tabelle 5.1: Vergleich der Kartenrepräsentationen

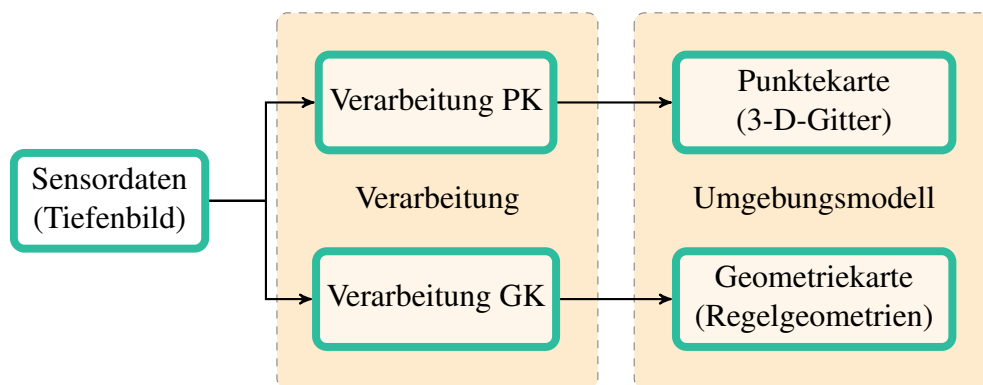


Abbildung 5.2: Grundlegendes Konzept für das Verfahren zur 3-D-Umgebungserfassung

diese über ein Netzwerk zu übertragen. Sollte ausreichend Bandbreite zur Verfügung stehen, kann zusätzlich auch die Punktekarte übertragen werden, da diese im Normalfall zusätzliche Information enthält und der Bediener so die Konsistenz beider Karten überprüfen kann. Auch für die Ableitung semantischer Information (A11) stellt die Darstellung durch Regelgeometrien eine gute Grundlage dar, da sie bereits verarbeitete Sensordaten mit höherwertiger Information vorhält.

Allerdings stellen bereits existierende Ansätze für geometrische Karten zumeist nur Ebenen dar, was der Forderung nach Vollständigkeit in A10 widerspricht. Um auch andere Regelgeometrien wie Zylinder, Kugeln oder gekrümmte Oberflächen abbilden zu können, muss eine geeignete Beschreibungsform für die Geometrietypen definiert werden. Da es sich bei allen Regelgeometrien um parametrische Modelle handelt, können grundlegende geometrische Informationen gut durch die Parameter des jeweiligen Modells beschrieben werden (z. B. eine Ebene durch ihren Normalenvektor und Abstand vom Ursprung). In der Regel ist diese Beschreibung jedoch nicht ausreichend. Beispielsweise haben Ebenen eine unendliche Ausdehnung, real existierende planare Flächen sind aber begrenzt. Aus diesem Grund wird als weitere Information für jeden Karteneintrag eine konkave Hülle, bestehend aus den Konturpunkten der Regelgeometrie abgelegt (vgl. Anhang D). Liegen die Sensordaten als farbige Punkt wolke vor, kann aus allen zur Geometrie gehörigen Punkten der Farbmittelwert gebildet und dieser ebenfalls in der GK gespeichert

Feld	Information
Modellparameter	geometrische Darstellung
Hüllpunkte	geometrische Darstellung
Farbwert	mittlere Farbe der zum Eintrag gehörigen Punkte
ID	Identifikation des Karteneintrags
Gewichtungsfaktor	Einfluss auf Fusion durch historische Information

Tabelle 5.2: In einem Karteneintrag enthaltene Information

werden.

Wenn neue Einträge in die Karte aufgenommen werden, müssen diese mit bestehenden fusioniert werden, falls sie dieselbe Oberfläche beschreiben. Um eine gegenüber fehlerhaften Einträgen robuste GK zu gewährleisten, wird die Historie von Fusionsvorgängen für jeden Karteneintrag als Gewichtungsfaktor abgelegt. Weiterhin müssen alle Karteneinträge eine eindeutige ID besitzen, um eine konsistente Karte zu garantieren und die Interaktion des Bedieners mit dem Umgebungsmodell (A8) sicherzustellen. Ist dies gegeben, können die Objekte durch den Bediener in einem grafischen Interface manipuliert (verschoben, gelöscht, hinzugefügt) werden. Tabelle 5.2 gibt einen Überblick über die in einem Karteneintrag gespeicherte Information.

5.2.2 Verarbeitungsschritte zur Erzeugung der Repräsentation

Nachdem die Art der Repräsentation festgelegt ist, stellt sich die Frage, welche Schritte und Verfahren notwendig sind, um ein solches Umgebungsmodell zu erzeugen. Grundsätzlich werden die Sensordaten durch mehrere Verarbeitungsschritte so aufbereitet, dass sowohl eine Punktekarte als auch eine Geometriekarte entsteht. Im Folgenden werden diese Schritte genauer untersucht und festgelegt.

Um eine PK erzeugen zu können, müssen die Sensordaten in einem fixen Koordinatensystem (Weltkoordinatensystem) registriert werden. Danach erfolgt die Aggregation in einem 3-D-Gitter. Das bedeutet, dass die Genauigkeit der Punktekarte (A2) hauptsächlich von der Genauigkeit des Registrierungsverfahrens abhängt. Eine weitere Aufgabe der Registrierung ist die Kompensation von Ungenauigkeiten in der Lokalisierung und damit die Erhöhung der Robustheit (A12). Die Anfälligkeit gegenüber Messungenauigkeiten (A13) kann zwar auch über eine robuste Registrierung abgemildert werden, besser ist es jedoch, eine Sensordatenvorverarbeitung durchzuführen. Durch entsprechende Filter ist es möglich, die Sensordatenqualität soweit zu erhöhen, dass sie sich nur noch geringfügig negativ auf die Qualität der Punktekarte auswirkt. Allerdings können durch zu starken Filtereinsatz auch korrekte Messwerte aus den Sensordaten entfernt werden.

Die Erzeugung einer GK baut auf den Ausgangsdaten der Registrierung auf. Der Grund dafür ist, dass durch die Registrierung bereits Fehlereinflüsse durch ungenaue Lokalisierung und Fehlmessungen beseitigt werden. Aus den registrierten Punktwolken können Regelgeometrien extrahiert werden. Relevant sind dabei vor allem Ebenen, aber auch gekrümmte Oberflächen wie Zylinder oder Kugeln. Aus dem Stand der Technik geht hervor, dass für eine erfolgreiche Extraktion von

Geometrien vor allem Verfahren zur Extraktion von 3-D-Punktmerkmalen und zur Segmentierung wichtig sind. Auch hier gilt wie bei der Registrierung, dass die ausgewählten Verfahren im Rahmen dieser Arbeit gründlich evaluiert und gegebenenfalls neue Methoden erarbeitet werden müssen. Die Forderung nach Genauigkeit der GK (A10) kann also nur durch entsprechend genaue Verfahren für die Erstellung der GK erfüllt werden. Die Robustheit der Umgebungserfassung gegenüber Fehlereinflüssen (A12, A13) kann über fehlertolerante Verfahren bei der Ableitung von Geometrien und bei der Kartenverwaltung zumindest für die GK weiter verbessert werden.

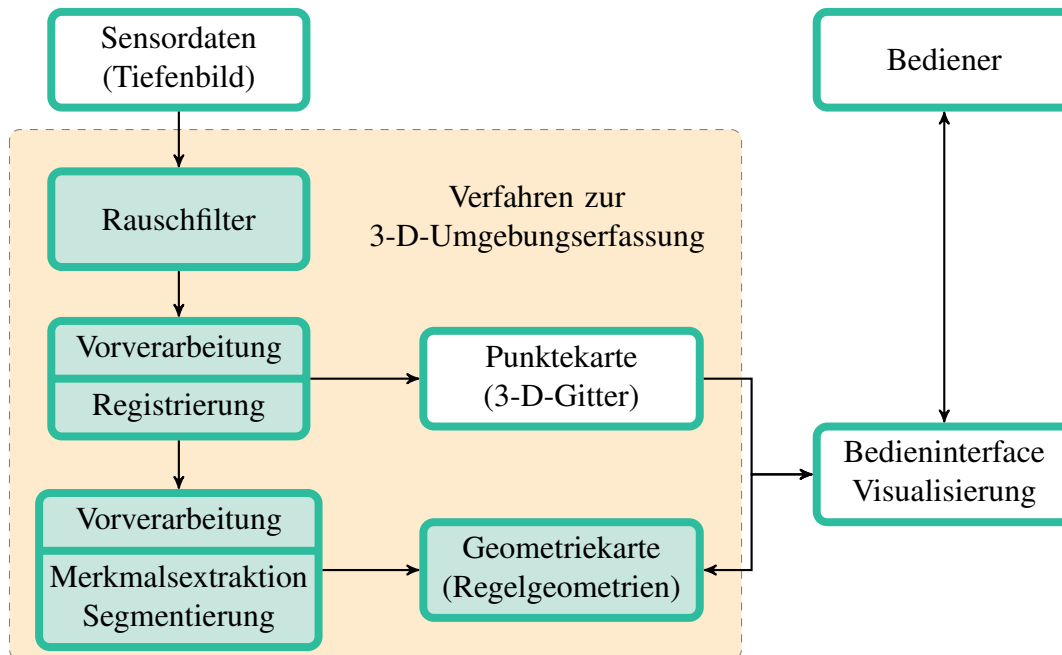


Abbildung 5.3: Detailliertes Konzept für das Verfahren zur 3-D-Umgebungserfassung (zu entwickelnde Komponenten sind eingefärbt)

Die Anforderungen an ausreichende Aktualisierungsgeschwindigkeit gelten einerseits für die PK (A3), andererseits für das Gesamtsystem (A6). Dies bedeutet, dass die Registrierung Einfluss auf beide Anforderungen hat. Generell gilt, dass eine ausreichende Aktualisierungsgeschwindigkeit durch die Auswahl schneller Verfahren für alle Teilschritte der Kartierung erreicht werden kann. Weiterhin kann durch Vorverarbeitung (z. B. durch Downsampling) die Geschwindigkeit nachfolgender Schritte erhöht werden. Mit den bisher erarbeiteten Ergebnissen ergibt sich ein Konzept wie in Abbildung 5.3 dargestellt.

6 Entwurf und Umsetzung der Einzelkomponenten

Wie aus dem Konzept hervorgeht, können einige der Anforderungen nur durch geeignete Wahl oder Entwicklung der Verfahren für die Einzelschritte der Umgebungserfassung erfüllt werden. Dies sind zum einen die bereits in Abbildung 5.3 genannten Schritte der Vorverarbeitung, Registrierung, Merkmalsextraktion/Segmentierung sowie zwei weitere Aspekte, die erst die Voraussetzungen zur Durchführung der anderen Schritte schaffen. Erstens ist es wichtig, in welchem Format die Punktwolken verarbeitet werden. Viele der in Kapitel 4 vorgestellten Algorithmen wurden für ungeordnete Punktwolken entwickelt. Das Ausgabeformat von Tiefenkameras ist jedoch ein Tiefenbild, also eine geordnete Punktwolke. Einen großen Unterschied, insbesondere für die Nachbarschaftssuche, macht diese Tatsache vor allem in Bezug auf die Rechengeschwindigkeit. Da die Nachbarschaftssuche z. B. bei der Segmentierung oder Registrierung eine wichtige Rolle spielt, soll im nächsten Abschnitt zunächst dieser Unterschied herausgearbeitet werden. Zweitens ist die Berechnung von Punktnormalen Voraussetzung einer leistungsfähigen Segmentierung und Merkmalsextraktion, so dass ihr ein eigener Abschnitt gewidmet werden soll. Im letzten Abschnitt wird schließlich die Aggregation einer GK aus den extrahierten Segmenten entwickelt.

Für die vergleichende Evaluierung der Verfahren werden für gewöhnlich sowohl die benötigte Rechenzeit als auch die Genauigkeit der Verfahren verglichen. Implementierungen der Verfahren werden grundsätzlich in C++ durchgeführt. Die Messung der Rechenzeit erfolgt auf einem Rechner mit Intel Core i7-2600 CPU und 16 GB RAM, auf dem Ubuntu Linux 10.10, 64-Bit läuft.

6.1 Nachbarschaftssuche in Punktwolken

Da existierende Verfahren zur Nachbarschaftssuche (ANN) nicht effizient auf Tiefenbildern rechnen, wird in diesem Abschnitt ein neuer Ansatz entwickelt. Auf Bildebene können die Nachbarn eines Punktes durch einfaches Iterieren über die Pixelkoordinaten gefunden werden. Dadurch entfällt die Betrachtung aller Punkte in der Wolke bei jedem Suchvorgang.

Der Ansatz bestimmt Nachbarpunkte in einem einstellbaren Pixelradius anstatt in einem metrischen Bereich zu suchen, was für die meisten Anwendungsfälle ausreichend ist. Ein Vorteil der Pixelsuche ist, dass der metrische Suchradius mit der Entfernung der Punkte vom Betrachter automatisch skaliert, was das Verfahren robuster gegenüber Rauschen macht. Der Berechnungsablauf ist in Algorithmus 1 zu sehen.

Es soll ein Nachbarschaftsbereich \mathcal{P}_k mit dem Pixelradius r um den Abfragepunkt \mathbf{p}_q gefunden werden. Zunächst wird eine Suchmaske M (Abbildung 6.1) über die Punktindizes m definiert. Diese besteht aus n_r quadratischen Rahmen um \mathbf{p}_q , welche den Abstand s_f zueinander haben. Auf jedem Rahmen werden Punkte mit dem Abstand s_p abgetastet. Aus dieser Herangehensweise ergibt sich die Möglichkeit, die Dichte der Nachbarschaftssuche zu bestimmen. Werden

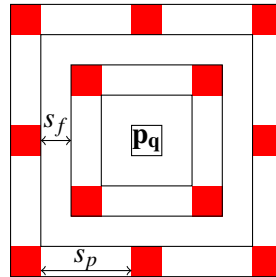
Algorithmus 1 Geordnete Nachbarschaftssuche**Require:** query point \mathbf{p}_q , search radius r , frame step s_f , point step s_p **Ensure:** neighborhood \mathcal{P}_k of \mathbf{p}_q get index i_q of \mathbf{p}_q **procedure** COMPUTESEARCMASK(r, s_f, s_p)**while** $-r \leq ns_f \leq r$ **do**Sample m with s_p in quadratic frame of size ns_f $M \leftarrow \{M, m\}$ $n \leftarrow n + 1$ **end while****return** M **end procedure****for all** indices i in M **do** $j \leftarrow i_q + i$ **if** $z_{\mathbf{q}_j} < z_{th}$ **then** $\mathcal{P}_k \leftarrow \{\mathcal{P}_k, \mathbf{q}_j\}$ **end if****end for**

Abbildung 6.1: Suchmaske

s_p und s_f erhöht, verringert sich die Anzahl der Nachbarpunkte in \mathcal{P}_k . Dies ist besonders bei Verfahren interessant, die nicht alle Punkte in der Nachbarschaft benötigen (z. B. die Normaleberechnung in Abschnitt 6.2). Für die Werte $s_f = s_p = 1$ hingegen werden alle Punkte in der x-y-Nachbarschaft gefunden. Die Bestimmung der Nachbarpunkte erfolgt durch Verschieben von M über alle Punkte und Addition der Indizes der Maske i zum Index des Abfragepunkts i_q . Weiterhin kann über den Parameter z_{th} eine Distanzschwelle für die Tiefenwerte z eingestellt werden, um den Suchbereich auch in dieser Richtung einzuschränken.

6.1.1 Bewertung

Die ANN-Suche in ungeordneten Punktwolken ist sicherlich genereller einsetzbar als die Suche auf geordneten Daten. Allerdings liegen die Daten der in dieser Arbeit verwendeten Sensoren immer geordnet vor, so dass dies kein Vorteil ist. Die durchschnittliche Komplexität der ANN-Suche für eine Punktwolke mit n Punkten der Dimension k ist $O(\log n)$, wobei noch Zeit für die

Konstruktion des Suchbaums mit der Komplexität $O(kn \log n)$ hinzukommt. Die Komplexität der geordneten Suche hingegen hängt nicht von n ab, da pixelweise gesucht wird. Damit ergibt sich $O(1)$. Weiterhin hängt die Suchzeit vom Suchradius r ab. Bei der geordneten Suche ergibt sich $O(r^2)$. Bei der ANN-Suche ist die Abhängigkeit vom Suchradius nicht trivial. Auf jeden Fall hängt sie nach (Elseberg u. a., 2012) stark von der maximal erlaubten Suchreichweite r_{max} ab und skaliert polynomial mit dieser, also gilt $O(r_{max}^k)$. Aus der Abschätzung der Komplexität wird deutlich, dass die geordnete Suche deutlich effizienter ist als die ANN-Suche.

6.2 Punktnormalen

Das im Rahmen dieser Arbeit entwickelte Verfahren zur Berechnung von Normalen durch Nachbarschaftsbeziehungen (NNB) nutzt zum einen die schnelle Suche in geordneten Punktwolken und zum anderen die hohe Genauigkeit der Normalenvektoren durch Auswertung von Nachbarschaftsbeziehungen. Damit vereint NNB die Vorteile der in Abschnitt 4.3.3 genannten Ansätze der Berechnung über PCA (NPCA) und über Integralbilder (NIB).

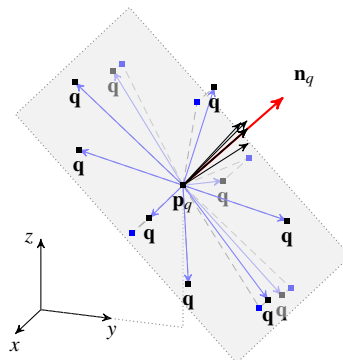


Abbildung 6.2: Berechnung des Normalenvektors nach NNB

Die Normalenberechnung sucht zunächst unter Verwendung des in Abschnitt 6.1 beschriebenen Verfahrens nach k Nachbarpunkten $\mathbf{q}_i, i = 1, \dots, k$ zum Abfragepunkt \mathbf{p}_q . Der Normalenvektor von \mathbf{p}_q ergibt sich dann zu

$$\mathbf{n}_q = \sum_{i,j,i \neq j}^k (\mathbf{q}_i - \mathbf{p}_q) \times (\mathbf{q}_j - \mathbf{p}_q) \quad (6.1)$$

als die Summe aller Kreuzprodukte der Vektoren zwischen \mathbf{p}_q und \mathbf{q}_i (Abbildung 6.2).

6.2.1 Vergleich der Verfahren zur Berechnung von Normalenvektoren

Die Verfahren zur Normalenberechnung sind bezüglich Genauigkeit und Rechenzeit zu evaluieren. Als Testszene wird eine ebene Wand gewählt, die im ersten Fall senkrecht zur optischen Achse des Sensors steht und im zweiten um 30° gedreht ist. In die Punktwolke der Wand wird mit

dem in Abschnitt 6.6.1 beschriebenen Verfahren eine Ebene eingepasst, um eine Grundwahrheit zu erzeugen.

Zur Evaluierung der Genauigkeit werden drei Messwerte herangezogen. Zunächst wird die Winkelabweichung der berechneten Normalen zum Erwartungswert betrachtet. Weiterhin wird der Prozentsatz „guter“ Normalen, die innerhalb des Toleranzbereichs von 14° liegen, angegeben. Schließlich wird noch das Verhältnis der berechneten Normalen zur Anzahl der Punkte in der Punktwolke untersucht. Der dritte Wert gibt darüber Aufschluss, ob ein Verfahren in bestimmten Situationen versagt. In Tabelle 6.1 sind die Messwerte dargestellt. Es fällt auf, dass die Genauigkeitswerte für NPCA und NNB fast identisch sind, wobei NIB etwas ungenauer rechnet. Auch beim Verhältnis der berechneten Normalen schneidet NIB leicht schlechter ab. Der Grund dafür liegt darin, dass NIB in der Nähe von Kanten und am Rand der Punktwolke keine Normalen berechnen kann.

	Neigung	NPCA	NIB	NNB
Mittlere Winkelabweichung (rad)	0°	0,004	0,014	0,005
Mittlere Winkelabweichung (rad)	30°	0,003	0,011	0,005
Anteil guter Normalen (%)	0°	86,4	74,4	86,0
Anteil guter Normalen (%)	30°	74,1	65,4	73,1
Verhältnis Normalen zur Punkteanzahl (%)	0°	87,5	86,1	87,5
Verhältnis Normalen zur Punkteanzahl (%)	30°	74,7	73,3	74,7

am schlechtesten mittel am besten

Tabelle 6.1: Genauigkeit der Verfahren zur Normalenberechnung

	NPCA	NIB	NNB
Rechenzeit (s)	2,03	0,062	0,025

Tabelle 6.2: Rechenzeit der Verfahren zur Normalenberechnung

Die Rechenzeit ergibt sich aus der wiederholten ($n = 1000$) Berechnung der Normalen mit anschließender Mittelwertbildung. Der Vergleich der Ergebnisse ist in Tabelle 6.2 zu sehen. Sie zeigt, dass NNB das schnellste Verfahren ist, gefolgt von NIB und NPCA.

Die Normalenberechnung durch NNB überzeugt aufgrund hoher Genauigkeit bei geringer Rechenzeit. Zwar erfüllt auch NIB diese Kriterien. Da aber Normalenvektoren gerade in der Nähe von Kanten für viele Verarbeitungsschritte notwendig sind, ist dieses Verfahren im Rahmen dieser Arbeit ungeeignet.

6.3 Rauschfilter

Rauschfilter dienen dazu, Messfehler der Sensoren zu kompensieren und damit die Rohsensordaten so aufzubereiten, dass eine Weiterverarbeitung erleichtert oder überhaupt erst möglich wird. In diesem Abschnitt werden zunächst Filter betrachtet, die speziell für Fehlmessungen von TOF-Kameras entworfen sind. Anschließend wird ein generell einsetzbarer Rauschfilter als effiziente Alternative zum statistischen Ausreißerfilter entwickelt und evaluiert. Schließlich wird analysiert, wie verschiedene Filter zu einer Filterkaskade zusammengeschaltet werden können, um das Filterergebnis zu optimieren.

6.3.1 Amplituden- und Abrisskantenfilter für TOF-Kameras

Auch wenn sich die Qualität der Sensordaten bei 3-D-Tiefenbildkameras in letzter Zeit deutlich verbessert hat, sind die Daten je nach Umgebungsbedingungen unterschiedlich stark rauschbehaftet. Besonders bei TOF-Kameras sorgen die Entstehung von Abrisskanten und die *non-ambiguity* Reichweite für verzerrte Messwerte (vgl. Abschnitt 4.1.1). Im Folgenden werden existierende Filterverfahren hinsichtlich der optimalen Parametereinstellung und in Bezug auf ihre Rechenzeit evaluiert.

Amplitudenfilter

Viele 3-D-Sensoren wie TOF-Kameras liefern zusätzlich zur Tiefeninformation auch einen Amplitudenwert für jedes Pixel. Dieser Wert ist abhängig von der Entfernung der Objekte zum Sensor und vom Umgebungslicht. Der in (Fuchs u. May, 2008) vorgestellte Amplitudenfilter eignet sich vorzugsweise dazu, Werte, die außerhalb der *non-ambiguity* Reichweite des Sensors liegen oder die durch Mehrfachreflexion erzeugt werden, aus der Punktwolke zu entfernen.

Die Arbeitsweise des Filters ist ein einfaches Schwellwertverfahren. Für jeden Punkt \mathbf{p}_i in der Punktwolke wird geprüft, ob die Bedingung

$$A_i < A_{th} \quad (6.2)$$

mit dem Amplitudenwert A_i und dem Schwellwert A_{th} zutrifft oder nicht. Normalerweise sollen Werte mit geringer Amplitude gefiltert werden. Das bedeutet, dass alle Punkte, für die Bedingung 6.2 erfüllt ist, aus der Punktwolke entfernt werden. Der Schwellwert A_{th} muss je nach Szene und Sensor bestimmt werden.

Abbildung 6.3 zeigt die Anwendung des Filters auf eine Punktwolke. Die Farbkodierung gibt die Amplitudenwertverteilung wieder (rot = niedrig, blau = hoch). In der Szene sind sowohl Bereiche mit großen Entfernungen (rechts oben) als auch mit schwachen Reflexionen (schwarze Felder des Schachbretts) zu finden, welche größtenteils durch den Filter entfernt werden.

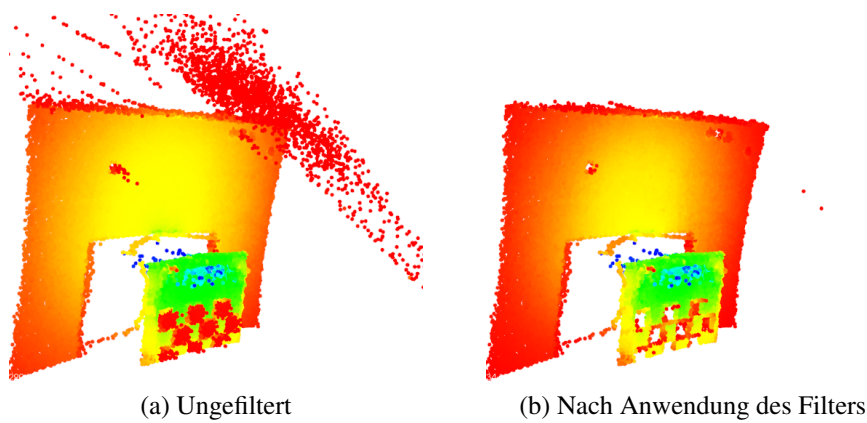


Abbildung 6.3: Anwendung des Amplitudenfilters auf eine Innenraumszene

Abrisskantenfilter

Abrisskanten sind ein weiterer störender Effekt, der besonders bei TOF-Kameras auftritt. Die an Kanten entstehenden fliegenden Pixel können mit Hilfe geometrischer Beziehungen identifiziert und entfernt werden. Dies setzt voraus, dass die Daten in geordneter Form vorliegen (Fuchs u. May, 2008).

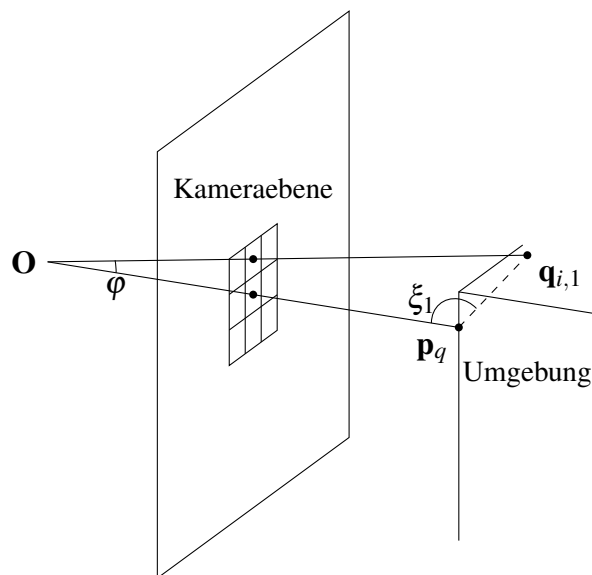


Abbildung 6.4: Geometrische Beziehungen für die Abrisskantenselektion

Die folgende Gleichung

$$\xi = \max_{k \in \{1, \dots, 8\}} \arcsin \left(\frac{\|\mathbf{q}_{i,k}\|}{\|\mathbf{q}_{i,k} - \mathbf{p}_q\|} \sin \varphi \right) \quad (6.3)$$

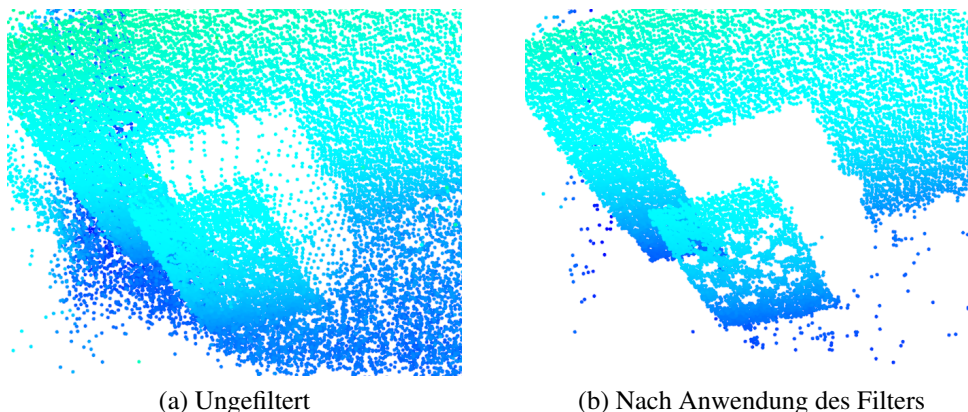


Abbildung 6.5: Anwendung des Abrisskantenfilters auf eine Innenraumszene

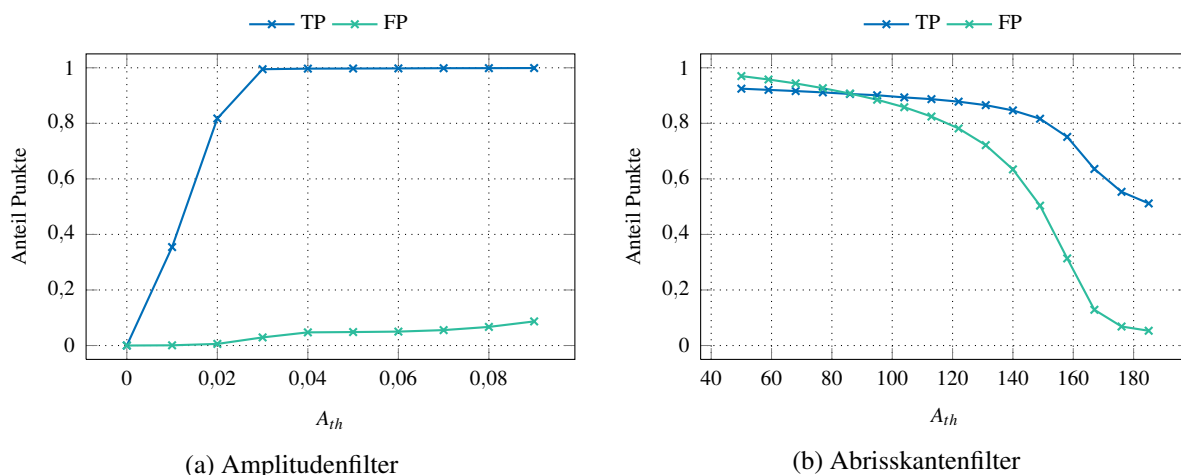


Abbildung 6.6: TP und FP in Abhängigkeit von Filterparametern

beschreibt den maximalen Winkel ξ zwischen dem Punkt \mathbf{p}_q und einem seiner acht direkten Nachbarn \mathbf{q}_i . Dabei ist φ der Winkel zwischen zwei benachbarten Pixeln bezogen auf das optische Zentrum \mathbf{O} . Abbildung 6.4 zeigt die geometrischen Beziehungen schematisch. Falls $\xi > \xi_{th}$ erfüllt ist, wird \mathbf{p}_q als einer Abrisskante zugehörig identifiziert und kann aus der Punktwolke entfernt werden.

Abbildung 6.5 zeigt die Entfernung von Abrisskanten aus einer Punktwolke. Die Punkte sind entsprechend der x -Koordinate farbkodiert. Es ist zu sehen, dass die Abrisskanten zwischen der Fläche im Vordergrund und dem Hintergrund gelöscht werden.

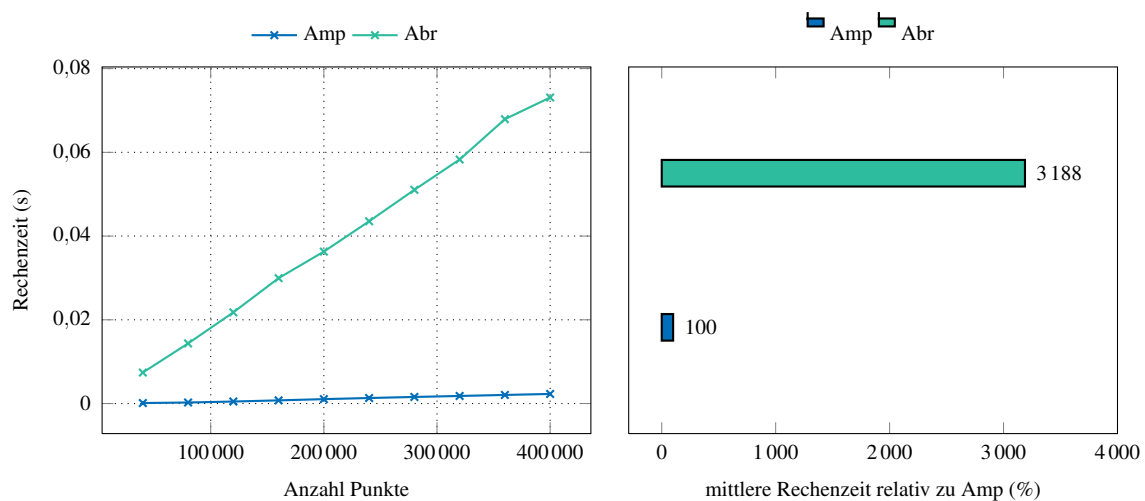


Abbildung 6.7: Verlauf der Rechenzeit für die einzelnen Filter in Abhängigkeit der Punktwolkengröße

Evaluierung

Für die Evaluierung hinsichtlich optimaler Parametereinstellung werden Daten von TOF-Kameras verwendet, da die zu filternden Fehlereffekte charakteristisch für diesen Kamertyp sind. Die Szenen sind so gewählt, dass sie die typischen Fehlereffekte der Sensoren herausstellen. Die zu entfernenden Rauschpunkte werden manuell markiert, um eine Aussage über die Genauigkeit der Filter machen zu können. Danach werden die Filter auf die Sensordaten angewendet, die *wahr Positiven* (*true positives*, TP) und *falsch Positiven* (*false positives*, FP) gezählt. Die TP sind die tatsächlich entfernten Rauschwerte, die FP hingegen entfernte, doch korrekte Messwerte.

Abbildung 6.6 zeigt Messwerte für die Evaluierung der Filtergenauigkeit für den Amplituden- und den Abrisskantenfilter. Als Testszene wird ein Schachbrett vor einer Wand verwendet (Abbildung 6.3). In mehreren Durchläufen wird der Schwellwert A_{th} variiert und der Anteil von TP und FP an den Gesamtpunkten aufgezeichnet. In Abbildung 6.6a ist zu sehen, dass bei $A_{th} = 0,03$ nahezu 100% des Amplitudenrauschens entfernt werden, wohingegen der Anteil der falsch entfernten Punkte bei unter 5% liegt. Dies stellt für diese Szene den optimalen Arbeitspunkt dar.

Beim Abrisskantenfilter werden die TP und FP in Abhängigkeit des Winkels ξ_{th} dargestellt (Abbildung 6.6b). Hier ist zu erkennen, dass mit steigendem Winkel sowohl die gefilterten Kantenpunkte als auch die falsch entfernten Punkte zunehmen. Dies liegt daran, dass der Abrisskantenfilter auch Punkte auf schräg zum Sensor stehenden Flächen, Gaußsches Rauschen und Punkte außerhalb der *non-ambiguity* Reichweite filtert. Weiterhin werden auch Punkte, welche direkt auf der Kante liegen, als Rauschen erfasst. Die optimale Einstellung des Filters hängt von der Anwendung ab. Soll konservativ gefiltert werden, um auf jeden Fall korrekte Messwerte zu erhalten, muss ξ_{th} möglichst groß gewählt werden. Sollen hingegen Rauschpunkte mit hoher Priorität entfernt werden, muss eine niedrige Winkelschwelle gesetzt werden.

Zur Abschätzung der benötigten Rechenzeit werden die Filter auf Punktwolken mit unterschiedlich vielen Punkten angewendet und die Filterzeit für alle Punkte gemessen. Um Punktwolken-

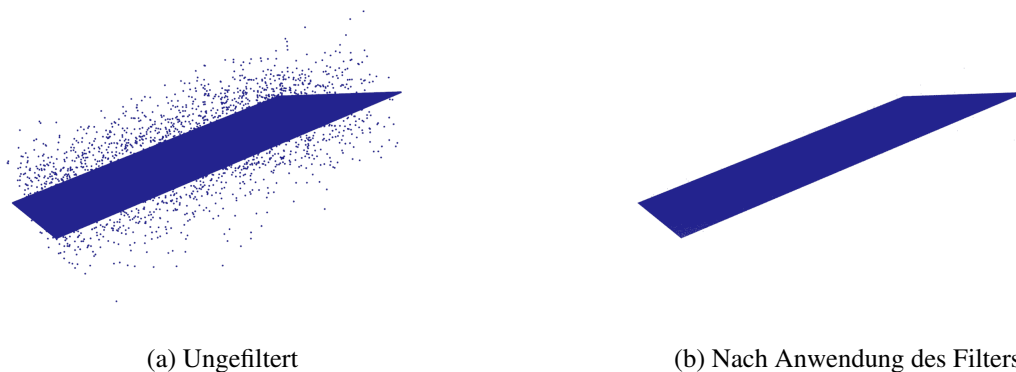


Abbildung 6.8: Anwendung des Specklefilters auf eine verrauschte Ebene

größen genau vorgeben zu können, werden die Daten künstlich generiert. Abbildung 6.7 zeigt die ermittelten Zeiten für Punktwolken mit Größen zwischen $n = 40\,000$ und $n = 400\,000$ Punkten. Es ist zu sehen, dass die Rechenzeit für beide Filter mit der Komplexität $O(n)$ skaliert. Die absolute Rechenzeit zeigt, dass beide Filter auch bei großen Punktwolken ausreichend schnell sind.

6.3.2 Allgemeine Rauschfilter

Allgemeine Rauschfilter entfernen Gaußsches Rauschen aus Punktwolken. Der im Stand der Technik beschriebene statistische Ausreißerfilter (SAF) ist ein solcher Filter, rechnet jedoch nicht effizient. Deswegen wird ein neuer Filteralgorithmus, der Specklefilter entworfen.

Specklefilter

Der Specklefilter ist ein allgemeiner Punktwolkenfilter, der versucht, homogene Bereiche in der Punktwolke zu identifizieren. Alle Pixel, die übrig bleiben, werden als Ausreißer gekennzeichnet. Somit kann Gaußsches Rauschen aus Tiefenbildern entfernt werden.

Dieser Algorithmus (vgl. Algorithmus 2) beginnt beim linken oberen Pixel und untersucht davon ausgehend die euklidische Distanz d_i der Nachbarschaft $\mathcal{P}_{i,j}$. Liegt diese in einem Toleranzbereich, d.h. $d_i < d_{th}$, so wird der Nachbarpixel als der aktuellen Region zugehörig markiert und einer Wellenfront w hinzugefügt. Solange sich noch Punkte in w befinden, kann die Region weiter wachsen. Unter Berücksichtigung dieser Regeln wird über alle Punkte der Wolke iteriert und diese somit in homogene Gebiete aufgeteilt. Regionen unterhalb einer Minimalgröße s_{min} werden dabei verworfen.

Abbildung 6.8 zeigt die Anwendung des Specklefilters auf eine Punktwolke. Es ist zu erkennen, dass die nicht zur homogenen Ebene gehörenden Punkte entfernt werden.

Algorithmus 2 Algorithmus für Specklefilter

```

Input: A pointcloud with points  $\mathbf{p}_{i,j}$ ,  $c$  columns and  $r$  rows
Define neighborhood of point  $\mathbf{p}_{i,j}$  as  $\mathcal{P}_{i,j} = \{\mathbf{p}_{i,j-1}, \mathbf{p}_{i,j+1}, \mathbf{p}_{i-1,j}, \mathbf{p}_{i+1,j}\}$ 
Label  $l_p$  of point  $\mathbf{p}_{i,j}$ 
for  $i = 1$  to  $c$  do
  for  $j = 1$  to  $r$  do
    if  $\nexists l_p$  then
      wavefront  $w \leftarrow \mathbf{p}_{i,j}$ 
      counter  $z \leftarrow 0$ 
       $l_p \leftarrow$  new label
    end if
    while  $w$  is not empty do
       $z \leftarrow z + 1$ 
      for all  $\mathbf{q} \in \mathcal{P}_{i,j}$  do
        if  $\nexists l_q$  and  $d_i < d_{th}$  then
           $w \leftarrow \{w, \mathbf{q}\}$ 
           $l_q \leftarrow l_p$ 
        end if
      end for
       $\mathbf{p}_{i,j} \leftarrow w(\text{end})$ 
    end while
    if  $z < s_{min}$  then
      drop speckle
    end if
  end for
end for

```

Evaluierung

Der neu entwickelte Specklefilter wird gegenüber SAF hinsichtlich Genauigkeit und Rechengeschwindigkeit evaluiert.

Zur Bewertung der Genauigkeit von statistischem Ausreißer- und Specklefilter werden die TP und FP bei unterschiedlichen Parametereinstellungen gezählt. Als Szene wird eine ebene Wand (Abbildung 6.8) gewählt, die Werte werden mit Gaußschem Rauschen ($\sigma = 0,05$ m) belegt. Abbildung 6.9a zeigt die Ergebnisse für den Specklefilter in Abhängigkeit von d_{th} . Es ist erkennbar, dass der Filter sehr konservativ arbeitet, da es keinerlei FP gibt. Die TP nehmen mit steigendem d_{th} ab. Daher ist es zumindest für diese Szene von Vorteil, eine möglichst kleine Schranke zu wählen. Die Ergebnisse für den statistischen Ausreißerfilter sind in Abbildung 6.9b in Abhängigkeit von d_{th} bei unterschiedlichem k aufgetragen. Für diese Szene hat k eine geringe Auswirkung auf das Filterergebnis. Wird d_{th} nicht zu gering gewählt, liegen die FP und TP nahezu im Optimum, wobei die TP mit steigendem d_{th} leicht abnehmen. Der optimale Arbeitspunkt für beide Filter liegt bei $d_{th} = 0,2$ m. Die Filtereigenschaften sind an diesem Punkt für beide Filter ähnlich, wobei SAF einen leicht besseren Wert für die TP zeigt.

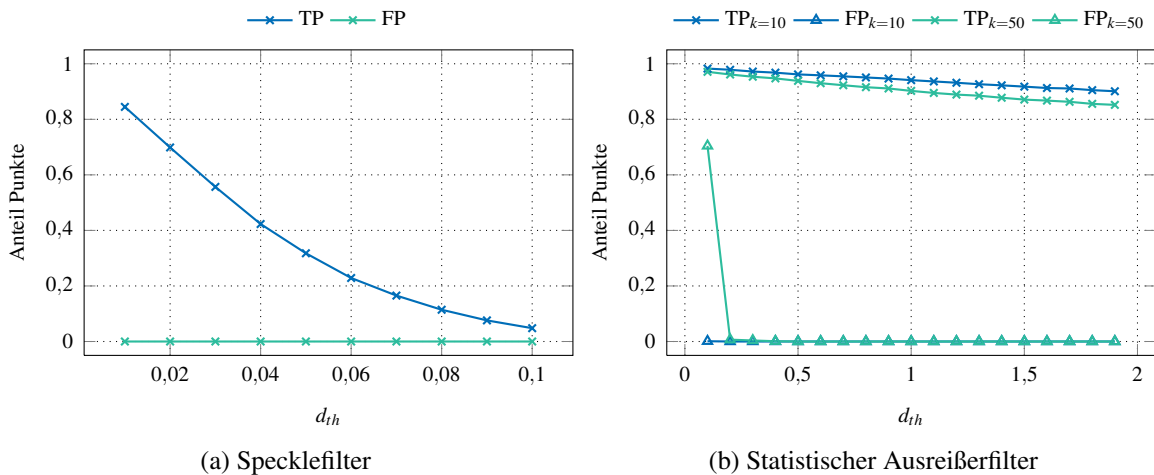


Abbildung 6.9: Wahr Positive (TP) und falsch Positive (FP) in Abhängigkeit von Filterparametern

Zur Abschätzung der benötigten Rechenzeit werden die Filter wiederum auf Punktwolken unterschiedlicher Größe angewendet und die Filterzeit für alle Punkte gemessen. Sowohl der Specklefilter als auch SAF skalieren linear ($O(n)$). SAF rechnet aber absolut gesehen deutlich langsamer, so dass er beispielsweise nicht für Punktwolken bei voller Sensorauflösung mit 30 Hz betrieben werden kann. Der Specklefilter hingegen ist durchschnittlich um den Faktor 60 schneller, wodurch er bei voller Bildrate eingesetzt werden kann.

6.3.3 Aufbau einer Filterkaskade

Häufig treten mehrere der Fehlereffekte wie Gaußsches Rauchen oder Abrisskanten gemeinsam in einer Punktwolke auf. Deswegen ist es von Vorteil, eine Filterkaskade aufzubauen, in der mehrere Filter sequenziell auf die Punktwolke angewendet werden. Die Reihenfolge der Filter in der Kaskade ist dabei ein wichtiger Faktor für die Qualität der resultierenden Wolke. Ein wichtiger Punkt ist, dass Filter, die geometrische Beziehungen in der Punktwolke verwenden (wie z. B. der Abrisskantenfilter) als erstes angewendet werden müssen, bevor die geometrische Struktur der Wolke durch andere Filter zerstört wird. Allgemeine Filter kommen sinnvollerweise am Ende der Kaskade zum Einsatz, um verbliebenes Rauschen zu entfernen. Somit ergibt sich eine Sequenz aus Abrisskanten-, Amplituden- und Specklefilter als optimale Reihenfolge der evaluierten Filter. Abbildung 6.11 zeigt die Anwendung der Filterkaskade auf eine Punktwolke. Die fehlerhaften Werte werden dabei schrittweise entfernt. Die Gesamtrechenzeit für die Kaskade bei einer typischen TOF-Kameraauflösung von 204 x 204 Bildpunkten ist mit 0,008 s sehr gering, so dass keine Probleme bei der Einhaltung der Aktualisierungsrate des Gesamtsystems auftreten.

Der Einsatz von Rauschfiltern, insbesondere in einer Filterkaskade, trägt dazu bei, die Anforderung nach Robustheit gegenüber ungenauen Messdaten (A13) zu erfüllen.

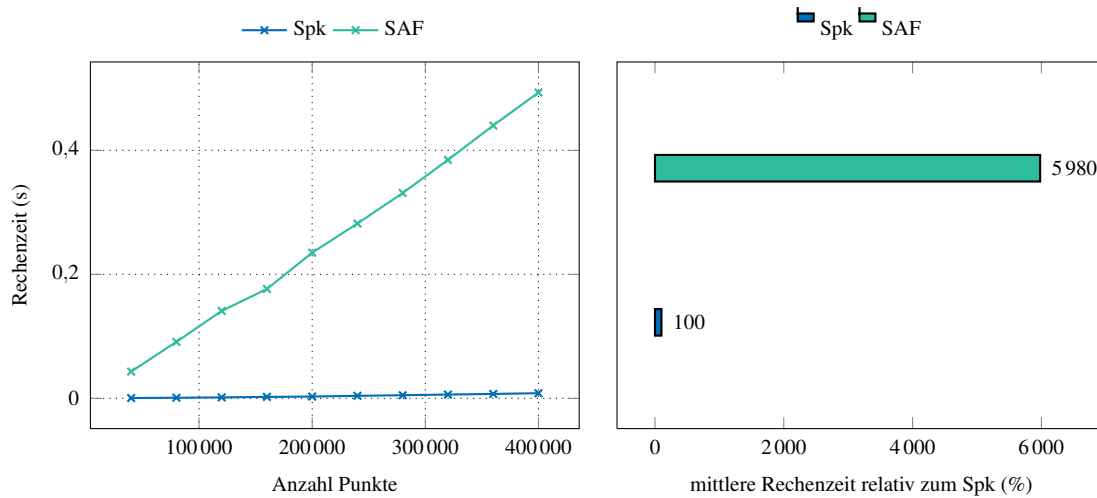


Abbildung 6.10: Verlauf der Rechenzeit für den Specklefilter (Spk) und den statistischen Ausreißerfilter (SAF) in Abhängigkeit der Punktwolkengröße

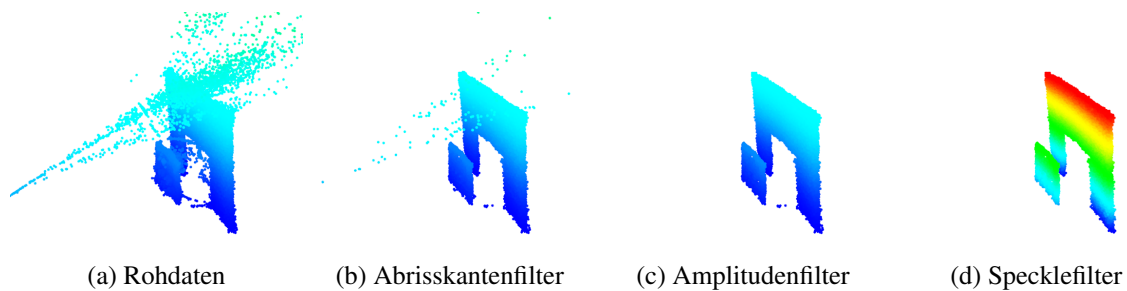


Abbildung 6.11: Filterkaskade

6.4 Registrierung

Da sowohl Sensordaten als auch die Schätzung der Roboterposition fehlerbehaftet sind, müssen Verfahren zur Registrierung angewendet werden, um eine konsistente Punktekarte der Umgebung erzeugen zu können. Grundsätzlich geht es bei der Registrierung darum, die Transformation T zwischen Sensordaten von unterschiedlichen Aufnahmepositionen zu finden. Dafür müssen zunächst Schlüsselpunkte in den Datensätzen identifiziert werden, die nachfolgend untereinander assoziiert werden (Korrespondenzfindung). Die Berechnung der Transformation aus den Korrespondenzen ist dann für gewöhnlich ein überbestimmtes Optimierungsproblem. In den letzten Jahren war die Registrierung im Fokus zahlreicher Forschungsarbeiten (Abschnitt 4.3.6). Dabei haben alle ICP-basierten Varianten das Problem, dass die Registrierung häufig in einem lokalen Minimum terminiert und der Rechenaufwand für die Korrespondenzfindung hoch ist. Weiterhin wurden viele Verfahren für schwenkbare Laserscanner entwickelt und optimiert, was ihren Einsatz bei Tiefenbildkameras erschwert. Aus diesen Gründen wird im Rahmen dieser Arbeit ein

neuer Registrierungsansatz entwickelt, der im Folgenden vorgestellt und gegenüber den wichtigsten gebräuchlichen Verfahren (ICP, GICP, FICP) evaluiert werden soll.

6.4.1 Fast Range Image Registration (FRIR)

Das im Rahmen dieser Arbeit entwickelte Registrierungsverfahren heißt *Fast Range Image Registration* (FRIR). Dieses ist auf extrem schnelle Verarbeitung der Daten bei gleichzeitig robuster und genauer Registrierung ausgelegt. Als Eingangsdaten verwendet es ausschließlich geordnete Punktwolken. Wie die vorherigen Verfahren gliedert sich auch FRIR in Assoziation und Transformationsberechnung. Punkte, die sich von ihrer Umgebung räumlich abgrenzen und damit einen erhöhten Informationsgehalt besitzen, werden zur Korrespondenzfindung herangezogen. Die Transformation wird über *Singular Value Decomposition* (SVD) (Arun u. a., 1987) ermittelt.

Das Verfahren basiert darauf, dass eine Transformation zwischen zwei korrespondierenden Punkten \mathbf{a}_i und \mathbf{b}_i der Punktwolken A und B

$$\mathbf{b}_i = \mathbf{R}\mathbf{a}_i + \mathbf{t} \quad (6.4)$$

in eine Translationsdistanz d_{ij}^t und eine Rotationsdistanz d_{ij}^r aufgespalten werden kann. Aus den beiden Punkten lässt sich jedoch nicht berechnen, wie groß die beiden Anteile sind. Bei einer beschränkten Bewegungsgeschwindigkeit des Sensors lassen sich obere Schranken d_{max}^t und d_{max}^r für die Distanzen angeben.

Extraktion von Punkten mit hohem Informationsgehalt

Die Grundidee dieses Schritts ist es, für alle Punkte \mathbf{a}_i der Punktwolke A die Korrespondenzmenge C_i aus der Punktwolke B gemäß

$$C_i = \{ \mathbf{b}_j \mid d_{ij}^t \leq d_{max}^t \wedge d_{ij}^r \leq d_{max}^r \} \quad (6.5)$$

zu bestimmen. Dabei können d_{ij}^t und d_{ij}^r nicht berechnet werden, da sie gekoppelt sind, es kann aber eine Ableitung der oberen Schwelle erfolgen. Als Abschätzung für die Translation gilt

$$d_{ij}^t \leq \left| \|\mathbf{a}_i\| - \|\mathbf{b}_j\| \right| \quad (6.6)$$

bei der Annahme eines maximalen Translationsanteils an der Transformation. Der tatsächliche Rotationsanteil wird über die zugehörige Kreissehne s definiert (siehe Abbildung 6.12). Die Sehne wird durch den Kreissektor beschrieben, der über den Vektor \mathbf{a}_i und den durch die Rotation \mathbf{R} entstandenen Vektor \mathbf{b}'_j bestimmt ist. Für die Abschätzung von d_{ij}^r hingegen wird die Strecke s' zu

$$s' = \|\mathbf{b}_j - \mathbf{a}_i\| - d_{max}^t \quad (6.7)$$

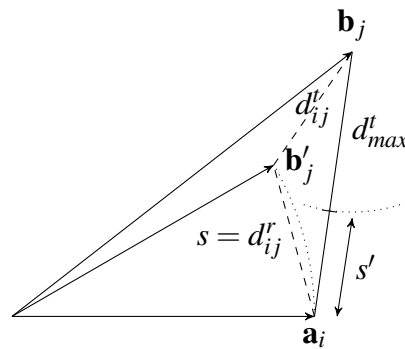


Abbildung 6.12: Abschätzung der Rotationsdistanz

berechnet. Nach der Dreiecksungleichung im von s und s' aufgespannten Dreieck gilt $s' \leq s$ und damit

$$d'_{ij} \leq s'. \quad (6.8)$$

Damit können die Bedingungen in (6.5) überprüft werden. Im zweiten Schritt wird der Informationsgehalt

$$I_i = \frac{1}{|C_i|} \quad (6.9)$$

für jeden Punkt \mathbf{a}_i bestimmt, wobei $|C_i| \neq 0$ gelten muss. Punkte in A haben also dann einen hohen Informationsgehalt, wenn die Anzahl ihrer möglichen Korrespondenzpunkte aus B gering ist.

Die effiziente Berechnung von I erfolgt in einem zweistufigen Ansatz, der speziell auf die Verarbeitung von Tiefenbildern zugeschnitten ist (vgl. Algorithmus 3). Die Gültigkeit der Berechnungen beruht auf zwei Annahmen. Erstens ist es für eine erfolgreiche Registrierung nicht nötig, für alle Punkte in A Korrespondenzen zu finden, sondern es genügt bereits eine Teilmenge. Weiterhin liefert die Mehrzahl der Punkte (z. B. solche auf kontinuierlichen Oberflächen) wenig Information für eine erfolgreiche Registrierung. Bei Verwendung aller Punkte kann daher sogar ein schlechteres Registrierungsergebnis auftreten. Zweitens wird $d'_{ij} = 0$ angenommen. Dies ist möglich, da die Bewegungsgeschwindigkeit der Kamera beschränkt und die Bewegung zwischen zwei Bildaufnahmen gering ist.

Folglich kann im ersten Schritt das elementweise Differenzbild $F = \{f_i\}$ von A und B nach

$$f_i = \begin{cases} 1, & \text{falls } |\mathbf{a}_i - \mathbf{b}_i| > d_{th} \\ 0, & \text{sonst} \end{cases} \quad (6.10)$$

gebildet werden.

Im zweiten Berechnungsschritt werden aus F Kantenpunkte extrahiert, um die Menge der Punkte für die Korrespondenzfindung weiter einzuschränken. Dabei sind nur „vordere“ Kantenpunkte in-

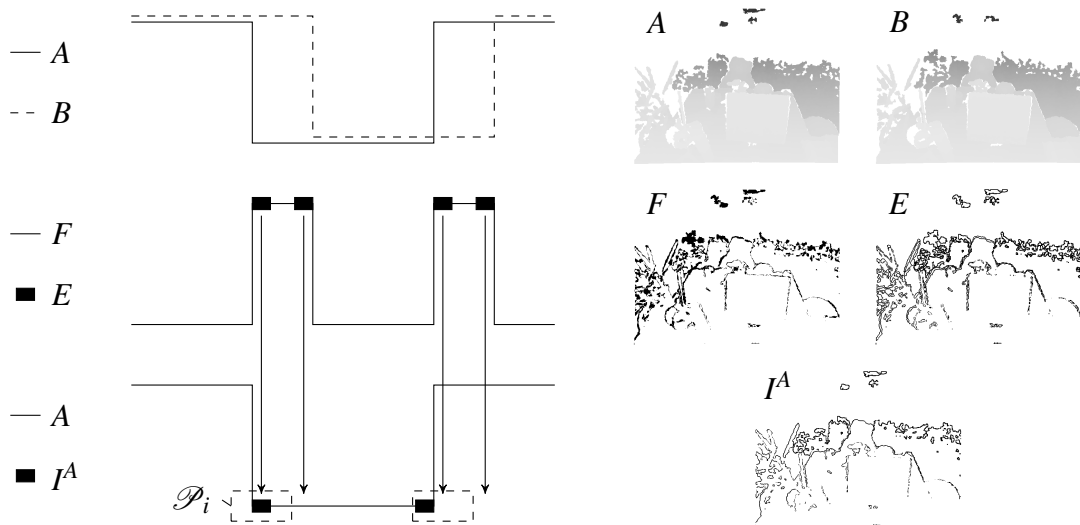


Abbildung 6.13: Extraktion von Punkten mit hohem Informationsgehalt

teressant, also solche, die auf der näher zum Blickpunkt liegenden Oberfläche liegen. Für die Bestimmung von Kantenpunkten wird die lokale Nachbarschaft jedes Punkts $f_i = 1$ betrachtet. Die Nachbarschaft um den Abfragepunkt \mathbf{q}_i wird über einen Pixelradius d^r zu $\mathcal{F}_i = \{f_i \mid d(\mathbf{q}_i) \leq d^r\}$ mit dem Pixelabstand $d(\mathbf{q}_i)$ definiert. Das Kantengewicht ergibt sich dann aus dem Anteil von Kantenpunkten in \mathcal{F} zu

$$w_i = \frac{|\{f_i \in \mathcal{F}_i \mid f_i = 0\}|}{|\mathcal{F}_i|}. \quad (6.11)$$

Die Menge an Kantenpunkten

$$E = \{f_i \mid w_{min} < w_i < w_{max}\} \quad (6.12)$$

wird über eine Auswertung der Gewichte bestimmt. Dabei werden einerseits Punkte mit kleinem Gewicht ausgeschlossen, da diese weit entfernt von einer Kante liegen. Andererseits sind Punkte mit sehr hohem Gewicht instabil, da sie direkt auf der Kante liegen und bereits durch geringes Rauschen auf den Bereich hinter der Kante rutschen können. Deswegen werden diese ebenfalls nicht betrachtet.

Da durch die Differenzbildung die Information verloren gegangen ist, ob die Kantenpunkte zu A oder B gehören, wird geprüft, in welchem Bild sich die zugehörige Kante tatsächlich befindet. Hierzu wird in der Nachbarschaft \mathcal{P}_i der Pixelkoordinaten von e_i lokal nach einer „vorderen“ Kante in A und B gesucht, was zur Teilmenge

$$\mathcal{Q}_i = \{p_i \mid d^t > d_{th}^t\} \quad (6.13)$$

führt. Die Kantenpunkte und damit Punkte mit hohem Informationsgehalt für A und B ergeben sich dann zu

$$I^A = \{e_i \mid |\mathcal{Q}_i^A| > 0\} \quad (6.14)$$

$$I^B = \{e_i \mid |\mathcal{Q}_i^B| > 0\}, \quad (6.15)$$

d. h. I^A und I^B enthalten alle Punkte aus E , die entweder in A oder in B an einer Kante liegen.

Für den Fall, dass die Koordinate von e_i auf einer „hinteren“ Kante liegt, wird sie so korrigiert, dass der Kantenpunkt auf eine „vordere“ Kante verschoben wird (siehe Abbildung 6.13). Damit sind die Mengen möglicher Korrespondenzpunkte für die Ableitung der Transformation bestimmt.

Algorithmus 3 FRIR: Extraktion von Punkten mit hohem Informationsgehalt

Require: Two ordered point clouds $A = \{\mathbf{a}_i\}$ and $B = \{\mathbf{b}_j\}$

Ensure: Information Points I^A, I^B

Binary differential image $F \leftarrow f(A - B, d_{th})$

Edge points E

for all $f_i \in F$ **do**

if $f_i = 1$ **then**

 Weight $w_i \leftarrow$ ratio of edge points in neighborhood \mathcal{F}_i

if $w_{min} < w_i < w_{max}$ **then**

$E \leftarrow \{E, f_i\}$

end if

end if

end for

for all $e_i \in E$ **do**

$\mathcal{Q}_i^A \leftarrow$ border points in A at coordinates of e_i

$\mathcal{Q}_i^B \leftarrow$ border points in B at coordinates of e_i

if $|\mathcal{Q}_i^A| > 0$ **then**

$I^A \leftarrow \{I^A, e_i\}$

end if

if $|\mathcal{Q}_i^B| > 0$ **then**

$I^B \leftarrow \{I^B, e_i\}$

end if

end for

Korrespondenzfindung und Transformationsberechnung

Bevor die Transformation berechnet werden kann, müssen aus den Punkten mit hohem Informationsgehalt I^A, I^B Korrespondenzkandidaten bestimmt werden. Hierzu werden für alle Punktpaare $(\mathbf{i}_i^A, \mathbf{i}_j^B)$ die Translationsdistanz d_{ij}^t aus (6.6) und die Rotationsdistanz d_{ij}^r aus (6.8) bestimmt. Aus dieser Berechnung ergeben sich Korrespondenzmengen C^A, C^B , die zu jedem Punkt in A

wobei sich \mathbf{V} und \mathbf{U} aus der Singulärwertzerlegung der Matrix

$$\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T = \sum_{i,j} w_{ij} \mathbf{a}_i^T \mathbf{b}_j \quad (6.20)$$

ergeben. Die Lösung des Singulärwertproblems wird iterativ numerisch berechnet.

Die resultierende Transformation wird auf B angewendet. Alle Schritte von Algorithmus 4 werden so lange wiederholt, bis sich \mathbf{T} zwischen zwei Iterationen nur noch um maximal ε ändert. Zusätzlich werden mit jeder Iteration die Maximalwerte für die Translations- und Rotationsdistanz

$$d_{max,t}^t \leq \frac{d_{max,t-1}^t}{2} \quad (6.21)$$

$$d_{max,t}^r \leq \frac{d_{max,t-1}^r}{2} \quad (6.22)$$

neu berechnet, um die Menge möglicher Korrespondenzpunkte mit jedem Schritt weiter einzuschränken.

6.4.2 Evaluierung

Die vorgestellten Verfahren werden in Bezug auf Genauigkeit und Echtzeitfähigkeit evaluiert. Dazu werden automatisch generierte, synthetische Szenen und der Datensatz der Universität Freiburg (Sturm u. a., 2012) verwendet. Die synthetischen Szenen zeigen größtenteils die Küchenumgebung am Fraunhofer IPA, wobei zusätzliche Objekte hinzugefügt wurden. Weiterhin wird unterschiedlich starkes Sensorrauschen simuliert. Der Datensatz der Universität Freiburg besteht größtenteils aus Büroszenen, in denen eine Tiefenbildkamera unter Beschränkung der Freiheitsgrade oder frei bewegt wird. Die Trajektorie der Kamera wird dabei aufgezeichnet, so dass eine Grundwahrheit vorhanden ist. Abbildung 6.15 zeigt Beispiele für einige der evaluierten Szenen. Es werden für die Szenen optimierte Parametersätze verwendet. Für ICP und GICP wird die maximale Korrelationsdistanz auf 0,2 m und die Abbruchsbedingung ε auf $10^{-3.5}$ gesetzt. Es werden maximal 40 Iterationen durchgeführt. Bei FRIR wird eine maximale Kamerageschwindigkeit von 0,1 m/s bzw. 0,1 rad/s angenommen. Der Wert für w_{min} liegt bei 1, w_{max} bei 17. Für FICP wird das maximale Distanzverhältnis zwischen Korrespondenzpunkten auf 0,95 gesetzt.

Für die Genauigkeitsuntersuchung werden die absoluten und relativen mittleren Transformationsfehler E^{abs} und E^{rel} berechnet. Der Transformationsfehler wird allgemein durch die Fehlertransformation

$$\mathbf{E} = \mathbf{T}_g^{-1} \mathbf{T}_e \quad (6.23)$$

zwischen der Transformation der Grundwahrheit \mathbf{T}_g und der von den Verfahren berechneten Transformation \mathbf{T}_e definiert. Der Fehler kann in einen Translationsanteil

$$e_t(\mathbf{E}) = \|\text{trans}(\mathbf{E})\| \quad (6.24)$$

Algorithmus 4 FRIR: Korrespondenzfindung und Transformationsberechnung**Require:** Information points I^A, I^B **Ensure:** Transformation $\mathbf{T} = (\mathbf{R}, \mathbf{t})$ that transforms B into A

```

while true do
  Correspondence candidates  $C^A, C^B$ 
  for all  $\mathbf{i}_i^a \in I^A$  do
    for all  $\mathbf{i}_j^b \in I^B$  do
       $d_{ij}^t \leftarrow$  translation distance  $(\mathbf{i}_i^a, \mathbf{i}_j^b)$ 
       $d_{ij}^r \leftarrow$  rotation distance  $(\mathbf{i}_i^a, \mathbf{i}_j^b)$ 
      if  $d_{ij}^t < d_{max}^t \wedge d_{ij}^r < d_{max}^r$  then
         $c_i^A \leftarrow \{c_i^A, j\}$ 
         $c_j^B \leftarrow \{c_j^B, i\}$ 
      end if
    end for
  end for
  Correspondence pairs  $P$ 
  for all  $c_i^A$  do
    for all  $c_j^B$  do
      weight  $w_{ij} \leftarrow \frac{1}{\min(\|c_i^A\|, \|c_j^B\|)}$ 
       $P \leftarrow \{P, (\mathbf{a}_i, \mathbf{b}_j, w_{ij})\}$ 
    end for
  end for
   $\mathbf{T} \leftarrow \arg \max (\sum_{i,j} w_{ij} \|\mathbf{T}\mathbf{b}_j - \mathbf{a}_i\|^2)$ 
   $d_{max}^t \leftarrow d_{max}^t / 2$ 
   $d_{max}^r \leftarrow d_{max}^r / 2$ 
  if  $\mathbf{T} - \mathbf{T}_{-1} < \varepsilon$  then
    break
  end if
end while

```

und in einen Rotationsanteil

$$e_r(\mathbf{E}) = \bar{d} \arccos \left(\frac{\text{spur}(\text{rot}(\mathbf{E})) - 1}{2} \right) \quad (6.25)$$

unterteilt werden. Dabei wird ausgenutzt, dass eine Rotation um mehrere Achsen immer durch eine Rotation um eine virtuelle Achse und damit durch einen einzigen Rotationswinkel beschrieben werden kann (Ma, 2004). \bar{d} wird durch den mittleren Abstand der Punkte der beiden Punktwolken um den Rotationsmittelpunkt definiert. Für die evaluierten Szenen gilt $\bar{d} = 2$ m.

Der relative Transformationsfehler

$$\mathbf{E}^{rel} = (\mathbf{Q}_i^{-1} \mathbf{Q}_{i+1})^{-1} (\mathbf{P}_i^{-1} \mathbf{P}_{i+1}) \quad (6.26)$$

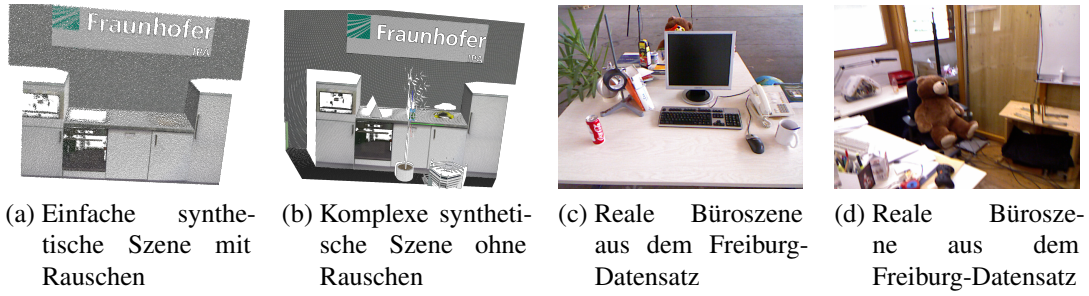


Abbildung 6.15: Beispiele für evaluierte Szenen

ist durch (6.23) mit der wahren Transformation \mathbf{Q} und der berechneten Transformation \mathbf{P} zwischen den Punktwolken i und $i + 1$ definiert. Damit gilt bei n registrierten Punktwolken, dass

$$e_r^{rel} = \frac{1}{n} \sum_{i=1}^n e_r(\mathbf{E}^{rel}) \quad (6.27)$$

$$e_t^{rel} = \frac{1}{n} \sum_{i=1}^n e_t(\mathbf{E}^{rel}). \quad (6.28)$$

Der absolute Transformationsfehler E^{abs} bestimmt den Fehler der berechneten Pose am Ende der Trajektorie in Bezug auf die Grundwahrheit. Somit kann er über

$$\mathbf{E}^{abs} = \frac{1}{n} \mathbf{T}_g^{-1} \mathbf{T}^e \quad (6.29)$$

mit

$$\mathbf{T}_g = \prod_{i=1}^n \mathbf{Q}_i \quad (6.30)$$

$$\mathbf{T}^e = \prod_{i=1}^n \mathbf{P}_i \quad (6.31)$$

berechnet werden. Die Gesamtfehler e^{rel} und e^{abs} ergeben sich aus der Summe der Translations- und Rotationsfehler.

Die über alle Szenen gemittelten Fehler sind in Abbildung 6.16 zu sehen und zeigen, dass FRIR sowohl den geringsten relativen (Abbildung 6.16a) als auch den geringsten absoluten (Abbildung 6.16b) Fehler hat. Allen Verfahren ist gemeinsam, dass der Translationsfehler größer als der Rotationsfehler ist (vgl. auch (Hampp, 2012)).

Interessant ist das in Abbildung 6.17 aufgetragene Verhalten bei Stillstand der Kamera. Bei FICP entsteht ein sehr großer Fehler, da die Genauigkeit der texturbasierten Merkmale sowohl unter der notwendigen Kalibrierung zwischen Farb- und Tiefenbild als auch unter dem Rauschen der Tiefenwerte leidet. ICP und GICP zeigen einen kleineren Fehler als während der Bewegung, der

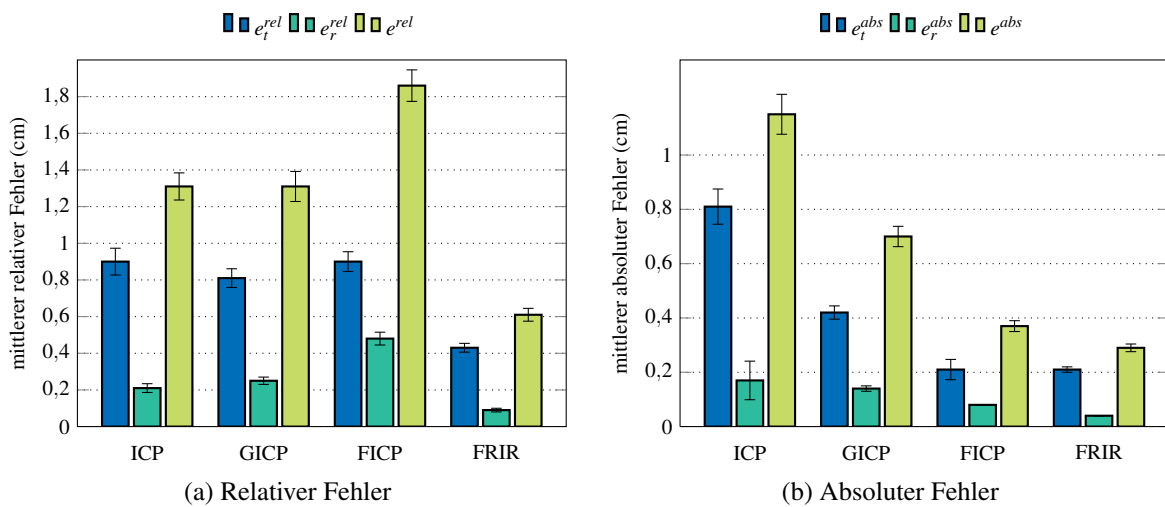


Abbildung 6.16: Transformationsfehler, gemittelt über alle registrierten Punktwolken. Standardabweichung als Fehlerindikator

aber durchaus signifikant ist. FRIR hingegen ist in der Lage, implizit zu erkennen, ob sich die Kamera bewegt. Dadurch entsteht lediglich ein sehr kleiner Fehler.

Da es durch eine falsche Registrierung zu einer Vergrößerung des Transformationsfehlers kommen kann, ist es notwendig, sich die relative Auswirkung der Registrierungsverfahren auf den Fehler anzuschauen. Dabei entspricht 100% der Annahme, dass sich der Sensor nicht bewegt hat. Dies ist gleichbedeutend mit der Tatsache, überhaupt keine Registrierung zu verwenden. In Abbildung 6.18a sind die Verfahren im Vergleich aufgetragen. Es ist zu erkennen, dass alle Verfahren den Transformationsfehler verkleinern, jedoch in unterschiedlichem Maße. ICP hat sein Maximum bei ca. 80%, was bedeutet, dass der Transformationsfehler nur zu einem kleinen Teil reduziert wird. Ähnliches gilt für GICP, wobei hier der höchste Wert bei 40% liegt. FICP und FRIR haben keine so ausgeprägte Spitze. Beide haben einen hohen Anteil an Szenen, bei denen der Fehler auf 0% reduziert wurde, was einer Konvergenz im globalen Minimum entspricht. FICP zeigt jedoch einen größeren Anteil an Szenen, bei denen sogar eine Vergrößerung des Fehlers im Vergleich zur Annahme des Stillstands auftritt. In Abbildung 6.18b ist zu sehen, um wieviel Prozent die Verfahren im Mittel den Transformationsfehler reduzieren. Während GICP, FICP und FRIR hier eine deutliche Reduktion von etwa 50% erzielen, schneidet ICP mit 26,5% deutlich schlechter ab.

Für den Vergleich der Rechenzeit wird über viele registrierte Punktwolken gemittelt, um Schwankungen durch unterschiedliche Prozessorauslastung auszugleichen. Tabelle 6.3 zeigt, dass FRIR am schnellsten ist, gefolgt von ICP. GICP und FICP sind die langsamsten Verfahren, was an den rechenintensiven Texturmerkmalen liegt. Der Speicherverbrauch wird als maximale Belegung pro Punktwolke angegeben. Auch hier liegt FRIR vorn. FICP benötigt am meisten Speicher während GICP und ICP dazwischen liegen.

Abbildung 6.19 zeigt die Verfahren im visuellen Vergleich anhand einer Szene des Freiburg-Datensatzes. Bei ICP ist zu sehen, dass die registrierten Punktwolken sich nicht völlig überde-

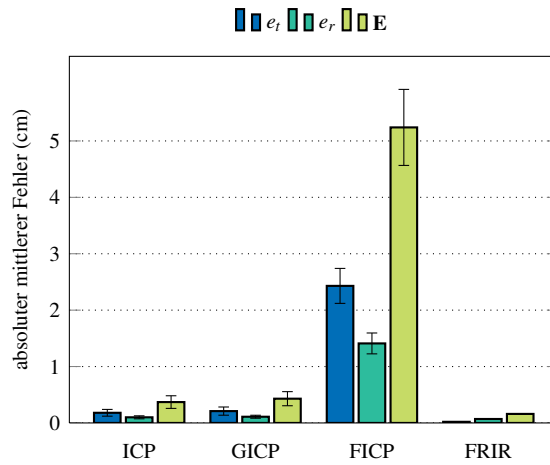
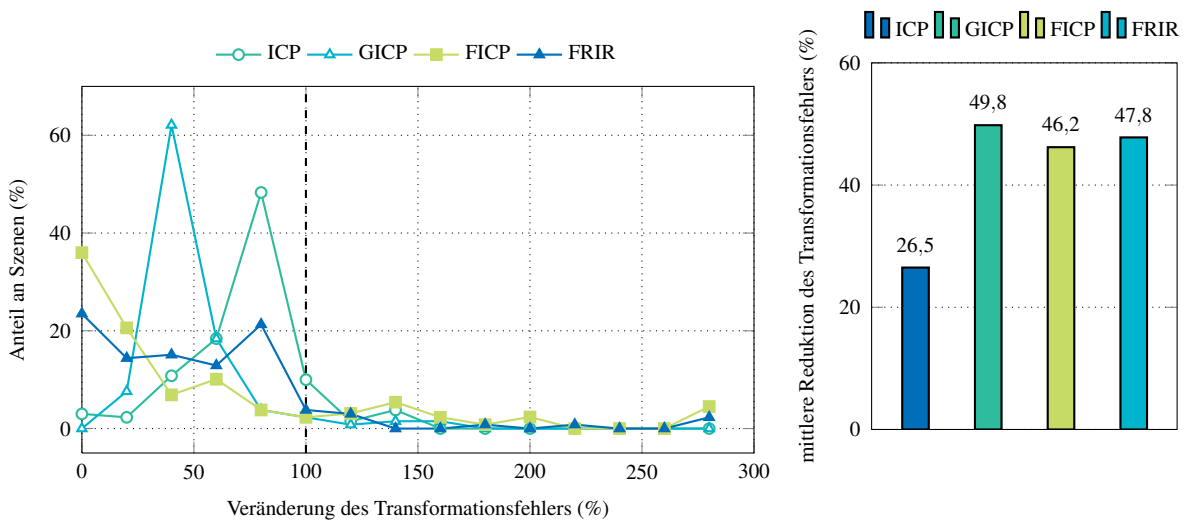


Abbildung 6.17: Absoluter Transformationsfehler bei Stillstand des Sensors, gemittelt über alle registrierten Punktwolken. Standardabweichung als Fehlerindikator



(a) Veränderung des Transformationsfehlers gemittelt über alle Szenen (b) Mittlere Reduktion des Transformationsfehlers im Vergleich

Abbildung 6.18: Relative Auswirkung der Registrierung auf den Transformationsfehler (100% entspricht keiner Registrierung)

Verfahren	Rechenzeit (s)	Speicherverbrauch (MB)
ICP	0,14	14
GICP	0,45	15
FICP	0,53	46
FRIR	0,02	2

Tabelle 6.3: Durchschnittliche Rechenzeit der Registrierungsverfahren im Vergleich (jeweils pro registriertem Tiefenbild)

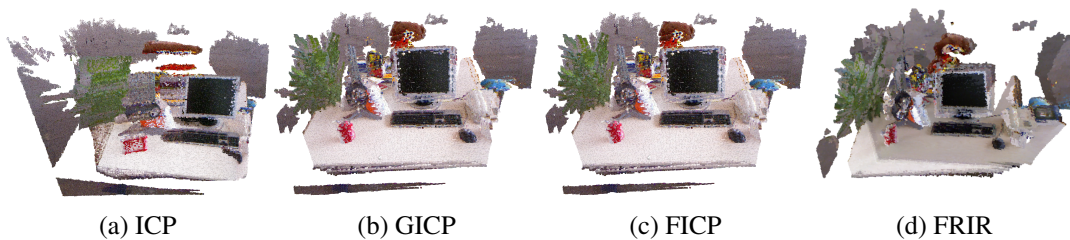


Abbildung 6.19: Visueller Vergleich der Verfahren

cken (Pflanze im Hintergrund stark verwischt). Bei GICP und FICP ist die Qualität höher, es sind jedoch leichte Fehler, z. B. beim Monitorrand oder an der Tischkante zu erkennen. FRIR zeigt auch im Bild die beste Qualität, was die Zahlenwerte der Evaluierung bestätigt.

Bei Betrachtung der Evaluierungsergebnisse ist festzustellen, dass FRIR das für diesen Verarbeitungsschritt am besten geeignete Verfahren ist. Die Genauigkeit ist bei allen gemessenen Kriterien höher als bei den anderen Verfahren. Durch die Erkennung von Stillstand ist FRIR in der Lage, den Registrierungsfehler noch weiter zu reduzieren. Bei Rechenzeit und Speicherverbrauch schneidet FRIR deutlich am besten ab. Dies liegt vor allem daran, dass die geordnete Datenstruktur von Tiefenbildern ausgenutzt wird, was bei den anderen Verfahren nicht der Fall ist. Die Genauigkeitsanforderung and die PK (A2) kann durch FRIR eingehalten werden, da das Verfahren einen maximalen absoluten Fehler von unter 0,3 cm aufweist. Ebenfalls erfüllt wird die Forderung A3, da FRIR mit einer vollen Bildrate von 30 Hz arbeiten kann.

6.5 3-D-Punktmerkmale

3-D-Punktmerkmale sind ein wichtiges Werkzeug, um Informationen aus Punktwolken zu extrahieren. Im Rahmen dieser Arbeit werden solche Merkmalstypen betrachtet, die jedem Punkt einer Punktwolke geometrische Eigenschaften zuordnen. Aus dem Stand der Technik (Abschnitt 4.3.5) wird deutlich, dass es bereits leistungsfähige und robuste Algorithmen in diesem Bereich gibt. Somit ist keine Neuentwicklung eines Merkmalstyps erforderlich. Vielmehr sollen in diesem Kapitel drei gängige Verfahren gegeneinander evaluiert und das für diese Arbeit am besten geeignete bestimmt werden. Die zu untersuchenden Punktmerkmalstypen sind *Radius-based Surface Descriptor*, *Principal Curvatures* und *Fast Point Feature Histograms*.

6.5.1 Radius-based Surface Descriptor (RSD)

Der *Radius-based Surface Descriptor* (RSD) beschreibt einen Punkt in einer Punktwolke durch die Radii von Kurven, die durch die Nachbarschaft des Punktes gelegt werden. Jeder Merkmalspunkt wird über den minimalen und maximalen Krümmungsradius (r_{min} , r_{max}) beschrieben. Algorithmus 5 zeigt die Berechnung des RSD für einen Punkt \mathbf{p}_q .

Algorithmus 5 Radius-based Surface Descriptor

Require: k normals \mathbf{n}_i and position vectors \mathbf{p}_i in neighborhood \mathcal{P}_k of \mathbf{p}_q

Ensure: minimum radius r_{min} , maximum radius r_{max}

init array $B[n][2]$

for all points \mathbf{p}_i in \mathcal{P}_k **do**

angle $\alpha_i \leftarrow \arccos(\mathbf{n}_q \cdot \mathbf{n}_i)$

distance $d_{iq} \leftarrow \|\mathbf{p}_i - \mathbf{p}_q\|$

$b_k \leftarrow \text{floor}(nd_{iq}/d_{max})$

$B[b_k][0] \leftarrow \min(\alpha_i, B[b_k][0])$

$B[b_k][1] \leftarrow \max(\alpha_i, B[b_k][1])$

end for

$r_{max} \leftarrow$ linear regression for $(\alpha_m = B[m][0], d_m = d_{max}(m + 0.5)/n)$ with $m = [0, \dots, n - 1]$

$r_{min} \leftarrow$ linear regression for $(\alpha_m = B[m][1], d_m = d_{max}(m + 0.5)/n)$ with $m = [0, \dots, n - 1]$

Die Distanz d von zwei Punkten auf einem Kreis mit dem Radius r und mit dem Winkel α der Punktnormalen ergibt sich nach dem Kosinussatz zu

$$d(\alpha) = \sqrt{2}r\sqrt{1 - \cos \alpha}. \quad (6.32)$$

Für $\alpha \in [0, \pi/2]$ lässt sich d unter Verwendung einer Taylor-Zerlegung durch

$$d(\alpha) = r\alpha \quad (6.33)$$

annähern und dadurch effizienter berechnen. Der Algorithmus verwendet eine $n \times 2$ Matrix für jeden Abfragepunkt \mathbf{p}_q . n gibt dabei die Anzahl der Teilungsschritte des Distanzbereichs $[0; d_{max}]$ an. Typischerweise gilt $n = 5$ und d_{max} wird auf den Radius der Nachbarschaftssuche r_f gesetzt. Jede Zeile der Matrix enthält den minimalen und den maximalen Winkel für die Normalenvektoren der Punkte \mathbf{p}_q und \mathbf{p}_i im entsprechenden Distanzbereich.

Eine lineare Regression über die beiden Winkel

$$r = \frac{\sum_{i=1}^n \frac{i-0.5}{n} d_{max} \alpha_i}{\sum_{i=1}^n \alpha_i^2} \quad (6.34)$$

führt zu den Radii r_{min} und r_{max} für \mathbf{p}_q . Abbildung 6.20 zeigt schematisch den Berechnungsvorgang.

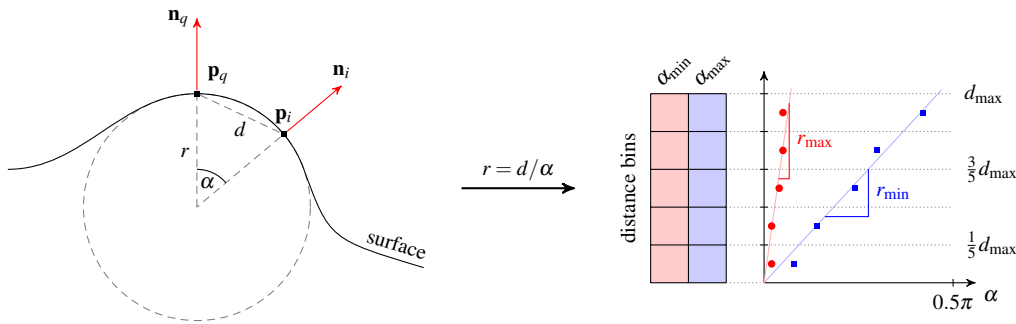


Abbildung 6.20: Der Krümmungsradius r im Punkt \mathbf{p}_q wird durch $r = d/\alpha$ angenähert. Alle Nachbarpunkte in \mathcal{P}_k werden auf eine 5×2 Matrix zugeordnet. Ein lineare Regression führt zu den Radii r_{min} und r_{max} .

6.5.2 Principal Curvatures (PC)

Das Verfahren der *Principal Curvatures* (PC) berechnet ein Maß für die lokale Oberflächengeometrie eines Punktes in Form von minimaler und maximaler Krümmung. Gleichzeitig werden auch die Richtungsvektoren der Krümmung erzeugt. Algorithmus 6 zeigt den Berechnungsablauf für den Abfragepunkt \mathbf{p}_q .

Algorithmus 6 Principal Curvatures

Require: k normals \mathbf{n}_i in neighborhood \mathcal{P}_k of \mathbf{p}_q

Ensure: max curvature c_{max} , min curvature c_{min} , max direction \mathbf{d}

transformation matrix $U \leftarrow I - \mathbf{n}_q \cdot \mathbf{n}_q^T$

for all normals \mathbf{n}_i in \mathcal{P}_k **do**

 projected normals $\mathbf{m}_i \leftarrow U \cdot \mathbf{n}_i$

end for

compute centroid $\bar{\mathbf{m}} \leftarrow 1/k \sum_{i=1}^k \mathbf{m}_i$

covariance matrix $\mathbf{C} \leftarrow 1/k \sum_{i=1}^k (\mathbf{m}_i - \bar{\mathbf{m}})(\mathbf{m}_i - \bar{\mathbf{m}})^T$

$E \leftarrow \text{eigendecomposition}(\mathbf{C})$

$\mathbf{d} \leftarrow \text{first eigenvector}(E)$

$c_{max} \leftarrow \text{first eigenvalue}(E)$

$c_{min} \leftarrow \text{second eigenvalue}(E)$

Die Krümmung c an einem Punkt \mathbf{p} einer *ebenen Kurve* ist nach (Bronstein u. Semendjajew, 2001) definiert als der reziproke Wert des Radius r des Krümmungskreises

$$c_p = \frac{1}{r_p}. \quad (6.35)$$

Diese Beziehung lässt eine große Ähnlichkeit zwischen PC und RSD erkennen. PC berechnet jedoch mehr Informationen über die Krümmung als RSD.

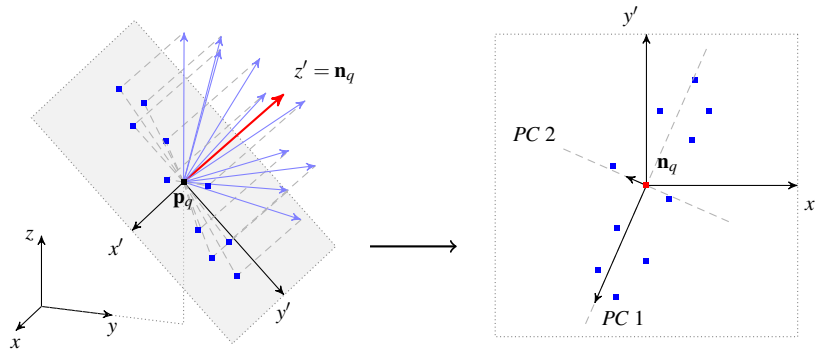


Abbildung 6.21: Projektion der lokalen Normalenvektoren auf die Tangentialebene des Abfragepunkts \mathbf{p}_q . Eine PCA führt zur minimalen und maximalen Krümmung.

Der Normalenvektor \mathbf{n}_q des Abfragepunktes \mathbf{p}_q definiert eine Tangentialebene des Punktes. Alle Normalen der Punkte in der Nachbarschaft \mathcal{P}_k werden durch

$$\mathbf{m}_i = (\mathbf{I} - \mathbf{n}_q \cdot \mathbf{n}_q^T) \cdot \mathbf{n}_i \quad (6.36)$$

auf diese Ebene projiziert, wobei \mathbf{I} die 3×3 Einheitsmatrix bezeichnet. Anschließend wird eine *Principal Component Analysis* (PCA) (vgl. Anhang C) für alle \mathbf{m}_i durchgeführt, um die Eigenwerte λ_i und Eigenvektoren \mathbf{x}_i von \mathcal{P}_k zu erhalten (siehe Abbildung 6.21). Die PCA wird über Auswertung der Kovarianzmatrix

$$\mathbf{C} = \frac{1}{k} \sum_{i=1}^k (\mathbf{m}_i - \bar{\mathbf{m}})(\mathbf{m}_i - \bar{\mathbf{m}})^T \quad (6.37)$$

mit dem Mittelwert $\bar{\mathbf{m}}$ aller \mathbf{m}_i durchgeführt. Die Eigenwerte und Eigenvektoren können durch Lösen der Gleichung

$$(\mathbf{C} - \lambda \mathbf{I})\mathbf{x} = 0 \quad (6.38)$$

gefunden werden. Das erste Wert-Vektor-Paar λ_0, \mathbf{x}_0 korrespondiert dabei mit der maximalen Krümmung c_{\max} und das zweite mit der minimalen Krümmung c_{\min} .

6.5.3 Fast Point Feature Histograms (FPFH)

Eine weitere Möglichkeit, lokale Geometrien in Punktwolken zu beschreiben, sind *Fast Point Feature Histograms* (FPFH). FPFH beschreiben die Beziehung der Normalenvektoren \mathbf{n}_i in der Nachbarschaft \mathcal{P}_k eines Abfragepunktes \mathbf{p}_q in Form von drei Winkelwerten. Die Winkelverteilung in der Nachbarschaft wird über ein Histogramm mit Einträgen für jeden Winkel und der Unterteilung jedes Winkels in Wertebereiche dargestellt. Algorithmus 7 zeigt die Implementierung des Verfahrens.

Zunächst wird für jeden Punkt \mathbf{p}_i in der Nachbarschaft \mathcal{P}_k des Abfragepunktes \mathbf{p}_q ein *Simple Point Feature Histogram* (SPFH) erzeugt. Danach wird für jedes SPFH(\mathbf{p}_i) ein neue Menge \mathcal{L}_l

Algorithmus 7 Fast Point Feature Histograms**Require:** k normals \mathbf{n}_i and position vectors \mathbf{p}_i in neighborhood \mathcal{P}_k of \mathbf{p}_q **Ensure:** histogram[33] \mathbf{H}_q $\mathcal{P}_k^+ \leftarrow \{\mathcal{P}_k, \mathbf{p}_q\}$ **for all** points \mathbf{p}_i in \mathcal{P}_k^+ **do** $\mathcal{Q}_l \leftarrow \text{nearestNeighbors}()$ of \mathbf{p}_i within r_f init histogram[33] \mathbf{S}_i **for all** points \mathbf{q}_{ij} and normals \mathbf{m}_{ij} in \mathcal{Q}_l **do**distance $\mathbf{d}_{ij} \leftarrow \mathbf{p}_i - \mathbf{o}_{ij}^l$ define source $\mathbf{p}_s, \mathbf{n}_s$ at smaller angle of $\arccos(\mathbf{n}_i \cdot \mathbf{d}_{ij})$ and $\arccos(\mathbf{m}_{ij}^l \cdot \mathbf{d}_{ij})$ define target $\mathbf{p}_t, \mathbf{n}_t$ oppositionalcreate vector frame $\mathbf{u} \perp \mathbf{v} \perp \mathbf{w}$ with $\mathbf{u} = \mathbf{n}_s$ angle $\alpha_{ij} \leftarrow \mathbf{v} \cdot \mathbf{n}_t$ angle $\varphi_{ij} \leftarrow \mathbf{u} \cdot (\mathbf{p}_t - \mathbf{p}_s) / (\|\mathbf{p}_t - \mathbf{p}_s\|)$ angle $\sigma_{ij} \leftarrow \text{atan2}(\mathbf{w} \cdot \mathbf{n}_t, \mathbf{u} \cdot \mathbf{n}_t)$ update \mathbf{S}_i according $(\alpha_{ij}, \varphi_{ij}, \sigma_{ij})$ **end for** $\mathbf{p}_i \leftarrow \{\mathbf{p}_i, \mathbf{S}_i\}$ **end for** $\mathbf{H}_q \leftarrow \mathbf{S}_q + 1/k \sum_{i=1}^k (\mathbf{S}_i / w_i)$

von l benachbarten Punkten \mathbf{q}_{ij} und ihren Normalenvektoren \mathbf{m}_{ij} generiert. Die Punkte werden in einem Merkmalsradius r_f ausgewählt. Für jedes Punktpaar $(\mathbf{p}_i, \mathbf{q}_{ij})$ wird nachfolgend ein Quellpunkt \mathbf{p}_s so gewählt, dass der Winkel φ zwischen dem Normalenvektor von \mathbf{p}_s und der Verbindungslinie zwischen den beiden Punkten minimal wird. Der andere Punkt wird als Zielpunkt \mathbf{p}_t bezeichnet. Anschließend wird ein Darboux-Koordinatensystem $\mathbf{u} \perp \mathbf{v} \perp \mathbf{w}$ mit dem Ursprung \mathbf{p}_s und den orthogonalen Vektoren

$$\mathbf{u} = \mathbf{n}_s \quad (6.39)$$

$$\mathbf{v} = \mathbf{u} \times \frac{\mathbf{p}_t - \mathbf{p}_s}{\|\mathbf{p}_t - \mathbf{p}_s\|} \quad (6.40)$$

$$\mathbf{w} = \mathbf{u} \times \mathbf{v} \quad (6.41)$$

definiert (Abbildung 6.22). Ein Darboux-Koordinatensystem bezeichnet ein Koordinatensystem an einem Punkt einer Kurve, wobei die Einheitsvektoren aus dem Einheitstangentialvektor, dem Einheitsnormalenvektor und dem Kreuzprodukt dieser beiden Vektoren gebildet werden. Zusammen mit dem Quellpunkt, dem Zielpunkt und dem Koordinatensystem können die drei Merkmalswerte wie folgt berechnet werden. Der erste Wert

$$f_1 = \cos(\alpha) = \mathbf{v} \cdot \mathbf{n}_t \quad (6.42)$$

misst den Winkel zwischen dem Normalenvektor \mathbf{n}_t und dem Darboux-Vektor \mathbf{v} . Der zweite

$$f_2 = \cos(\varphi) = \mathbf{u} \cdot \frac{\mathbf{p}_t - \mathbf{p}_s}{\|\mathbf{p}_t - \mathbf{p}_s\|} \quad (6.43)$$

bezeichnet den Winkel zwischen dem Normalenvektor \mathbf{n}_s und dem Differenzvektor zwischen Quell- und Zielpunkt. Der dritte Wert

$$f_3 = \sigma = \text{atan2}(\mathbf{w} \cdot \mathbf{n}_t, \mathbf{u} \cdot \mathbf{n}_t) \quad (6.44)$$

stellt den Winkel zwischen der Projektion von \mathbf{n}_t auf die Ebene (\mathbf{u}, \mathbf{w}) und dem Darboux-Vektor \mathbf{u} dar. Aus den Winkelwerten für alle Punkte in \mathcal{Q}_l wird nun ein Histogramm mit 11 Klassen je Winkel erzeugt. Die Winkelwerte werden vor dem Eintragen mit $100/l$ normalisiert. Das resultierende Histogramm mit 33 Klassen stellt das SPFH in einem Punkt dar.

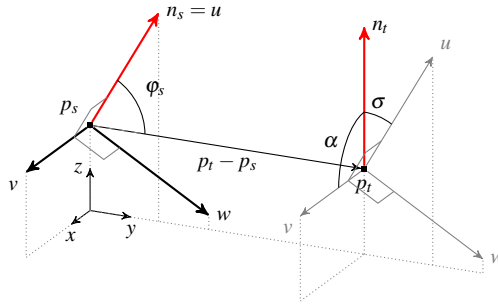


Abbildung 6.22: Darboux Koordinatensystem

In einem zweiten Verarbeitungsschritt wird aus den SPFHs ein FPFH generiert. Dazu werden alle k SPFHs in \mathcal{P}_k betrachtet und das FPFH gemäß

$$\text{FPFH}(p_q) = \text{SPFH}(p_q) + \frac{1}{k} \sum_{i=1}^k \frac{1}{w_i} \cdot \text{SPFH}(\mathbf{p}_i) \quad (6.45)$$

mit dem Gewicht w_i erzeugt. Das Gewicht ist definiert als

$$w_i = \|\mathbf{p}_q - \mathbf{p}_i\|. \quad (6.46)$$

6.5.4 Klassifizierung von Oberflächenpunkten

Nachfolgend zur Extraktion müssen die Merkmalspunkte klassifiziert werden, um geometrische Information aus der Punktwolke zu gewinnen. Die Methoden zur Klassifizierung sind für verschiedene Merkmalstypen unterschiedlich.

Für RSD und PC können die Merkmale durch Überprüfung von Schwellwerten für r_{\min} und r_{\max} (RSD) bzw. c_{\min} und c_{\max} (PC) klassifiziert werden. Abbildung 6.23 zeigt die Klassifizierungsgebiete für beide Merkmalstypen. Unterscheidbare Klassen sind hierbei Kanten (E), Ecken (Co), Zylinder (Cy), Kugeln (S) und Ebenen (P). Dabei wird r_{\min} (c_{\max}) als Schwelle zwischen Kanten,

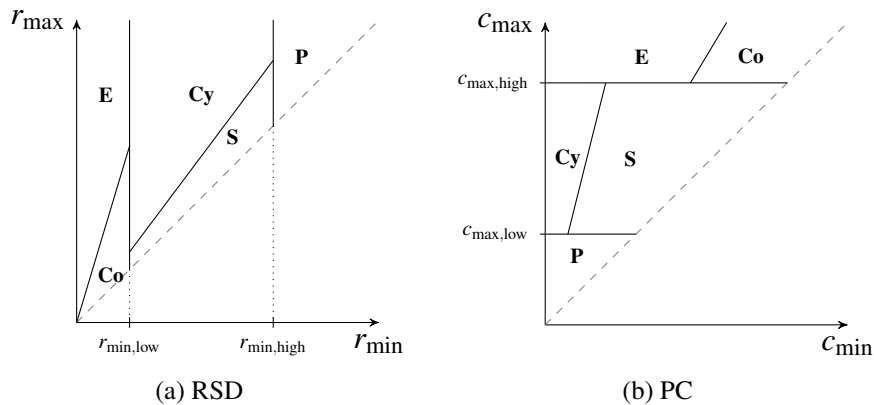


Abbildung 6.23: Klassifizierungsbereiche abhängig von Radius bzw. Krümmung für RSD und PC

gekrümmten Flächen und Ebenen verwendet, wohingegen die Unterscheidung zwischen Zylinder und Kugel bzw. Kanten und Ecken auf dem Verhältnis zwischen r_{\min} und r_{\max} (c_{\max} und c_{\min}) basiert. Diese Schwellen werden im Folgenden mit $x_{(Cy,S)}$ und $x_{(E,Co)}$ bezeichnet. Es ist zu sehen, dass aufgrund der mathematischen Verwandtschaft von RSD und PC auch die Klassifizierung ähnlich verläuft.

Da der Deskriptor für die FPFH-Merkmale hochdimensional ist, gelingt bei diesem Merkmalstyp eine Klassifizierung auf Schwellwertbasis nicht. Stattdessen wird eine *Support Vector Machine* (SVM) verwendet. Diese muss mit geeigneten Daten trainiert werden, um zwischen verschiedenen Geometrien unterscheiden zu können. Typischerweise erfolgt das Training auf synthetischen Daten, die mit unterschiedlich starkem Rauschen belegt sind (vgl. (Fuchs, 2011)).

6.5.5 Evaluierung

Die Evaluierung der drei Merkmalstypen in Bezug auf Klassifizierungsgenauigkeit und Rechenzeit erfolgt anhand von 8 Innenraumszenen, welche mit einer RDG-D Kamera aufgenommen wurden. Der Datensatz unterteilt sich in Szenen für den Nahbereich (bis zu 1,8 m Entfernung) und für den Fernbereich (bis zu 3,0 m Entfernung). Beispiele für die Szenen sind in Abbildung 6.6 zu sehen, eine genaue Beschreibung des Datensatzes findet sich in Anhang B. Es wird nicht nur die Genauigkeit bei Betrachtung aller fünf Klassen untersucht, sondern auch der Fall, dass nur Ebenen, Kanten und komplexere Oberflächen unterschieden werden sollen. Dieser Anwendungsfall ist speziell für weiter entfernte Szenen oder Situationen geeignet, in denen vorzugsweise Ebenen detektiert werden sollen.

Die Genauigkeit der Klassifizierung wird anhand der *micro-average* Werte R_{mic} und der *macro-average* Werte R_{mac} , P_{mac} und F_{mac} bewertet (siehe Anhang A). Tabelle 6.4 zeigt die unterschiedliche Parametereinstellung der Verfahren für die Nahbereichs- und Fernbereichsszenen. Es werden je nach Entfernung unterschiedliche Parametersätze verwendet. Einstellbar sind die Suchra-

dien für Nachbarpunkte bei der Normalenberechnung r_n und bei der Merkmalsberechnung r_f und die Schwellwerte für die Klassifizierung r_{min} , c_{min} und x .

	Nahbereich			Fernbereich		
	RSD	PC	FPFH	RSD	PC	FPFH
$r_n(m)$	0,03	0,03	0,03	0,045	0,045	0,050
$r_f(m)$	0,03	0,03	0,055	0,045	0,045	0,070
$r_{min,low}(m)$	0,035	0,02		0,09	0,12	
$r_{min,high}(m)$	0,08	0,09		0,09	0,12	
$x_{(Cy,S)}$	4,75	7,0		4,75	7,0	
$x_{(E,Co)}$	3,5	2,75		3,5	2,75	

Tabelle 6.4: Parameter für jedes untersuchte Verfahren für die beiden Distanzkategorien

Die Ergebnisse der Untersuchung der Klassifizierungsgenauigkeit werden in Tabelle 6.5 gezeigt. Es ist zu erkennen, dass FPFH besonders bei der Klassifizierung vieler kleiner Objekte in den Nahbereichsszenen besser abschneidet als die anderen Deskriptoren. Insgesamt gesehen zeigt jedoch PC die höchste Genauigkeit über alle Szenen. Die Klassifizierungseigenschaften von RSD sind aufgrund der algorithmischen Gemeinsamkeiten ähnlich, jedoch insgesamt gesehen etwas schlechter, was auf die höhere Rauschanfälligkeit zurückzuführen ist. Bei der Auswertung von nur drei Klassen (3c) ist zu beobachten, dass die Klassifizierungsergebnisse für alle Deskriptoren besser werden. Dies ist darauf zurückzuführen, dass Ebenen und Kanten leichter klassifiziert werden können als Zylinder oder Kugeln. Der Grund dafür ist, dass Ebenen und Kanten jeweils an einem Ende des Klassifizierungsbereichs liegen und dadurch die Parameter besser eingestellt werden können. Weiterhin sind Ebenen meist größere Objekte, so dass die Rauschanfälligkeit auch bei größeren Entfernungen geringer ist.

Tabelle 6.6 zeigt beispielhaft die Oberflächenklassifizierung an vier Szenen. Es ist zu erkennen, dass PC speziell bei den Fernbereichsszenen die Ebenen gut klassifizieren kann. FPFH ist dagegen in der Lage, kleine Objekte im Nahbereich genauer zu bestimmen.

Abbildung 6.24 zeigt den Vergleich der Verarbeitungszeiten für die Merkmalstypen. Es ist zu sehen, dass PC und RSD ungefähr gleich schnell sind, wohingegen die Rechenzeit für FPFH deutlich länger ist. Der Suchradius für Nachbarpunkte r_n hat wenig Auswirkung auf die Rechenzeit.

Generell ist zu sagen, dass die Verfahren auf dem Stand der Technik für die Merkmalserkennung im Rahmen dieser Arbeit ausreichen und somit für diesen Schritt keine neue Methode entwickelt werden muss. Bei Betrachtung der Ergebnisse für Rechenzeit und Genauigkeit erscheint PC als am besten für die 3-D-Umgebungserfassung geeignet. RSD schneidet bei allen Kategorien schlechter ab. FPFH kommt sowohl wegen der Ausführungsgeschwindigkeit als auch wegen des Trainingsaufwands für die Klassifizierung nicht in Frage, obwohl die Detailklassifizierung besser ist.

Szene	R_{mic}			R_{mac}			P_{mac}			F_{mac}		
	RSD	PC	FPFH	RSD	PC	FPFH	RSD	PC	FPFH	RSD	PC	FPFH
kitchen close (3c) ^a	0,613	0,757	0,830	0,703	0,702	0,809	0,455	0,517	0,586	0,553	0,596	0,679
kitchen close	0,594	0,738	0,816	0,491	0,499	0,615	0,327	0,364	0,446	0,392	0,421	0,517
kitchen far (3c)	0,458	0,667	0,586	0,616	0,735	0,562	0,397	0,443	0,398	0,483	0,553	0,466
kitchen far	0,454	0,659	0,583	0,479	0,469	0,394	0,302	0,342	0,299	0,370	0,396	0,340
table close (3c)	0,621	0,701	0,704	0,723	0,744	0,764	0,539	0,572	0,586	0,617	0,647	0,663
table close	0,590	0,661	0,676	0,608	0,595	0,632	0,410	0,389	0,475	0,490	0,471	0,543
table far (3c)	0,744	0,819	0,481	0,621	0,726	0,517	0,439	0,495	0,421	0,515	0,589	0,464
table far	0,734	0,804	0,476	0,478	0,570	0,378	0,285	0,317	0,306	0,357	0,408	0,338
office close (3c)	0,686	0,813	0,718	0,645	0,696	0,519	0,442	0,498	0,419	0,524	0,581	0,464
office close	0,682	0,801	0,713	0,496	0,462	0,403	0,342	0,381	0,322	0,405	0,417	0,358
office far (3c)	0,471	0,636	0,536	0,537	0,574	0,635	0,404	0,439	0,432	0,461	0,497	0,514
office far	0,462	0,628	0,529	0,494	0,558	0,513	0,314	0,347	0,332	0,384	0,428	0,403
cupboard close (3c)	0,598	0,666	0,634	0,625	0,665	0,567	0,459	0,482	0,441	0,529	0,559	0,496
cupboard close	0,586	0,643	0,617	0,477	0,452	0,414	0,353	0,369	0,330	0,406	0,407	0,368
cupboard far (3c)	0,571	0,682	0,721	0,658	0,706	0,576	0,473	0,517	0,459	0,551	0,597	0,511
cupboard far	0,559	0,668	0,708	0,553	0,601	0,514	0,391	0,448	0,374	0,458	0,514	0,433
Mittelwert	0,583	0,700	0,640	0,510	0,526	0,483	0,341	0,370	0,361	0,408	0,433	0,413
Mittelwert (3c)	0,595	0,718	0,651	0,641	0,694	0,619	0,451	0,495	0,468	0,529	0,577	0,532

am schlechtesten
 mittel
 am besten

^a (3c) steht für die Evaluierung mit drei Klassen

Tabelle 6.5: Ergebnisse der Genauigkeitsuntersuchung nach Szenen

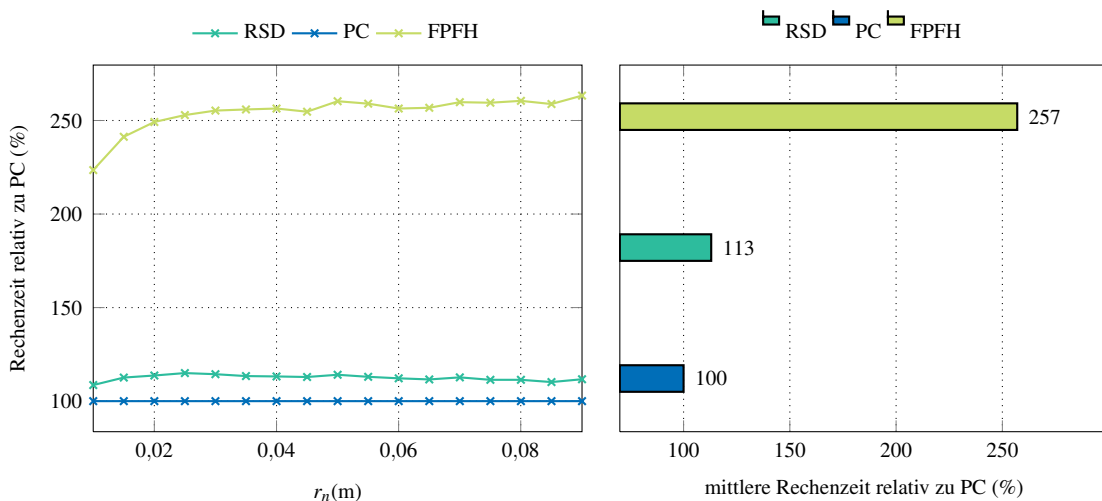


Abbildung 6.24: Rechenzeit für die Merkmalsextraktoren

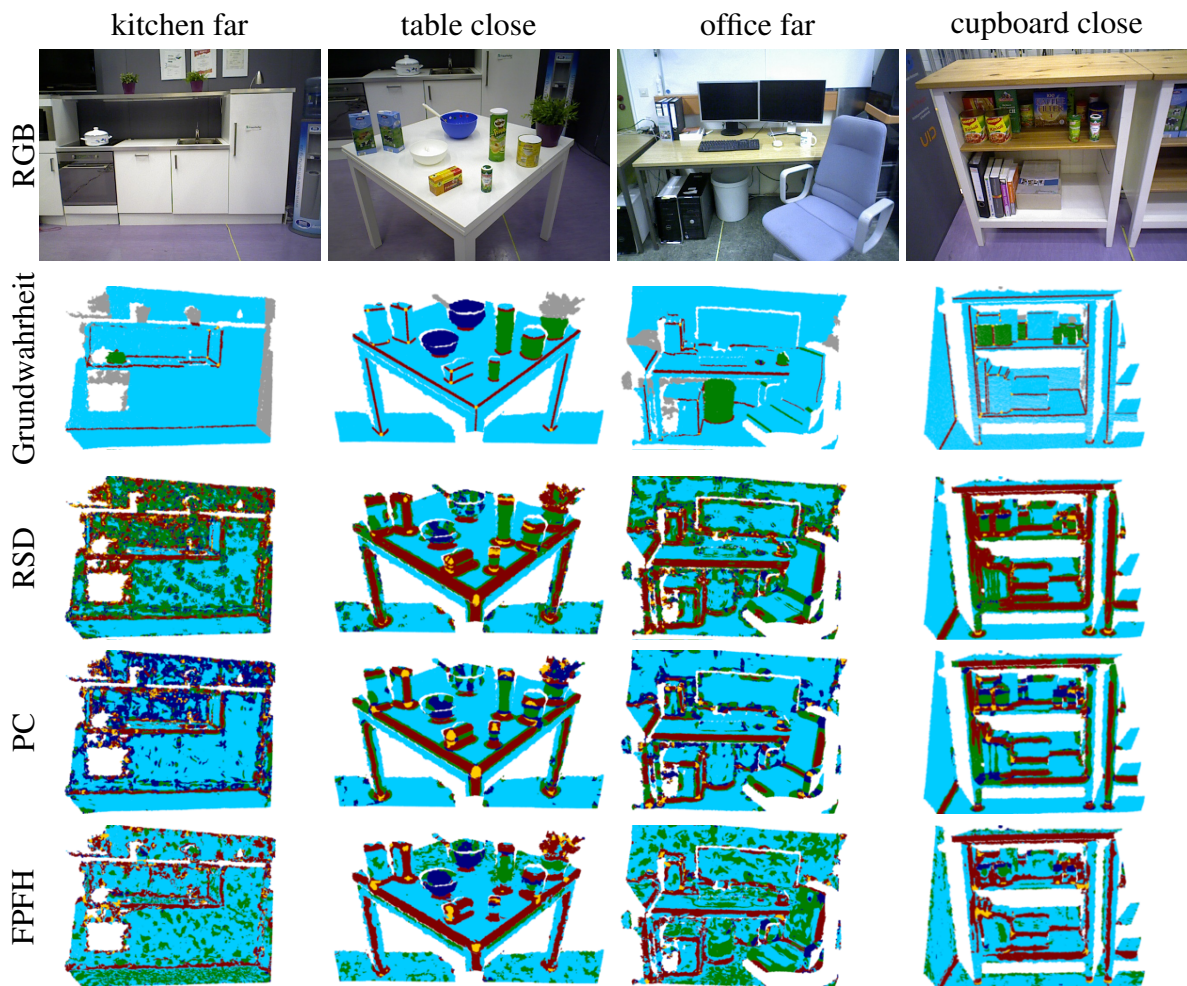


Tabelle 6.6: Beispielszenen für die Merkmalsextraktion

6.6 Segmentierung

Ziel der Segmentierung ist die Unterteilung einer Punktwolke in Regionen mit identischen geometrischen Eigenschaften und deren Klassifikation. Aus dem Stand der Technik in Abschnitt 4.3.7 geht hervor, dass vor allem RANSAC-basierte Verfahren als auch *Region Growing* vielversprechende Ansätze sind.

Im Folgenden werden zwei im Rahmen dieser Arbeit entwickelte Verfahren vorgestellt und verglichen. Dies ist zum einen eine auf RANSAC basierende Ebenenextraktion und zum anderen eine Segmentierung durch *Region Growing*, bei der Punktnormalen als Segmentgrenzen und die in Abschnitt 6.5 vorgestellten Punktmerkmale zur Klassifikation verwendet werden. Die beiden Verfahren werden gegenüber dem Segmentierungsverfahren *Multi Plane Segmentation* (MPS) verglichen.

6.6.1 Ebenensegmentierung mit RANSAC (ERS)

Im Rahmen dieser Arbeit wird ein sequentielles Verfahren zur Ebenenextraktion vorgestellt, welches iterativ die jeweils dominante Ebene in einer Punktwolke findet, extrahiert und aus der Punktwolke entfernt. Das Verfahren verwendet RANSAC zur Extraktion von Ebenen aus Punktwolken. Dabei wird der Ansatz von Rusu (Rusu u. a., 2009a) um einen Vorverarbeitungsschritt erweitert, um die fehlende Modelltreue der RANSAC-basierten Verfahren zu kompensieren (siehe Algorithmus 8).

Algorithmus 8 ERS

Require: Point cloud $P = \{\mathbf{p}_i\}$

Ensure: Planar hull polygons

$C \leftarrow$ Cluster P

for all c_i **do**

Fit plane

Project inliers

Calculate concave hull

end for

Zunächst wird eine Punktwolke in räumlich zusammenhängende Bereiche unterteilt. Demnach gehören Punkte $\mathbf{p}_i, \mathbf{p}_j$ genau dann zum selben Bereich, wenn die Bedingung

$$\max \|\mathbf{p}_i - \mathbf{p}_j\|_2 \leq d_{th} \quad (6.47)$$

mit der Distanzschwelle d_{th} erfüllt ist. Das Ergebnis sind Unterbereiche (*Cluster*) c_i der Punktwolke, die nachfolgend einzeln untersucht werden können. Durch diesen Schritt gelingt es, den Suchraum für das Einpassen des Ebenenmodells zu verkleinern, was die Wahrscheinlichkeit erhöht, einen korrekten Einpassvorgang durchzuführen.

In jeden *Cluster* c_i wird im nächsten Schritt versucht, ein parametrisches Ebenenmodell

$$e : ax + by + cz + d = 0 \quad (6.48)$$

einzupassen (siehe Anhang D). Um das zu erreichen, wird RANSAC verwendet. Dieser Algorithmus zieht so viele zufällige Stichproben aus der Punktwolke, wie zur exakten Bestimmung einer Ebene notwendig sind, legt eine Ebene durch diese Punkte und vergleicht den Anteil der Punktwolke, der in dieser Ebene liegt (*Inliers*) mit dem Anteil, der nicht zur Ebene gehört (*Outliers*). Das Ziehen der Stichprobe wird solange wiederholt, bis die Ebene mit den meisten *Inliers* gefunden wurde. Einstellbare Parameter sind die maximale Anzahl von Iterationen i_{max} und die Toleranzschwelle d_a , welche über zum Modell gehörige und Ausreißer entscheidet. Das Verfahren liefert sowohl die Ebenenparameter a, b, c und d als auch die Indizes der *Inliers* zurück. Wird eine Ebene im *Cluster* gefunden, werden die zugehörigen Punkte aus der Wolke entfernt und die Methode erneut angewendet, um die nächste Ebene einzupassen. Dies wird so oft wiederholt, bis keine Ebene mehr gefunden wird oder eine minimale *Clustergröße* erreicht ist.

Um eine geometrische Beschreibung der Flächen zu erhalten, wird im nächsten Schritt eine konkave Hülle gesucht, die zusammen mit den Ebenenparametern eine vollständige geometrische

Repräsentation darstellt und außerdem das Datenvolumen drastisch reduziert. Hierzu werden zunächst alle zur Ebene gehörigen Punkte gemäß

$$\mathbf{p}_{pr} = \mathbf{p} - \delta \mathbf{n}. \quad (6.49)$$

in die Ebene projiziert. Dabei ergibt sich δ mit der Normalform

$$e_n : \mathbf{n} \cdot (\mathbf{x} - \mathbf{a}) = 0 \quad (6.50)$$

der Ebene zu

$$\delta = \mathbf{n} \cdot (\mathbf{p} - \mathbf{a}). \quad (6.51)$$

Der Normalenvektor der Ebene folgt aus (6.48) zu

$$\mathbf{n} = (a \ b \ c)^T. \quad (6.52)$$

Durch die Projektion ist die Berechnung einer konkaven Hülle auf ein zweidimensionales Problem reduziert. Zur Bestimmung der konkaven Hülle werden α -shapes (vgl. Abschnitt 4.3.8) verwendet. α -shapes nutzen die Delauney-Triangulation als Ausgangspunkt und nähern die geometrische Form einer Punktmenge an. Dabei können sowohl konvexe oder konkave Formen als auch Formen mit Löchern entstehen. Der zugehörige Radius α variiert dabei die Granularität der zu erzeugenden konkaven Hülle. Ein großer Wert führt zu gröbereren Formen. Abbildung 6.25 veranschaulicht die Verarbeitungskette anhand einer Küchenszene.

6.6.2 Graph-basiertes Region Growing und Punktklassifikation (GRG)

Das zweite im Rahmen dieser Arbeit entwickelte Verfahren verbessert den Stand der Technik in zweierlei Hinsicht. Zum einen wird ein zusätzlicher Optimierungsschritt vorgesehen, der Über- und Untersegmentierung reduziert. Zum anderen können nicht nur Ebenen segmentiert werden, sondern auch andere geometrische Basisformen. GRG nutzt die schnelle Verarbeitung von geordneten Punktwolken. Basierend auf den Punktnormalen werden Kanten gesucht. Diese werden nachfolgend als Segmentgrenzen in einem *Region Growing* Schritt verwendet. Die entstehenden Segmente werden in einem weiteren Schritt über ein auf Graphen basierendes Verfahren optimiert, bevor sie über Punktdeskriptoren geometrisch klassifiziert werden. Der Berechnungsablauf des Algorithmus ist in Algorithmus 9 dargestellt.

Normalen-basiertes Region Growing

Zunächst müssen für alle Punkte der Punktwolke die lokalen Normalenvektoren berechnet werden. Da als Eingangsdaten ausschließlich Tiefenbilder verwendet werden, kann die Berechnung der Punktnormalen wie in Kapitel 6.1 beschrieben erfolgen. Im nächsten Schritt werden zusammengehörige Segmente erzeugt, wobei *Region Growing* zum Einsatz kommt. Das Segmentwachstum erfolgt wie beim Specklefilter durch Vorantreiben einer Wellenfront (vgl. Abschnitt

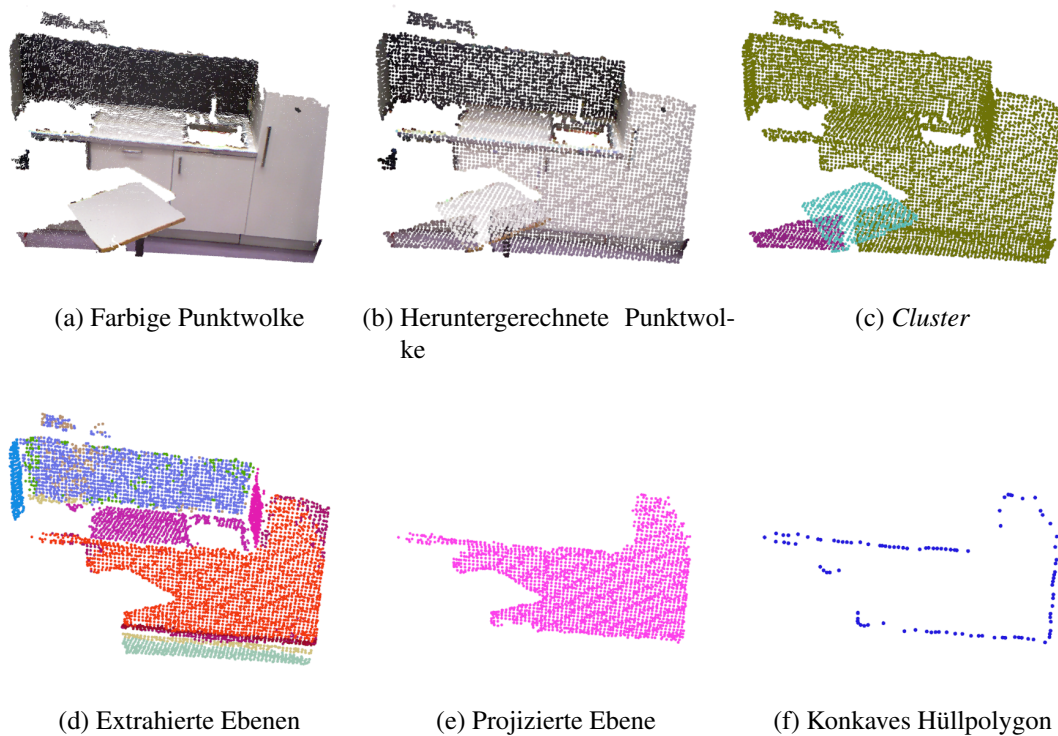


Abbildung 6.25: Verarbeitungskette für die Ebenenextraktion

6.3.2). Während des Wachstums eines Segments wird der mittlere Normalenvektor $\bar{\mathbf{n}}$ über alle bisher erfassten Punkte berechnet. Als Abbruchkriterium wird der Winkel

$$\alpha = \arccos \left(\frac{\bar{\mathbf{n}} \cdot \mathbf{n}_i}{\|\bar{\mathbf{n}}\| \|\mathbf{n}_i\|} \right) \quad (6.53)$$

zwischen den normierten Normalen \mathbf{n}_i und $\bar{\mathbf{n}}$ betrachtet. Überschreitet dieser einen Schwellwert α_{th} wird das Wachstum am Punkt p_i beendet. Von allen neuen Segmentpunkten werden zunächst die verarbeitet, für die α minimal ist. Dadurch wächst ein Segment zunächst in den homogenen Bereichen und erst danach in die Randbereiche, was die Segmentierung robuster macht. Die Vorgehensweise mit Normalenbetrachtung führt für gewöhnlich zu einer Übersegmentierung der Punktwolke, da gerade bei gekrümmten Flächen der Mittelwert $\bar{\mathbf{n}}$ nach kurzem Segmentwachstum deutlich von den Punktnormalen abweicht. Dies ließe sich mit einem laufenden Mittelwert korrigieren, was jedoch implementierungstechnisch nicht sinnvoll ist, da es zu einer deutlich ineffizienteren Berechnung führt.

Aus diesem Grund, wird bereits während des *Region Growing* eine Graphenstruktur aufgebaut. Diese besteht aus Segmenten S als Knoten, welche über die geometrischen Kanten E als Graphkanten miteinander verbunden sind (siehe Abbildung 6.26). Jede Kante E_{jk} verbindet zwei Segmente S_j, S_k und besitzt eine Menge aus Grenzpunktpaaren $B = \{\langle \mathbf{b}_j, \mathbf{b}_k \rangle\}$, welche sich aus den benachbarten Segmenten ergeben. Der Graph ermöglicht es, nebeneinander liegende Segmente zu finden und die Verbindungskanten genauer zu analysieren.

Algorithmus 9 GRG**Require:** Ordered point cloud $P = \{\mathbf{p}_i\}$ with c columns and r rows**Ensure:** Surface class for each segment S **for all** \mathbf{p}_i **do** Calculate point normal \mathbf{n}_i **end for****procedure** GROW SEGMENTS(P) Define neighborhood of point $\mathbf{p}_{i,j}$ as $\mathcal{P}_{i,j} = \{\mathbf{p}_{i,j-1}, \mathbf{p}_{i,j+1}, \mathbf{p}_{i-1,j}, \mathbf{p}_{i+1,j}\}$ Label l_p of point $\mathbf{p}_{i,j}$ **for** $i = 1$ to c **do** **for** $j = 1$ to r **do** **if** $\nexists l_p$ **then** wavefront $w \leftarrow \mathbf{p}_{i,j}$ $l_p \leftarrow$ new label **end if** **while** w is not empty **do** Compute average normal $\bar{\mathbf{n}}$ **for all** $\mathbf{q} \in \mathcal{P}_{i,j}$ **do** **if** $\nexists l_q$ and $\arccos\left(\frac{\bar{\mathbf{n}} \cdot \mathbf{n}_i}{\|\bar{\mathbf{n}}\| \|\mathbf{n}_i\|}\right) < \alpha_{th}$ **then** $w \leftarrow \{w, \mathbf{q}\}$ $l_q \leftarrow l_p$ **end if** **end for** $\mathbf{p}_{i,j} \leftarrow w(\text{end})$ **end while** **end for** **end for** Extend Graph G **end procedure****procedure** REFINE SEGMENTATION(S, G) Use G to determine adjacent segments S_j, S_k Compute smoothness s of border point pairs $\langle \mathbf{b}_j, \mathbf{b}_k \rangle$ **if** $s > s_{th}$ **then** Merge S_j, S_k **end if****end procedure****procedure** CLASSIFY(S) **for all** $S_i \in S$ **do**

Compute PCA over point normals

Compute PC on non-planar segments

Apply label according to dominant class

end for**end procedure**

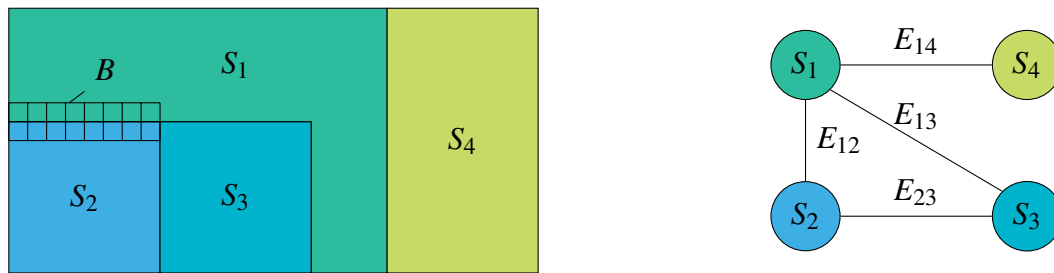


Abbildung 6.26: Beispiel einer Graphenstruktur für vier Segmente

Optimierung

Ziel des Optimierungsschrittes ist die Fusion von sich berührenden Segmenten mit gleichen geometrischen Eigenschaften. Hierzu wird überprüft, ob der Übergang zwischen zwei Segmenten stetig ist. Für zwei Segmente S_j und S_k werden die Grenzpunktepaare B betrachtet. Für jedes Punktepaar $b = \langle \mathbf{b}_j, \mathbf{b}_k \rangle$ erfolgt die Neuberechnung der Punktnormalen, wobei nur Nachbarpunkte im selben Segment betrachtet werden, um Einfluss der Nachbarsegmente auszuschließen. Danach wird analog zu (6.53) das normierte Skalarprodukt zwischen den Normalenvektoren für jedes Paar berechnet und bei der Erfüllung eines Schwellwertkriteriums der Menge stetiger Grenzpunkte

$$B_s = \left\{ b \mid \frac{\mathbf{n}_j \mathbf{n}_k}{\|\mathbf{n}_j\| \|\mathbf{n}_k\|} > d_{th} \right\} \quad (6.54)$$

hinzugefügt. Über den Schwellwert s_{th} kann dann die Segmentgrenze gemäß

$$\|B_s\| > s_{th} \quad (6.55)$$

als stetig oder unstetig klassifiziert werden. Werden genügend viele Punktepaare als stetig klassifiziert, wird angenommen, dass die gesamte Kante einen stetigen Übergang darstellt. In diesem Fall werden die beiden benachbarten Segmente vereint.

Oberflächenklassifikation

Für eine geometrische Beschreibung müssen die Segmente klassifiziert werden. Dafür werden zunächst große Segmente betrachtet, da es sich dabei in typischen Innenraumszenen meistens um Ebenen handelt. Für diese Segmente wird eine PCA (vgl. Anhang C) über die Punktnormalen durchgeführt und geprüft, ob es sich tatsächlich um Ebenen handelt. Es werden die Verhältnisse der Eigenwerte λ_1/λ_2 und λ_1/λ_3 berechnet. Liegen diese über einem Schwellwert, ist das Segment eine Ebene. Für alle übrigen Segmente kommt PC aus Abschnitt 6.5.2 zum Einsatz. Im Unterschied dazu wird nun jedes Segment separat betrachtet, was zu einer höheren Klassifizierungsgenauigkeit führt. Die Randpunkte der Segmente werden nicht betrachtet, da diese weniger Aussagekraft über die Segmentgeometrie haben. Die Klasse eines Segments ergibt sich aus

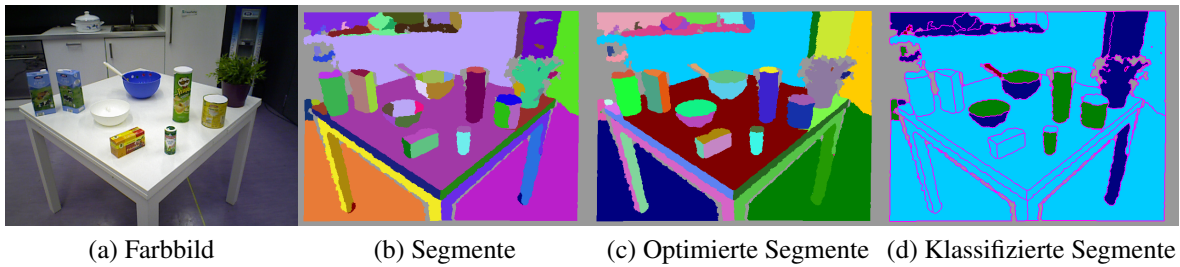


Abbildung 6.27: Verarbeitungskette für die Segmentierung und Klassifikation

der dominanten Klasse für die Segmentpunkte. Abbildung 6.27 zeigt beispielhaft die Verarbeitungsschritte anhand einer Tischszene. Man kann gut erkennen, dass besonders die gekrümmten Flächen der Schalen und Dosen zunächst übersegmentiert sind. Im Optimierungsschritt werden diese Segmente dann zusammengeführt und anschließend klassifiziert. Dabei stellen die hellblauen Segmente Ebenen dar, die grünen zylindrische Objekte und die dunkelblauen komplexere Strukturen.

Beschreibung der Segmente

Für jedes klassifizierte Segment wird anschließend ein konkaves Hüllpolygon berechnet, welches in der geometrischen Karte verarbeitet werden kann. Dazu werden die Randpunkte über eine Maske verfolgt und somit die Kontur des Segments bestimmt. In einem weiteren Schritt werden Zwischenpunkte entfernt, um die Kontur auszudünnen. Anders als bei ERS sind die Parameter der Basisgeometrien nach der Klassifikation nicht intrinsisch bekannt. Deswegen müssen diese aus den Segmenteigenschaften abgeleitet werden.

Für Ebenensegmente gilt es, die Ebenenparameter in (6.48) zu finden. Nach (6.52) bedeutet dies, dass der Normalenvektor \mathbf{n} und der Abstandsparameter d bestimmt werden müssen. Die bereits durchgeführte PCA über die Punktnormalen liefert die Normale (siehe Anhang C). Für die Eigenwerte gilt $\lambda_1 > \lambda_2 > \lambda_3$. Der zum größten Eigenwert λ_1 gehörige Eigenvektor \mathbf{v}_1 ist identisch mit der Normalen der Ebene, d. h.

$$\mathbf{n} = \mathbf{v}_1. \quad (6.56)$$

Der Abstandsparameter d ergibt sich aus \mathbf{n} und dem Flächenschwerpunkt \mathbf{c} des Segments zu

$$d = \mathbf{n} \cdot \mathbf{c}. \quad (6.57)$$

Da es sich bei Zylindersegmenten um 3-D-Strukturen handelt, gestaltet sich die Ableitung der Parameter komplizierter als bei den Ebenensegmenten. Ein Zylinder wird eindeutig durch den Radius r , die Höhe h und ein Koordinatensystem K_z , welches seine Lage im Raum bestimmt, beschrieben. Die z -Achse \mathbf{z}_k des Koordinatensystems soll in der Symmetrieachse des Zylinders liegen und die x -Achse \mathbf{x}_z zum Flächenschwerpunkt \mathbf{c} der Hülle zeigen (siehe Abbildung 6.28).

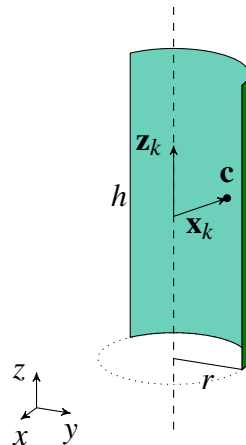


Abbildung 6.28: Notwendige Parameter zur Beschreibung eines Zylindersegments

\mathbf{z}_k lässt sich aus den Ergebnissen der PC-Berechnung ableiten, da diese auch die Krümmungsrichtung liefert. Die Symmetrieachse verläuft immer in Richtung der kleinsten Krümmung des Segments. Um diese zu bestimmen, wird eine PCA über die Krümmungen aller Segmentpunkte durchgeführt. Danach wird \mathbf{x}_k so bestimmt, dass sie orthogonal zu \mathbf{z}_k ist und in Richtung \mathbf{c} zeigt, womit K_z eindeutig bestimmt ist.

Um r zu berechnen, werden alle Segmentpunkte nach K_z transformiert und nur die x - und y -Werte betrachtet, wodurch sich ein Kreissegment bildet. In dieses kann mit RANSAC ein Kreismodell eingepasst werden. Der Radius des Kreises entspricht dem des Zylinders.

Schließlich wird $h = \{h_{min}, h_{max}\}$ aus der Symmetrieachse und den Segmentpunkten \mathbf{p}_i abgeleitet. In K_z ergibt sich so

$$h_{max} = \max \mathbf{p}_{i,z} \quad (6.58)$$

$$h_{min} = \min \mathbf{p}_{i,z}. \quad (6.59)$$

Liegt eine farbige Punktwolke vor, kann der dominante Farbwert aus allen Punkten abgeleitet und dem Segment zugewiesen werden. Da eine einfache Mittelwertbildung im RGB-Farbraum zu beliebig falschen Farben führen kann, wird eine histogrammbasierte Methode im HSV-Farbraum entwickelt. Zunächst werden die als RGB vorliegenden Farbwerte nach HSV gewandelt (Szeli-ski, 2011). Danach wird ein Histogramm für die H-Werte erstellt. Die Klasse mit den meisten Einträgen im H-Histogramm bestimmt den dominanten Farbton. Der Mittelwert der Sättigungswerte aller zur dominanten H-Klasse gehörigen Punkte ergibt den mittleren S-Wert. Der durchschnittliche V-Wert (Helligkeit) hingegen wird über den Mittelwert aller V-Werte berechnet. Die resultierende Farbe wird schließlich in den RGB-Raum zurücktransformiert.

6.6.3 Evaluierung

In diesem Abschnitt werden ERS, GRG gegenüber einem dem Stand der Technik entsprechenden Verfahren (MPS) evaluiert. Für die Evaluierung kommen wieder die acht Innenraumszenen zum

Einsatz, die bereits in Abschnitt 6.5 verwendet wurden (siehe auch Anhang B). Für ERS wird $d_{th} = 0,06$ m, $d_a = 0,04$ m, α auf 0,05 m und der Radius für die Normalenberechnung auf 0,1 m gesetzt. Für GRG wird die minimale Segmentgröße auf 150 Punkte gesetzt, da kleinere Segmente nicht ausreichend genau erkannt werden können. Für den RG-Schwellwert gilt $\alpha_{th} = 0,611$ rad und für den Stetigkeitsschwellwert $s_{th} = 0,873$ rad.

Für die Evaluierung der Genauigkeit wird zunächst GRG untersucht. Da es sich aus den Schritten Segmentierung und Klassifizierung zusammensetzt, werden diese zunächst getrennt bewertet. Danach wird die kombinierte Genauigkeit beider Schritte evaluiert. In Tabelle 6.7 ist die Genauigkeit der Segmentierung bei GRG zu sehen. Für alle Szenen wurde als Grundwahrheit die Anzahl von Segmenten gezählt. Für die Evaluierung wurde dann anhand der segmentierten Bilder die Anzahl korrekter Segmente ermittelt. Für die falsch segmentierten Regionen wird unterschieden, ob diese unter- oder übersegmentiert sind. Es ist zu erkennen, dass der Prozentsatz R richtig segmentierter Flächen bei allen Szenen bis auf einer (*kitchen far*) bei über 80% liegt. Die falschen Segmente zeigen einen Hang zur Untersegmentierung.

Szene	Anzahl erkannter Segmente	davon korrekt	davon übersegmentiert	davon untersegmentiert	R (%)
kitchen close	23	21	1	1	91,3
kitchen far	14	10	1	3	71,4
table close	41	39	1	1	95,1
table far	28	23	4	1	82,1
office close	24	22	0	2	91,7
office far	35	30	1	4	85,7
cupboard close	38	36	0	2	94,7
cupboard far	51	44	1	6	86,3
Mittelwert (σ)	31,8	28,4	1,1	2,5	87,3 ($\pm 0,079$)

Tabelle 6.7: Genauigkeit der Segmentierung bei GRG

In Tabelle 6.8 sind die Konfusionsmatrizen für die Segmentklassifizierung bei GRG zu sehen. Es sind die erkannten Klassen (Index p) über den wahren Klassen (Index e) aufgetragen. Als Basis für die Klassifizierung dienen die von GRG bestimmten Segmente. Die zu erwartende Klasse wurde für jedes Segment manuell festgelegt. GRG benutzt nicht alle von PC unterscheidbaren Klassen und kann daher nur Ebenen (P), Zylinder (C) und komplexere Strukturen (U) klassifizieren. Die Gründe dafür sind zum einen, dass Ecken und Kanten für die zu erzeugende GK nicht relevant sind. Zum anderen besitzen komplexe Strukturen ähnliche Deskriptoren wie Kugeln, so dass keine robuste Unterscheidung möglich ist. Aus den Matrizen ist abzulesen, dass das Verfahren insgesamt gute Klassifizierungsergebnisse liefert (die Mehrzahl der Werte liegen auf den Hauptdiagonalen) und bei Nahbereichsszenen besser abschneidet als bei weiter entfernten. Dies liegt hauptsächlich am höheren Sensorrauschen bei größerer Entfernung.

Um die Ergebnisse noch besser zu verdeutlichen, werden noch die *micro* und *macro average* Werte berechnet (siehe Anhang A). Diese sind in Tabelle 6.9 zu sehen und bestätigen, dass GRG

<table border="1"> <thead> <tr><th>P_p</th><th>C_p</th><th>U_p</th></tr> </thead> <tbody> <tr><td>P_e</td><td>7</td><td></td></tr> <tr><td>C_e</td><td></td><td>2</td></tr> <tr><td>U_e</td><td>1</td><td>3</td></tr> </tbody> </table>	P_p	C_p	U_p	P_e	7		C_e		2	U_e	1	3	<table border="1"> <thead> <tr><th>P_p</th><th>C_p</th><th>U_p</th></tr> </thead> <tbody> <tr><td>P_e</td><td>9</td><td></td></tr> <tr><td>C_e</td><td></td><td>0</td></tr> <tr><td>U_e</td><td>1</td><td>2</td></tr> </tbody> </table>	P_p	C_p	U_p	P_e	9		C_e		0	U_e	1	2	<table border="1"> <thead> <tr><th>P_p</th><th>C_p</th><th>U_p</th></tr> </thead> <tbody> <tr><td>P_e</td><td>23</td><td></td></tr> <tr><td>C_e</td><td></td><td>5</td></tr> <tr><td>U_e</td><td></td><td>10</td></tr> </tbody> </table>	P_p	C_p	U_p	P_e	23		C_e		5	U_e		10	<table border="1"> <thead> <tr><th>P_p</th><th>C_p</th><th>U_p</th></tr> </thead> <tbody> <tr><td>P_e</td><td>18</td><td></td></tr> <tr><td>C_e</td><td></td><td>2</td></tr> <tr><td>U_e</td><td>4</td><td>7</td></tr> </tbody> </table>	P_p	C_p	U_p	P_e	18		C_e		2	U_e	4	7				
P_p	C_p	U_p																																																					
P_e	7																																																						
C_e		2																																																					
U_e	1	3																																																					
P_p	C_p	U_p																																																					
P_e	9																																																						
C_e		0																																																					
U_e	1	2																																																					
P_p	C_p	U_p																																																					
P_e	23																																																						
C_e		5																																																					
U_e		10																																																					
P_p	C_p	U_p																																																					
P_e	18																																																						
C_e		2																																																					
U_e	4	7																																																					
(a) kitchen close	(b) kitchen far	(c) table close	(d) table far																																																				
<table border="1"> <thead> <tr><th>P_p</th><th>C_p</th><th>U_p</th></tr> </thead> <tbody> <tr><td>P_e</td><td>18</td><td></td></tr> <tr><td>C_e</td><td>1</td><td>1</td></tr> <tr><td>U_e</td><td></td><td>4</td></tr> </tbody> </table>	P_p	C_p	U_p	P_e	18		C_e	1	1	U_e		4	<table border="1"> <thead> <tr><th>P_p</th><th>C_p</th><th>U_p</th></tr> </thead> <tbody> <tr><td>P_e</td><td>26</td><td></td></tr> <tr><td>C_e</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>U_e</td><td></td><td>3</td><td>7</td></tr> </tbody> </table>	P_p	C_p	U_p	P_e	26		C_e	1	1	1	U_e		3	7	<table border="1"> <thead> <tr><th>P_p</th><th>C_p</th><th>U_p</th></tr> </thead> <tbody> <tr><td>P_e</td><td>29</td><td></td></tr> <tr><td>C_e</td><td></td><td>4</td></tr> <tr><td>U_e</td><td>2</td><td>5</td></tr> </tbody> </table>	P_p	C_p	U_p	P_e	29		C_e		4	U_e	2	5	<table border="1"> <thead> <tr><th>P_p</th><th>C_p</th><th>U_p</th></tr> </thead> <tbody> <tr><td>P_e</td><td>30</td><td>5</td></tr> <tr><td>C_e</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>U_e</td><td></td><td></td><td>8</td></tr> </tbody> </table>	P_p	C_p	U_p	P_e	30	5	C_e	1	1	1	U_e			8
P_p	C_p	U_p																																																					
P_e	18																																																						
C_e	1	1																																																					
U_e		4																																																					
P_p	C_p	U_p																																																					
P_e	26																																																						
C_e	1	1	1																																																				
U_e		3	7																																																				
P_p	C_p	U_p																																																					
P_e	29																																																						
C_e		4																																																					
U_e	2	5																																																					
P_p	C_p	U_p																																																					
P_e	30	5																																																					
C_e	1	1	1																																																				
U_e			8																																																				
(e) office close	(f) office far	(g) cupboard close	(h) cupboard far																																																				

Tabelle 6.8: Konfusionsmatrizen für die Klassifizierung bei GRG

vor allem im Nahbereich sehr gute Erkennungswerte hat. Aber auch für die fernen Szenen sind die Werte noch akzeptabel. Insgesamt ergibt sich eine Erkennungsrate von über 80%.

Szene	R_{mic}	R_{mac}	P_{mac}	F_{mac}
kitchen close	0,923	0,958	0,917	0,937
kitchen far	0,917	0,633	0,556	0,888
table close	1	1	1	1
table far	0,871	0,939	0,879	0,908
office close	0,958	0,982	0,833	0,902
office far	0,872	0,696	0,678	0,687
cupboard close	0,95	0,889	0,905	0,897
cupboard far	0,848	0,846	0,730	0,784
Mittelwert	0,917	0,868	0,812	0,875

Tabelle 6.9: Genauigkeit der Klassifizierung nach Szenen

Interessant ist auch die Genauigkeit von GRG bei kombinierter Betrachtung von Segmentierung und Klassifizierung. Es wird bewertet, in wie vielen Segmenten mit korrekten Ausmaßen und korrekter Oberflächenklasse die Verarbeitung resultiert. Da bei dieser Auswertung nur die TP und FN bestimmt werden können (Falschsegmentierungen können keiner Klasse zugeordnet werden), kann auch nur der *recall* R_p und R_{mic} angegeben werden. Die Ergebnisse in Tabelle 6.10 sind erwartungsgemäß schlechter als die in Tabelle 6.9, da nun auch die Segmentierungsfehler eine Rolle spielen. Dennoch liefert GRG in vielen Szenen sehr gute Resultate. Generell ist wieder die Abhängigkeit von der Entfernung zu erkennen. Die Genauigkeitswerte für Ebenen R_p sind deutlich höher, was auf eine Stärke von GRG bei Ebenen schließen lässt. Bildlich dargestellte Klassifizierungsergebnisse für einige Szenen finden sich in Abbildung 6.29. Diese bestätigen die Zahlenwerte.

Szene	n	Ebene		Zylinder		komplex		R_P	R_{mic}
		e	p	e	p	e	p		
kitchen close	22	12	11	2	0	8	3	0,916	0,636
kitchen far	9	8	7	1	0	0	0	0,875	0,778
office close	21	17	15	4	1	0	0	0,882	0,762
office far	34	30	21	3	1	1	0	0,700	0,647
cupboard close	41	34	29	7	6	0	0	0,823	0,854
cupboard far	52	48	29	4	1	0	0	0,604	0,577
table close	35	26	23	5	3	4	2	0,885	0,800
table far	22	16	8	4	0	2	1	0,500	0,409
Mittelwert								0,773	0,683

n:Anzahl Segmente e: erwartet p: gefunden

Tabelle 6.10: Kombinierte Genauigkeit von GRG

Der Vergleich von ERS und GRG gegenüber MPS beschränkt sich auf die Evaluierung von Ebenen, da sowohl MPS als auch ERS nur diese extrahieren können. Die Ergebnisse sind in Tabelle 6.11 dargestellt. Bei dem Anteil gefundener Ebenen liegt GRG mit durchschnittlich 96,1% weit vor MPS mit 51,3% und ERS mit 29,4%. Der Mittelwert der Winkelabweichung $\Delta\alpha$ der Normalenvektoren ist bei ERS und bei MPS besser als bei GRG. Dies liegt daran, dass beide Verfahren nur größere Ebenen finden, welche robuster sind als kleinere. Beim Median von $\Delta\alpha$ zeigt sich, dass der Großteil der Ebenen bei GRG sehr kleine Fehler aufweist. Dies lässt auf einzelne Ausreißer bei sehr kleinen Ebenen schließen. Um die Konturen der Ebenenpolygone zu bewerten, wird der Flächenfehler ΔA betrachtet. Hier zeigt GRG mit nur 0,006 m² gegenüber MPS mit 0,054 m² und ERS mit 0,112 m² eine sehr kleine Abweichung. Dies zeigt, dass die Konturen durch GRG am besten beschrieben werden. Die Evaluierung zeigt, dass es auch durch den zusätzlichen Vorverarbeitungsschritt des *Clustering* nicht gelingt, die Modelltreue von ERS signifikant zu erhöhen. GRG hingegen weist durch den die Über- und Untersegmentierung vermindernenden Optimierungsschritt sowohl bei der Vollständigkeit der Beschreibung als auch bei der Genauigkeit deutlich bessere Werte auf als MPS.

Schließlich soll noch anhand der Szenen in Abbildung 6.29 eine visuelle Analyse durchgeführt werden, um zu überprüfen, ob eine subjektive Bewertung der Bilder mit den objektive Zahlenwerten übereinstimmt. Die Szene *kitchen far* besteht fast nur aus größeren Ebenen. Dadurch erzielen alle Verfahren gute Ergebnisse. *Table close* ist eine Szene, bei der ERS relativ gut abschneidet. Es ist zu erkennen, dass große Ebenen zuverlässig segmentiert werden. Die Objekte auf dem Tisch werden alle als jeweils ein Segment erkannt. GRG segmentiert hier deutlich detailreicher, erkennt beispielsweise auch die einzelnen Ebenen der Objekte auf dem Tisch. Zu erkennen ist auch die mangelnde Modelltreue von ERS. So wird der Boden der Schüssel der Tischoberfläche zugerechnet, da die Punkte etwa in der gleichen Ebene liegen. Bei den Szenen *office far* und *cupboard close* verhindert das Vorhandensein von kleinen Strukturen eine gute Segmentierung durch ERS. Bei der Regalszene können nur vier Ebenen gefunden werden, von denen zwei mitten durch mehrere Segmente gelegt werden. Bei GRG sehen die segmentierten Szenen der Grundwahrheit sehr ähnlich, was die Zahlenwerte bestätigt. MPS zeigt gegenüber ERS deutlich bessere Ergebnisse, im Vergleich mit GRG fällt eine Übersegmentierung bei allen Szenen auf.

Szene	Gef. Ebenen (%)			durchschnittl. $\Delta\alpha$ (rad)			Median $\Delta\alpha$ (rad)			$\Delta A(m^2)$		
	MPS	ERS	GRG	MPS	ERS	GRG	MPS	ERS	GRG	MPS	ERS	GRG
kitchen close	45,5	27,2	100	0,014	0,038	0,017	0,041	0,047	0	0,077	0,138	0,003
kitchen far	100	100	100	0,034	0,050	0	0,045	0,042	0	0,290	0,234	0,019
table close	39,1	21,7	100	0,016	0,013	0,025	0,035	0,011	0	0,077	0,105	0,006
table far	28,6	21,4	92,9	0,090	0,026	0,259	0,139	0,03	0,01	0,139	0,123	0,015
office close	50,0	18,8	100	0,016	0,059	0,227	0,029	0,017	0	0,027	0,083	0,001
office far	53,3	26,7	96,7	0,046	0,139	0,058	0,091	0,063	0	0,058	0,103	0
cupboard close	50,0	10,5	92,1	0,015	0,130	0,067	0,027	0,047	0	0,020	0,073	0,001
cupboard far	43,5	8,7	87,0	0,026	0,140	0,046	0,054	0,151	0	0,015	0,037	0,005
Mittelwert	51,3	29,4	96,1	0,032	0,074	0,087	0,052	0,051	0	0,054	0,112	0,006

am schlechtesten
 mittel
 am besten

Tabelle 6.11: Vergleich von ERS, MPS und GRG

Die Evaluierung der Rechenzeit erfolgt durch mehrmaliges Segmentieren verschiedener Szenen und Berechnung des Mittelwerts der Ausführungszeiten. Die Zeiten gelten für die vollständige Segmentierung und Klassifikation inklusive Normalenberechnung. Für das ERS-Verfahren wurde die Punktwolke über einen Voxelfilter mit einer Kantenlänge von 0,03 m heruntergerechnet. In Tabelle 6.12 ist zu sehen, dass die Zeiten für ERS und GRG ähnlich sind. MPS ist das schnellste Verfahren. Der Grund für die längere Rechenzeit von GRG liegt zum einen am Optimierungsschritt und zum anderen an der gegenüber den anderen Algorithmen zusätzlich vorhandenen Funktionalität der Segmentierung gekrümmter Flächen. Für einen Vergleich der Rechenzeit bei gleicher Funktionalität wird die Rechenzeit von GRG betrachtet, wenn nur Ebenen segmentiert werden. Die Messergebnisse zeigen, dass in diesem Fall die MPS nur noch um den Faktor 1,71 schneller ist.

	ERS	MPS	GRG	GRG (nur Ebenen)
Rechenzeit (s)	0,211	0,028	0,191	0,048

Tabelle 6.12: Rechenzeit der Verfahren zur Segmentierung

Insgesamt ist zu sagen, dass GRG bei der Genauigkeit den beiden anderen Methoden überlegen ist. Darüber hinaus kann GRG mehrere Oberflächenklassen unterscheiden und erreicht auch bei den Ebenen eine höhere Segmentierungsrate als die Vergleichsverfahren. Somit kann die Forderung A10 erfüllt werden, da GRG sowohl genauer rechnet als auch die Szene mit einem höheren Vollständigkeitsgrad wiedergeben kann als existierende Verfahren. Die Rechenzeit ist bei GRG zwar höher als bei MPS, für die Anforderung (A6) dieser Arbeit aber durchaus ausreichend.

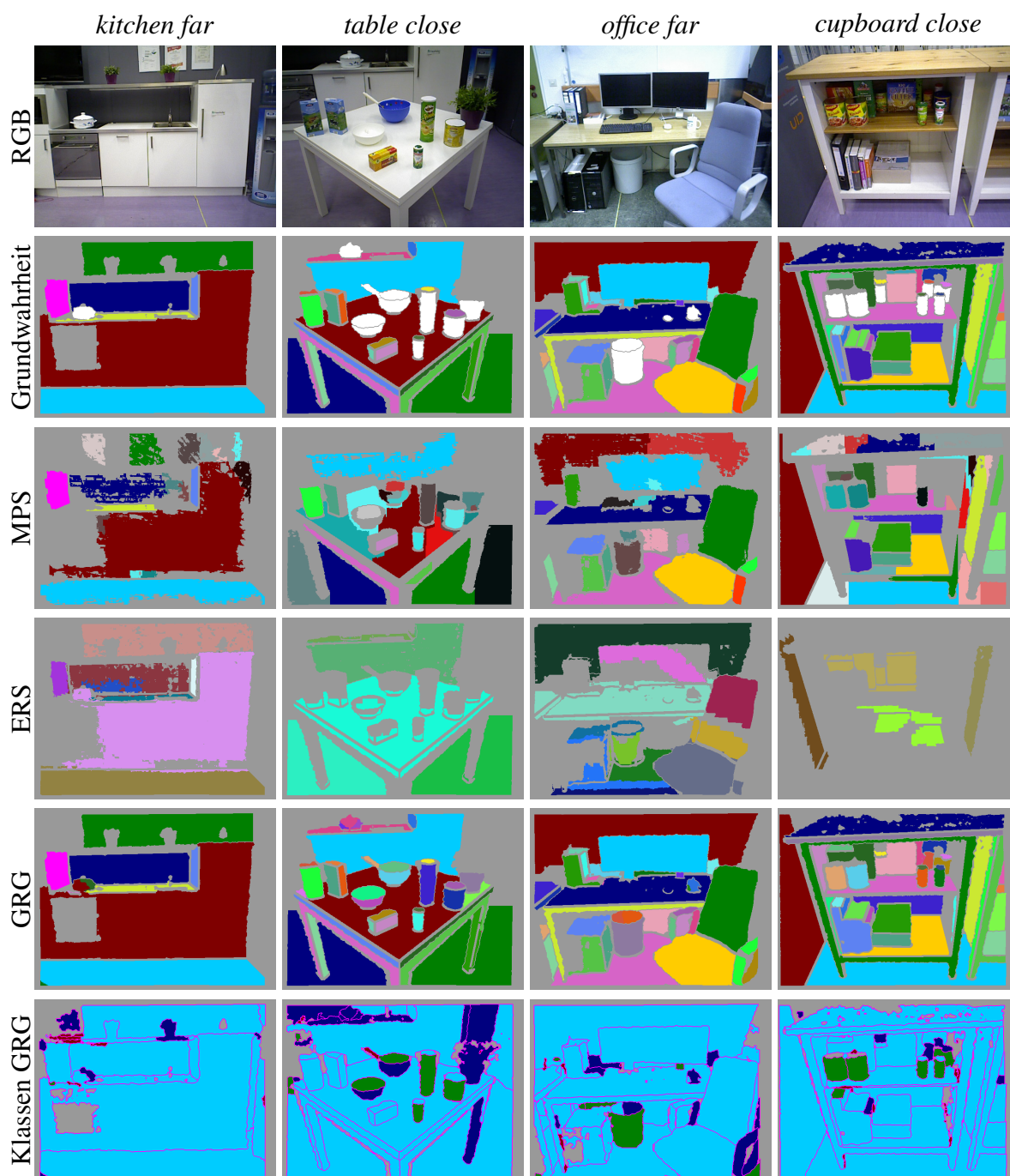


Abbildung 6.29: Visueller Vergleich von MPS, ERS und GRG. Klassifizierung von GRG.

6.7 Aufbau einer geometrische Karte

Werden geometrische Merkmale mit den in Abschnitt 6.6 beschriebenen Verfahren extrahiert, müssen diese in einer Karte zusammengeführt werden, um eine ganzheitliche Repräsentation der Umgebung zu erhalten. Dabei liegt die Herausforderung darin, dass dieselben Regionen der Umgebung während der Roboterbewegung mehrmals und aus unterschiedlichen Blickwinkeln observiert werden. Dadurch werden viele Geometrien mehrfach extrahiert. Dies führt zu teilweise redundanten, teilweise überlappenden Formen, die in der geometrischen Karte so vereinigt werden müssen, dass eine möglichst minimale und dabei doch vollständige Beschreibung der Umgebung entsteht.

Die Untersuchungen im Bereich der Segmentierung haben gezeigt, dass vor allem Ebenenpolygone und Zylinder robust extrahiert werden können, deswegen soll der Aufbau der geometrischen Karte für diese beiden Basisgeometrien gezeigt werden. Für komplexere Geometrien bedarf es anderer Ansätze, die außerhalb des Fokus dieser Arbeit liegen.

6.7.1 Verarbeitung von Ebenenpolygonen

Die Ebenenpolygone werden von der Segmentierung in Form der Ebenenparameter (D.1) und der konkaven Hülle ausgegeben. Diese Information wird in der Karte genutzt, um zusammengehörige Polygone zu fusionieren (mergen). Zunächst wird untersucht, ob es aufgrund der Ebenenparameter geeignete Merge-Kandidaten gibt, d. h. ob die Ebenen ähnlich sind. Das Ähnlichkeitskriterium wird durch den Winkel α zwischen den beiden normierten Ebenennormalen \mathbf{n}_1 , \mathbf{n}_2 der Polygone P_1 und P_2 und dem Winkel β der Verbindungslinie zwischen den Flächenschwerpunkten \mathbf{c}_1 , \mathbf{c}_2 und dem Normalenvektor \mathbf{n}_1 bestimmt (Abbildung 6.30). Das bedeutet, dass P_1 und P_2 genau dann Merge-Kandidaten sind, wenn die Bedingungen

$$\arccos(\mathbf{n}_1 \cdot \mathbf{n}_2) < \alpha_{th} \quad (6.60)$$

$$\arccos\left(\frac{\mathbf{n}_1 \cdot (\mathbf{c}_2 - \mathbf{c}_1)}{\|\mathbf{c}_2 - \mathbf{c}_1\|}\right) < \beta_{th} \quad (6.61)$$

erfüllt sind. Im Folgenden wird angenommen, dass sich P_1 bereits in der Karte befindet und P_2 ein neues Polygon ist.

Sind zwei Merge-Kandidaten identifiziert worden, werden sie im nächsten Schritt fusioniert. Zunächst wird eine virtuelle Ebene

$$P_3 : \frac{(wa_1 + a_2)x + (wb_1 + b_2)y + (wc_1 + c_2)z + (wd_1 + d_2)}{\sqrt{(wa_1 + a_2)^2 + (wb_1 + b_2)^2 + (wc_1 + c_2)^2}} = 0 \quad (6.62)$$

definiert, da P_1 und P_2 leicht unterschiedliche Parameter besitzen können und somit eine mittlere vereinte Ebene erzeugt wird. Das Gewicht w hat bei neuen Einträgen den Wert 1, wird nach jeder Fusion erhöht und bewirkt dadurch einen stärkeren Einfluss von Polygonen, die schon länger in der Karte sind und mehrmals aktualisiert wurden. Dies erhöht die Robustheit gegenüber Ausreißern. Weiterhin wird w mit der Fläche des Polygons multipliziert, was sich ebenfalls positiv auf die Robustheit auswirkt. In P_3 wird ein Koordinatensystem mit der z -Achse gemäß (D.3)

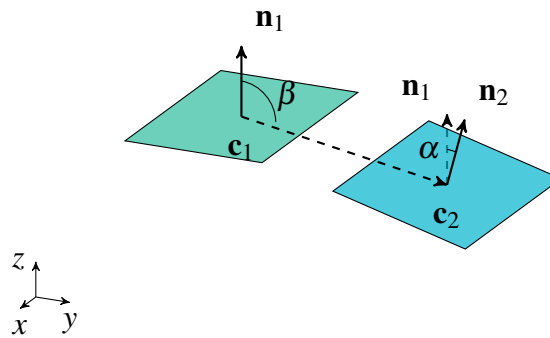


Abbildung 6.30: Bestimmung von Merge-Kandidaten

definiert. Die x - und y -Achsen können frei gewählt werden und müssen nur in der Ebene liegen. Der Ursprung des Koordinatensystems wird über den Flächenschwerpunkt \mathbf{c}_3 des Polygons festgelegt.

Im nächsten Schritt werden P_1 und P_2 in das Koordinatensystem transformiert und auf die x - y -Ebene projiziert. Dies führt zu Projektionsfehlern bei der Hülle, was aber bei kleinem Winkel α zwischen den Ebenen vernachlässigt werden kann. Durch die Projektion reduziert sich das Merge-Problem von drei auf zwei Dimensionen und es können 2-D-Polygonschnittverfahren angewendet werden. In dieser Arbeit wird das Verfahren von Vatti (vgl. Abschnitt 4.3.8) benutzt, da es effizient rechnet, mit konvexen und konkaven Polygonen umgehen kann und auch mit Löchern zurechtkommt. Zunächst wird geprüft, ob sich die Polygone überlappen. Ist dies der Fall, so wird ein vereintes Polygon erzeugt und das Gewicht w erhöht. Abschließend wird das neu entstandene Polygon zurück in das ursprüngliche Koordinatensystem transformiert. Zusätzlich wird noch der Abstandsparameter d gemäß

$$d = \mathbf{n}_3 \cdot \mathbf{c}_3 \quad (6.63)$$

aus Normalenvektor \mathbf{n}_3 und Flächenschwerpunkt \mathbf{c}_3 neu berechnet. Abbildung 6.31 zeigt eine Merging-Sequenz über zehn Schritte. Es ist zu erkennen, dass die konkaven Hüllpolygone während der Kamerabewegung durch Fusion mit neuen Segmenten wachsen.

Liegen Segmente mit einem dominanten Farbwert vor, können die Farbwerte über eine mit w gewichtete Mittelwertbildung im RGB-Farbraum neu berechnet werden. Im Gegensatz zur Segmentierung muss nicht im HSV-Raum gerechnet werden, da die Eingangspolygone ähnliche Farben besitzen.

6.7.2 Verarbeitung von Zylindern

Da es sich bei Zylindern um „echte“ 3-D-Strukturen handelt, erscheint es zunächst deutlich komplexer, diese in einer geometrischen Karte zu verarbeiten und zu mergen. Bei genauerer Betrachtung fällt jedoch auf, dass sich eine Zylinderhülle „abrollen“ lässt und somit wieder ein 2-D-Polygon erzeugt werden kann. Dieses kann dann ähnlich behandelt werden wie im vorherigen Abschnitt beschrieben.

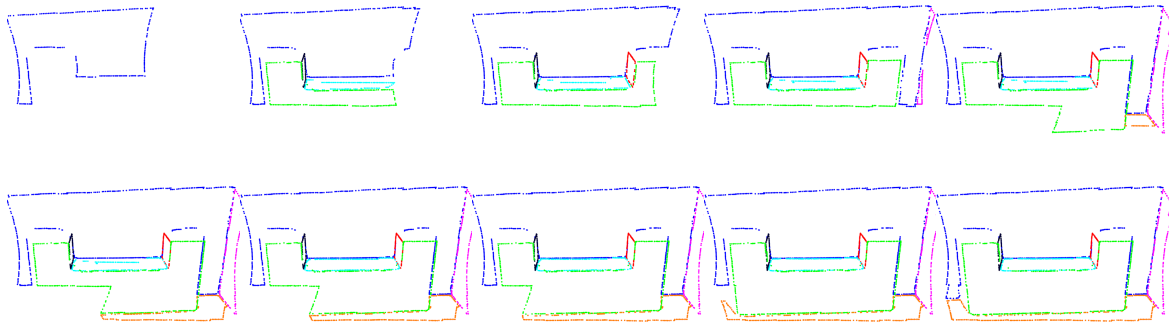


Abbildung 6.31: Merge-Sequenz für Ebenen

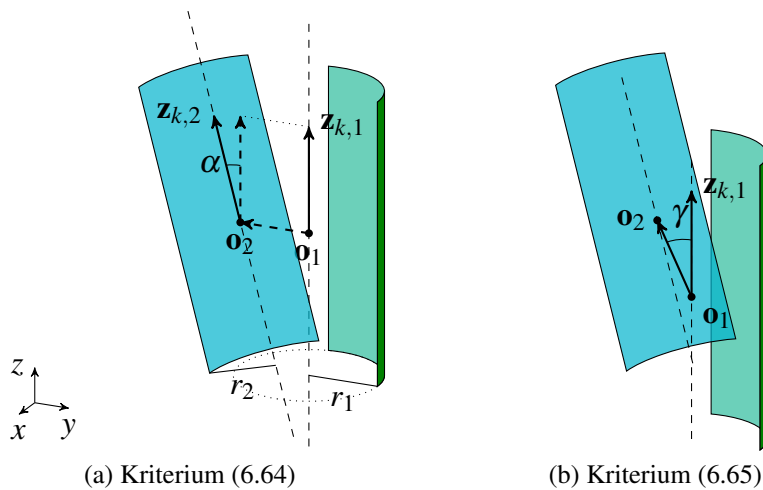


Abbildung 6.32: Bestimmung von Merge-Kandidaten

Im ersten Schritt werden wiederum Merge-Kandidaten ermittelt. Ähnlich wie bei den Ebenenpolygonen werden die Geometrieparameter (Anhang D) verglichen. Bei Zylindern ist es dabei ausreichend, die Symmetrieachsen $\mathbf{z}_{k,1}$ und $\mathbf{z}_{k,2}$ sowie die Radien r_1 und r_2 gegenüberzustellen, falls die Koordinatenursprünge \mathbf{o}_1 und \mathbf{o}_2 nahe beieinander liegen. Dies führt zu den Kriterien

$$\left. \begin{aligned} \arccos(\mathbf{z}_{k,1} \cdot \mathbf{z}_{k,2}) &< \alpha_{th} \\ \|r_1 - r_2\| &< r_{th} \end{aligned} \right\} \text{ falls } \|\mathbf{o}_2 - \mathbf{o}_1\| < o_{th}. \quad (6.64)$$

Liegen die Koordinatenursprünge weiter auseinander, kann es sich bei den Zylindersegmenten trotzdem um Merge-Kandidaten handeln, falls das zusätzliche Kriterium

$$\arccos\left(\frac{(\mathbf{o}_2 - \mathbf{o}_1) \cdot \mathbf{z}_{k,1}}{\|\mathbf{o}_2 - \mathbf{o}_1\| \|\mathbf{z}_{k,1}\|}\right) < \gamma_{th} \quad (6.65)$$

erfüllt ist. Dies deckt den Fall ab, dass die Schwerpunkte entlang der Symmetrieachse gegeneinander verschoben sind. In Abbildung 6.32 sind die geometrischen Beziehungen dargestellt.

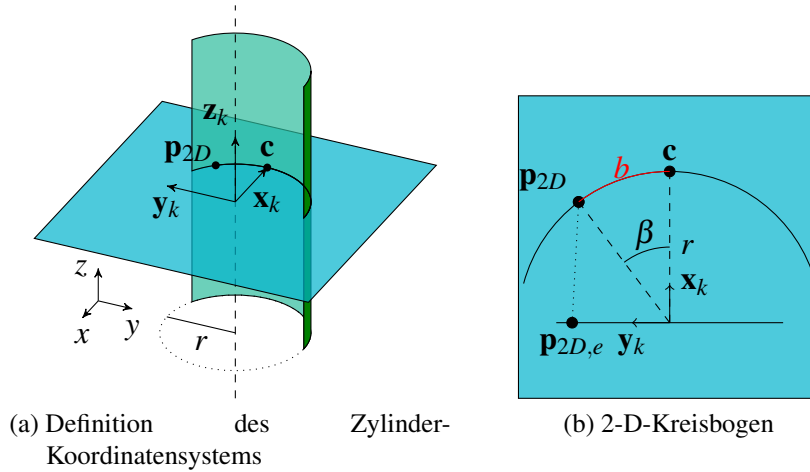


Abbildung 6.33: Projektion des Zylinders

Um zwei Zylinder Z_1 und Z_2 zu fusionieren, wird zunächst Z_2 in das Koordinatensystem K_z (vgl. Abschnitt 6.6.2) transformiert. Nachfolgend werden beide Zylinderhüllen in eine Ebene projiziert (Abbildung 6.33). Um dies zu erreichen, werden nur die x - und y -Werte der Hüllpunkte \mathbf{p} betrachtet, welche einen 2-D-Kreisbogen beschreiben. Für jeden 2-D-Hüllpunkt \mathbf{p}_{2D} wird der Winkel

$$\beta = \arccos \left(\frac{\mathbf{x}_k \cdot \mathbf{p}_{2D}}{\|\mathbf{x}_k\| \cdot \|\mathbf{p}_{2D}\|} \right) \quad (6.66)$$

zur x -Achse \mathbf{x}_k des Zylinderkoordinatensystems berechnet. Aus diesem kann dann über

$$b = r\beta \quad (6.67)$$

die Bogenlänge b bestimmt werden. Diese stellt die y -Koordinate der projizierten Hülle dar. Die neuen Koordinaten ergeben sich folglich zu

$$\mathbf{p}_{e,2D} = (b, p_z). \quad (6.68)$$

Das Verfahren resultiert in einem Merge-Problem für zwei 2-D-Polygone, welches genauso behandelt werden kann, wie im vorherigen Abschnitt beschrieben. Nach erfolgreicher Fusion wird der neue Zylinder zurück in das 3-D-Koordinatensystem transformiert. Abbildung 6.34 zeigt eine Merge-Sequenz für einen Zylinder. Die Kamera wird zunächst nach unten geneigt, wodurch die volle Zylinderhöhe erfasst wird. Anschließend erfolgt eine Bewegung um den Zylinder, wodurch dieser in der Karte anwächst.

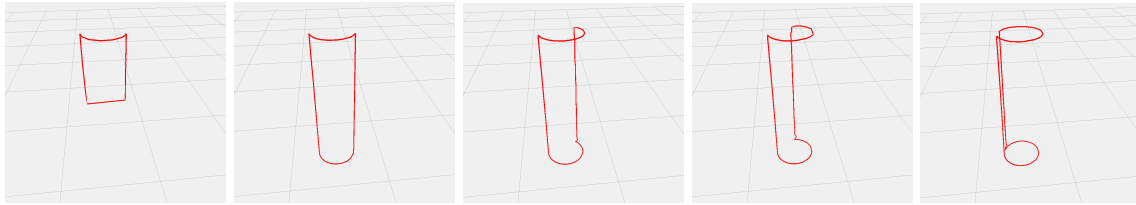
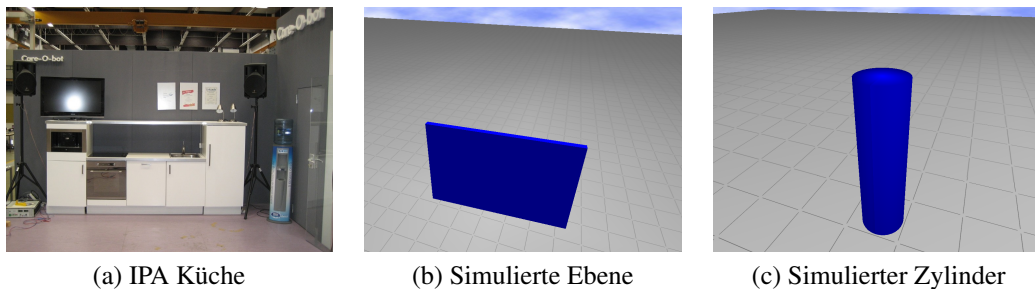


Abbildung 6.34: Merge-Sequenz für einen Zylinder



(a) IPA Küche

(b) Simulierte Ebene

(c) Simulierter Zylinder

Abbildung 6.35: Szenen für die Evaluierung

6.7.3 Evaluierung

Auch für die Erstellung der Geometrie Karte wird sowohl die Genauigkeit als auch der Rechenaufwand evaluiert. Die Genauigkeitsuntersuchung wird für Ebenen und Zylinder getrennt behandelt, da unterschiedliche Parameter bewertet werden. Die Evaluierung der Ebenen wird sowohl auf simulierten als auch auf realen Szenen durchgeführt. Für die synthetische Szene wird eine einzelne Ebene modelliert (Abbildung 6.35b), wohingegen die echte Szene in der Modellküche am Fraunhofer IPA aufgenommen wurde (Abbildung 6.35a). Die Untersuchung der Zylinder erfolgt in der Simulation anhand eines modellierten Zylinders. Die zugehörige Szene ist in Abbildung 6.35c zu sehen. Bei den simulierten Szenen wird das Verhalten der GK ohne und mit hinzugefügtem Gaußschen Rauschen mit $\mu = 0$ m und $\sigma = 0,015$ m untersucht.

Für die Genauigkeitsuntersuchung werden die eingehenden Einzelsegmente mit den resultierenden Einträgen in der Karte verglichen. Bei den Ebenen werden dafür die Winkelabweichung der Normalenvektoren sowie die Abweichung des Ebenenparameters d betrachtet. Als Parameter für die Fusion werden $\alpha_{th} = 0,97$ rad und $\beta_{th} = 0,05$ rad gewählt. Für die Zylinder werden die Richtung der Symmetrieachse, der Radius und der Ursprung verglichen. Für die Parameter gilt $\alpha_{th} = 0,97$ rad, $r_{th} = 0,1$ m, $o_{th} = r + 0,1$ m und $\gamma_{th} = 0,97$ rad.

Die Verläufe der Messwerte für die simulierte Ebene sind in Abbildung 6.36 und Abbildung 6.37 dargestellt. Da die eingehenden Segmente durch die fast perfekten Simulationsdaten schon sehr genau sind, ist kaum ein Unterschied zur resultierenden Karte zu erkennen. Die maximale Abweichung liegt schon bei den Einzelsegmenten bei weniger als 0,001 rad für den Normalenfehler bzw. bei weniger als 1 mm für d . Somit dient diese Untersuchung nur als Beweis, dass das

Fusionsverfahren korrekt rechnet. Bei der rauschbehafteten Ebene hingegen ist eine deutliche Reduzierung beider Fehler in der Karte gegenüber den Einzelsegmenten zu beobachten.

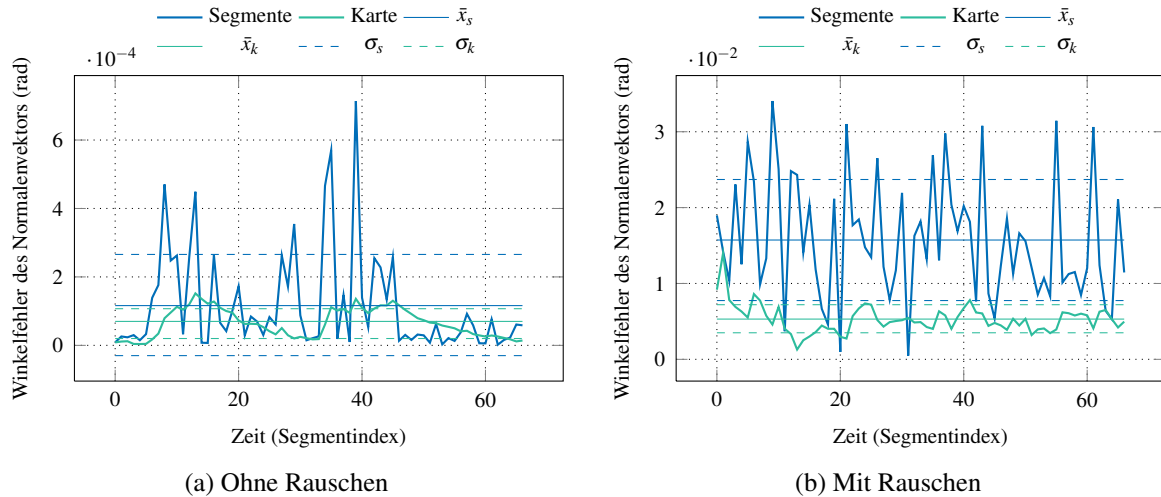


Abbildung 6.36: Winkelfehler des Normalenvektors bei der Ebene (jeweils mit Mittelwert \bar{x} und Standardabweichung σ)

Nachfolgend wird die Genauigkeit anhand der realen Küchenszene untersucht. In Abbildung 6.38 sind die Ergebnisse für die Rückwand hinter der Küche dargestellt, da hier am meisten Polygon-schnittoperationen auftreten. Im Zeitraum zwischen Index 800 und 1 300 werden keine Segmente mit der Rückwand fusioniert, wodurch sich der Karteneintrag nicht verändert. Durch die verauschten Sensordaten sind sowohl die Fehler der Einzelsegmente als auch die der Karte größer als in der Simulation. Es ist jedoch zu erkennen, dass der mittlere Fehler und auch die Standardabweichung in der Karte deutlich geringer sind als bei den Einzelsegmenten. Dies kommt daher, dass durch die Mittelung der Ebenen bei der Fusion und die Gewichtungsfaktoren ein Tiefpassverhalten auftritt. Insgesamt gesehen ist die Genauigkeit der Karte ausreichend, da der mittlere Winkelfehler bei 0,031 rad und der mittlere Abstandsfehler bei 0,004 m liegt.

Interessant ist auch das Fusionsverhalten in der Karte über die Zeit. Die in Abbildung 6.39 gezeigten Verläufe verdeutlichen, wie sich ein Karteneintrag über die Zeit verändert und wie von Zeit zu Zeit mehrere Einträge verbunden werden. Aufgetragen ist die Winkelabweichung der Normalen und die Winkel zwischen Normale und Verbindungslinie der Schwerpunkte nach (6.60). Bei Index 579 werden zwei Karteneinträge fusioniert. Der Punkt zeigt den Parameterwert des neuen Segments, welches ebenfalls verbunden wird. Es ist zu sehen, dass die Parameter des neuen Segments dazu führen, dass die Kriterien für die Fusion erfüllt werden. Gleiches gilt für Index 1 335. Dort werden jedoch gleich vier Karteneinträge fusioniert.

Für Zylinder wird ebenfalls der Verlauf der Modellparameter (Ursprung, Radius, Symmetrieachse) evaluiert. Die Testszene ist in Abbildung 6.35c zu sehen. Der Roboter bewegt sich einmal um den Zylinder, wobei die Kamera auf und ab bewegt wird. Es werden ebenfalls die Fälle ohne Rauschen und mit hinzugefügtem Rauschen untersucht. In den Abbildungen 6.40 bis 6.42 sind die Ergebnisse aufgetragen. Es ist zu sehen, dass die Fehler beim Fall mit Rauschen größer

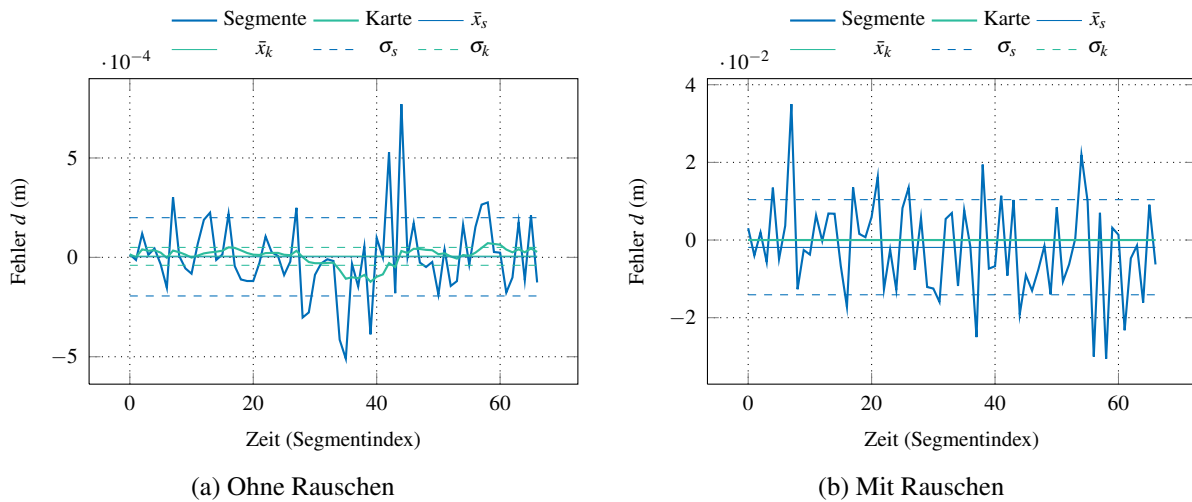


Abbildung 6.37: Fehler des Parameters d bei der Ebene (jeweils mit Mittelwert \bar{x} und Standardabweichung σ)

sind als ohne Rauschen, insgesamt jedoch klein bleiben (der maximale Fehler für Ursprung und Radius ist sehr viel kleiner als 1 cm). Beim Winkelfehler der Symmetrieachse ist eine deutliche Annäherung an den tatsächlichen Wert in der Karte erkennbar. Dies ist durch die kreisförmige Bewegung der Kamera um den Zylinder zu erklären. Der maximale Winkelfehler ist kleiner als 0,005 rad. Auch beim Zylinder ist wiederum ein Tiefpassverhalten erkennbar, d. h. Ausreißer bei den Segmenten wirken sich nur abgeschwächt auf die Karte aus.

Der Rechenaufwand wird durch Hinzufügen eines einzelnen Segments bei unterschiedlichen Kartengrößen gemessen und wird in Abbildung 6.43 gezeigt. Es sind die Zeiten für den kompletten Merge-Vorgang sowie die Teilschritte Kandidatenprüfung und Polygonschnitt über der Anzahl der Karteneinträge aufgetragen. Die Zeiten für die Kandidatenprüfung und den kompletten Vorgang steigen linear mit der Kartengröße an ($O(n)$). Dies liegt daran, dass immer alle Kartenelemente geprüft werden müssen, wenn ein neues Segment in die Karte kommt. Die Ausführungszeit für den Polygonschnitt bleibt hingegen konstant. Insgesamt ist die Ausführungszeit mit durchschnittlich 2 ms sehr gering.

Die Evaluierung zeigt, dass das vorgestellte Verfahren zum Aufbau einer Geometrikarte ausreichend genau ist, um konsistente Karten zu erzeugen. Somit trägt dieser Teilschritt dazu bei, Anforderung A10 zu erfüllen. Auch die Robustheit gegenüber Rauschen (A12, A13) wird durch das Verfahren erhöht. Die benötigte Rechenzeit ist dabei gering, so dass das Verfahren echtzeitfähig (vgl. Anforderung A6) ist.

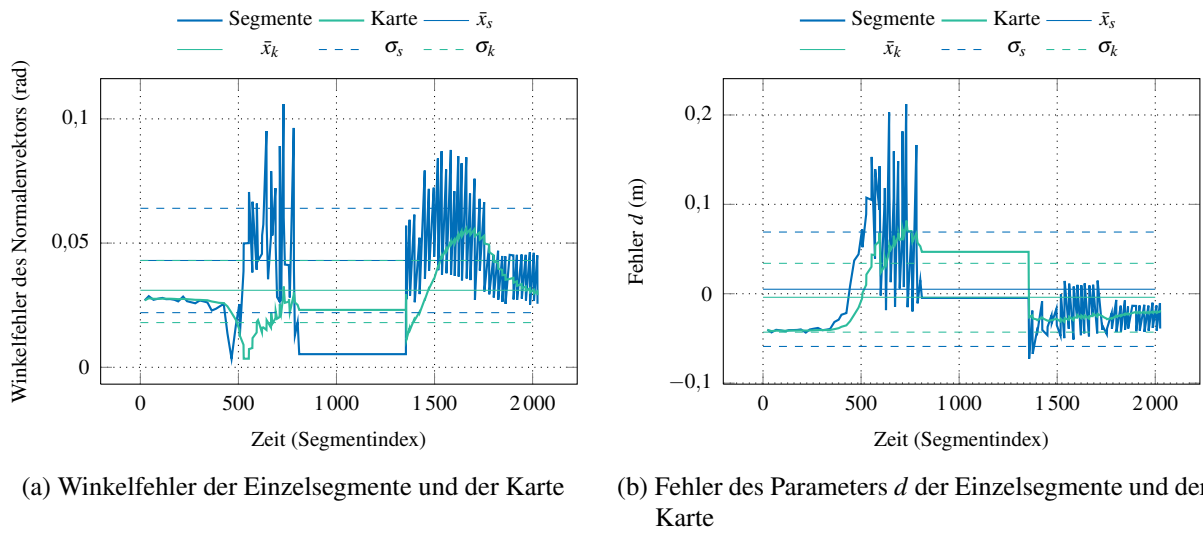


Abbildung 6.38: Messung der Genauigkeit für die Rückwand in der Küchenszene (jeweils mit Mittelwert \bar{x} und Standardabweichung σ).

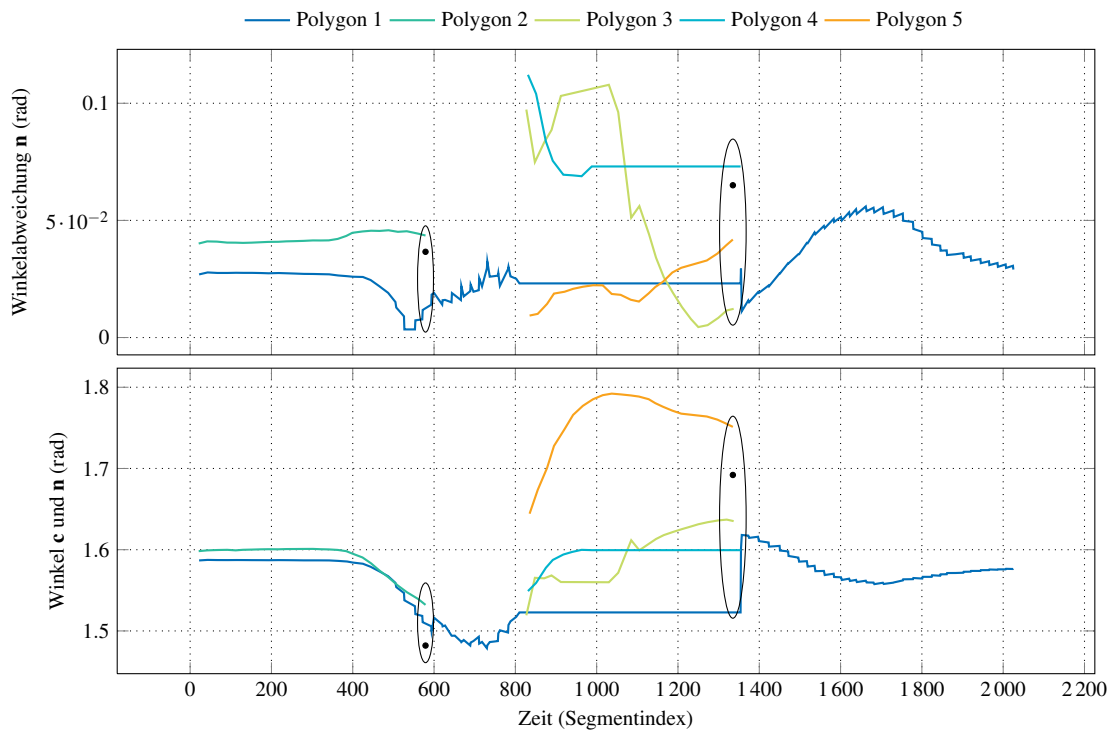


Abbildung 6.39: Merge-Sequenz für einen Karteneintrag

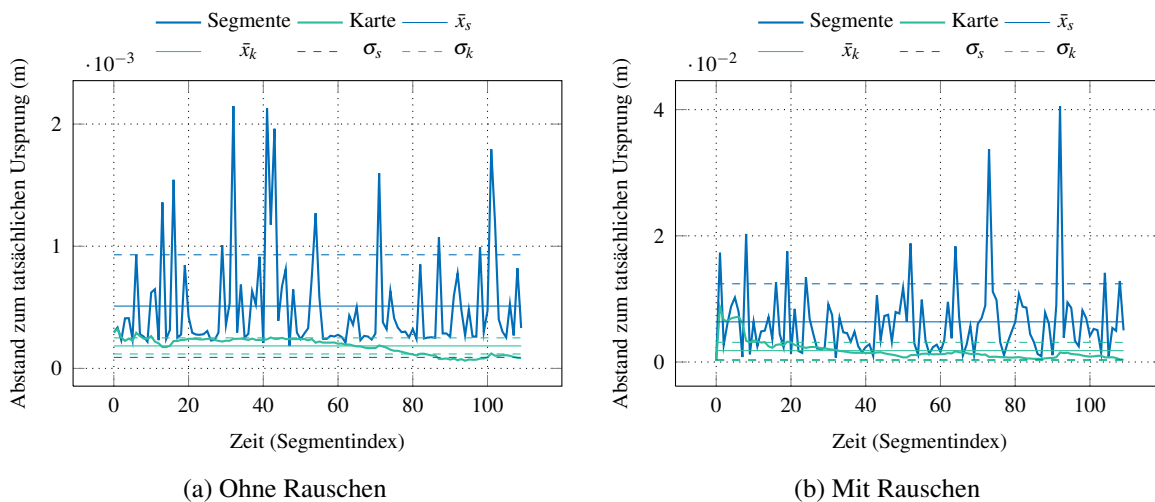


Abbildung 6.40: Abweichung des Ursprungs beim Zylinder (jeweils mit Mittelwert \bar{x} und Standardabweichung σ).

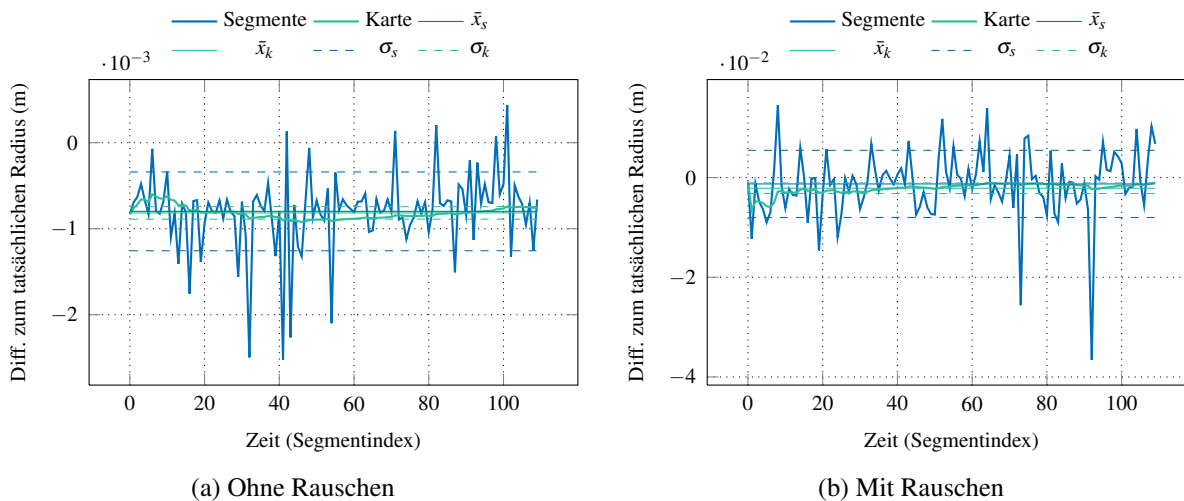


Abbildung 6.41: Abweichung des Radius beim Zylinder (jeweils mit Mittelwert \bar{x} und Standardabweichung σ)

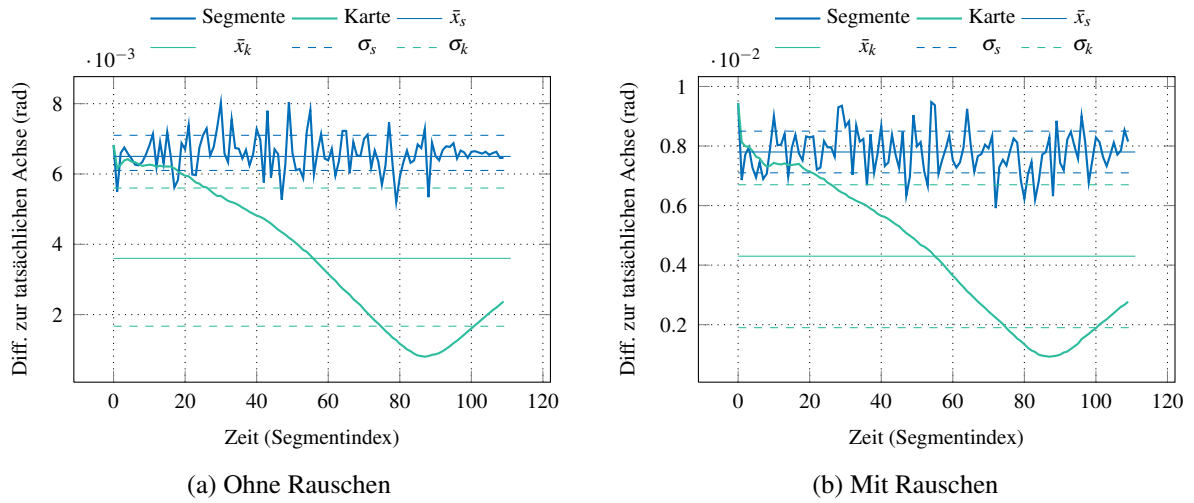


Abbildung 6.42: Abweichung der Symmetrieachse beim Zylinder (jeweils mit Mittelwert \bar{x} und Standardabweichung σ)

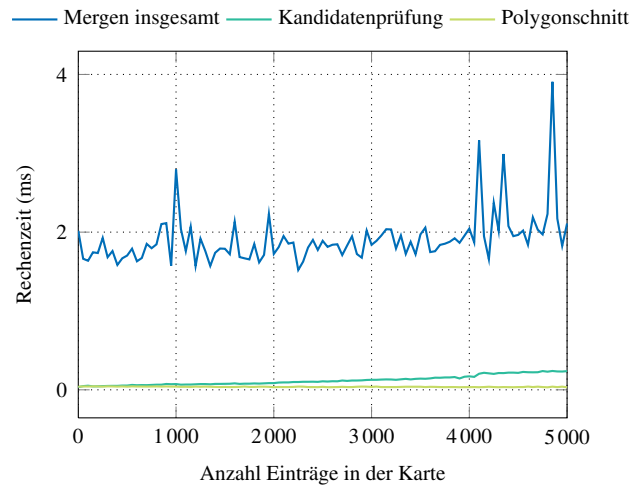


Abbildung 6.43: Rechenzeit für das Hinzufügen von Elementen in der GK

7 Verknüpfung der Einzelkomponenten und Validierung

Auf Basis der Ergebnisse von Kapitel 6 kann nun das vollständige Verfahren zur 3-D-Umgebungserfassung dargestellt werden. Es werden jeweils die Einzelkomponenten ausgewählt, die bei der Evaluierung am besten bewertet wurden. Nachfolgend wird die Übertragung des Umgebungsmodells über ein Netzwerk analysiert. Anschließend wird geprüft, ob durch das entwickelte Verfahren die Anforderungen eingehalten werden. Am Ende des Kapitels wird die Funktionalität des Verfahrens zur 3-D-Umgebungserfassung in Simulation und auf Care-O-bot[®] 3 validiert, bevor die Funktionsweise anhand eines Anwendungsbeispiels veranschaulicht wird.

7.1 Verknüpfung der Einzelkomponenten

Die Auswahl der Einzelkomponenten erfolgt aufgrund der Evaluierungsergebnisse in den entsprechenden Abschnitten. Für die Sensordatenvorverarbeitung wird eine Filterkaskade aus Abrißkanten-, Amplituden- und Specklefilter gewählt, da diese die besten Filtereigenschaften besitzt. Diese Vorverarbeitung ist jedoch nur dann nötig, wenn TOF-Kameras eingesetzt werden. Für Tiefenbildkameras wie *Microsoft Kinect* ist keine Filterung notwendig. Allerdings wird ein Passthrough-Filter (Rusu u. Cousins, 2011) eingesetzt, da die Kameras auch Messwerte außerhalb ihrer Maximalreichweite zurückgeben, die jedoch extrem verrauscht sind. Der Schwellwert für den Filter wird auf 5 m gesetzt.

Bei der Registrierung hat sich FRIR als leistungsfähigste Methode erwiesen. Mit einer Rechenzeit für einen Registrierungsschritt von 0,02 s ist es das mit Abstand schnellste Verfahren. Eine Vorverarbeitung in Form von Downsampling ist somit nicht nötig. Die Genauigkeitsuntersuchung bestätigt FRIR einen Genauigkeitsvorteil gegenüber dem Stand der Technik.

Für die geometrische Segmentierung zeigt sich GRG als das beste Verfahren bezüglich des Funktionsumfangs und der Genauigkeit. Innerhalb von GRG werden Punktnormalen unter Ausnutzung der geordneten Datenstruktur effizient berechnet (vgl. Kapitel 6.2) und *Principal Curvatures* als 3-D-Merkmale verwendet, da diese bei der Evaluierung in Kapitel 6.5 die besten Ergebnisse gezeigt haben.

Somit ergibt sich das endgültige Konzept des Verfahrens wie in Abbildung 7.1 dargestellt.

7.2 Übertragung des Umgebungsmodells über ein Netzwerk

Das Übertragungsverhalten des Umgebungsmodells im Netzwerk ist ein wichtiger Punkt für den teil-autonomen Betrieb. In Abschnitt 4.5 werden als Möglichkeiten die Kompression von 3-D-Daten oder die Wahl einer für die Übertragung geeigneten Kartenrepräsentation gezeigt. Mit dem hybriden Umgebungsmodell, bestehend aus 3-D-Gitter und Regelgeometriekarte ist bereits

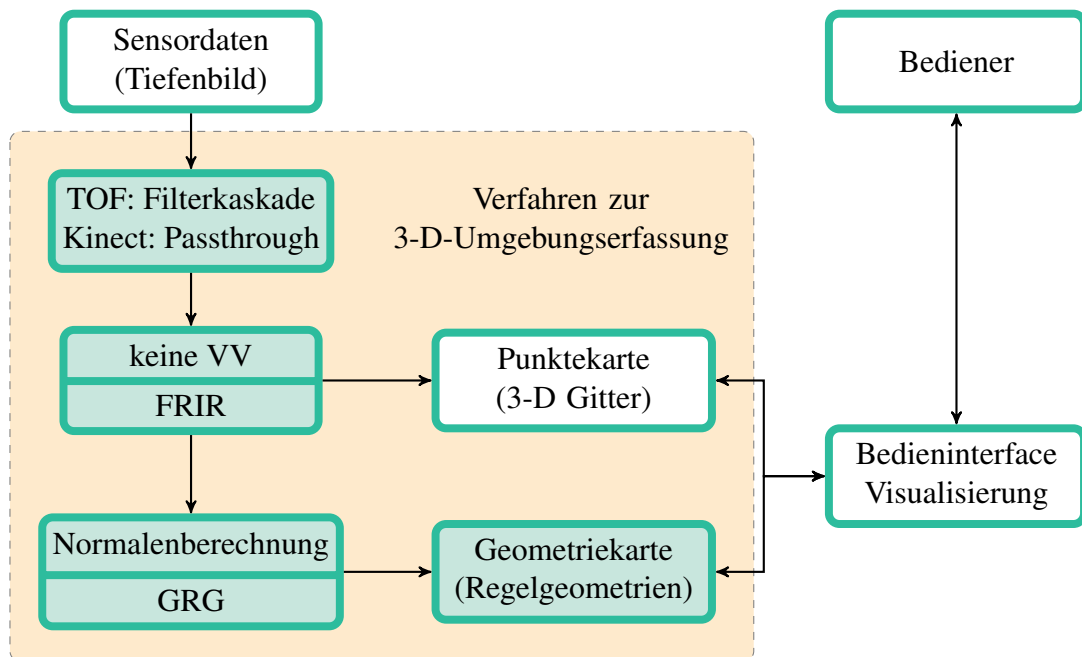


Abbildung 7.1: Endgültige Architektur für das System zur Umgebungserfassung

	Punktwolke	PK	GK
Anzahl Punkte	20 275 200	770 412	535
Datenvolumen (MB)	243,3	9,24	0,0065
Reduktion auf (%)	100	2,6	0,003

Tabelle 7.1: Datenreduktion der Kartenrepräsentationen

ein Kartenformat gegeben, welches gut für die Übertragung im Netzwerk geeignet ist. Vorzugsweise wird nur die GK zum Bediener übertragen, aber auch die PK kann für den Bediener relevante Information enthalten, deswegen soll auch das Übertragungsverhalten für das 3-D-Gitter untersucht werden.

Zunächst wird die Speicherreduktion zwischen Punktwolke und GK analysiert. Dazu wird eine Szene aus der Modellküche am Fraunhofer IPA mit insgesamt 66 registrierten Punktvolken gewählt. Alle Punktvolken zusammen enthalten 20 275 200 Punkte, welche bei drei 32-Bit-Werten pro Punkt insgesamt 243,3 MB Speicher benötigen. Farbige Punktvolken belegen sogar noch mehr Speicher, hier wird aber eine konservative Abschätzung durchgeführt. Eine Gitterkarte mit 1 cm Auflösung beinhaltet nur noch 770 412 Punkte oder 9,24 MB an Daten. Die GK schließlich besteht aus 7 Polygonen mit insgesamt 535 Hüllpunkten und je vier Parametern pro Polygon. Dies führt zu einer Kartengröße von 6,5 kB, was einer Reduktion auf 0,003% der Originalgröße entspricht (siehe auch Tabelle 7.1). Bei einer Aktualisierungszeit von 0,222 s ergibt sich somit eine benötigte Bandbreite von 35,8 kB/s für die GK und von 51,3 MB/s bei der Übertragung beider Karten. Die Übertragung der GK stellt also selbst in langsamen Netzwerken kein Pro-

blem dar. Sollen jedoch beide Karten übertragen werden, können schnell Engpässe entstehen, zumal die Karten mit der Zeit größer werden. Um dieses Problem zu lösen, kann differentielle Übertragung genutzt werden.

Die Idee hinter der differentiellen Übertragung ist, dass auf beiden Seiten ein vollständiges Umgebungsmodell vorgehalten wird. Dadurch müssen nur noch die Änderungen seit der letzten Aktualisierung übertragen werden. Die durchschnittliche Überlappung von aufeinander folgenden Punktwolken bei der Registrierung liegt bei einer Rotationsgeschwindigkeit der Kamera von 1,05 rad/s bei ungefähr 60%. Daraus folgt, dass pro Update ca. 40% der Datenmenge einer Punktwolke übertragen werden müssen. Obwohl diese Annahmen sehr konservativ sind (normalerweise bewegt sich die Kamera deutlich langsamer), ergibt sich damit ein zu übertragendes Datenvolumen pro Update von 286 kB für die PK. Dies entspricht einer benötigten Bandbreite von 1,59 MB/s, was in den meisten Netzwerken kein Problem darstellt. Für die GK ergibt sich eine geringere Einsparung, da die Einträge auch dann neu übertragen werden müssen, wenn sich nur ein kleiner Teil geändert hat. Unter der Annahme, dass sich maximal 20 Regelgeometrien geändert haben oder hinzugekommen sind, ergibt sich eine benötigte Bandbreite von ca. 10,5 kB/s. Dies ist im Vergleich zur PK vernachlässigbar gering.

7.3 Überprüfung der Anforderungen

Die in Kapitel 3 beschriebenen Anforderungen werden nachfolgend Merkmalen des entwickelten Verfahrens gegenübergestellt und deren Erfüllung überprüft. Durch den hybriden Ansatz der Kartenrepräsentation werden die Forderung nach einer Punktekarte (A1) und einer Geometrie-karte (A9) erfüllt. Diese Repräsentation bietet auch die Möglichkeit, Änderungen in der Karte zu erfassen (A4). Die Forderung nach einfacher Visualisierbarkeit (A5), Möglichkeit zur Interaktion (A8) und die Ableitbarkeit semantischer Information (A11) werden über die Darstellung in einer GK erfüllt, da die Einträge einzelne geometrische Basisformen darstellen.

Die benötigte Gesamtrechenzeit für das Verfahren ergibt sich aus der Addition der Zeiten für die Einzelkomponenten zu 0,222 s bzw. 0,23 s beim zusätzlichen Einsatz der Filterkaskade (siehe Tabelle 7.2). Damit wird die in den Anforderungen (A6) genannte Höchstdauer von 0,5 s deutlich unterschritten. Die Rechenzeit für die Erstellung der PK über die Registrierung von 0,02 s ist gering genug, um Anforderung A3 zu erfüllen. Der Vergleich mit den genauesten Algorithmen auf dem Stand der Technik (GICP und MPS) mit einer Ausführungszeit von 0,478 s ergibt, dass das neu entwickelte Verfahren mehr als doppelt so schnell rechnet.

Durch den absoluten mittleren Fehler von weniger als 0,3 cm stellt das gewählte Registrierungsverfahren die Einhaltung von Anforderung A2 sicher, wodurch die PK genau genug für Manipulationsaufgaben wird. Ein hoher Genauigkeitsgrad der geometrischen Information (A10) wird dadurch sicher gestellt, dass sowohl Registrierung als auch Segmentierung deutlich genauer rechnen als existierende Verfahren. Die ebenfalls von A10 geforderte Vollständigkeit der geometrischen Beschreibung wird dadurch erreicht, dass sowohl die Segmentierung als auch die GK mehr geometrische Basisformen darstellen können als Methoden auf dem Stand der Technik. Ebenso wird die gewünschte Robustheit (A12, A13) sowohl durch die im Rahmen dieser Arbeit entwickelten Teilkomponenten als auch durch den Einsatz von Rauschfiltern sichergestellt.

Komponente	Rechenzeit (s)	
Vorverarbeitung (Passthrough/Filterkaskade)	0,000	0,008
Registrierung	0,020	
Segmentierung	0,200	
Update Geometriekarte	0,002	
Gesamt	0,222	0,23

Tabelle 7.2: Gesamtrechenzeit als Summe der Zeiten der Einzelkomponenten

Schließlich bleibt noch die Überprüfung der Forderung nach Übertragbarkeit in einem Netzwerk (A7) und nach Modularität (A14). A14 ist bereits durch die Unterteilung des Verfahrens in eigenständige Einzelschritte abgedeckt. A7 wird erfüllt durch eine differentielle Übertragung der GK, wodurch lediglich eine Bandbreite von 10,5 kB/s benötigt wird.

Anforderung	Erfüllt	Durch folgende Maßnahmen
A1	✓	Repräsentation über 3-D-Gitter im hybriden Umgebungsmodell
A2	✓	Genauigkeit der Registrierung, absoluter Fehler < 0,3 cm
A3	✓	Geschwindigkeit der Registrierung von 0,02 s pro Tiefenbild
A4	✓	Repräsentation über 3-D-Gitter und Regelgeometrien
A5	✓	Repräsentation über Regelgeometriekarte
A6	✓	Geschwindigkeit des Gesamtsystems von 0,222 s pro Tiefenbild
A7	✓	Differentielle Übertragung der GK, benötigte Bandbreite von 10,5 kB/s
A8	✓	Repräsentation über Regelgeometriekarte, IDs in Karte
A9	✓	Repräsentation über Regelgeometriekarte im hybriden Umgebungsmodell
A10	✓	Genauigkeit der Registrierung und Segmentierung, Vollständigkeit der Segmentierung und GK besser als Stand der Technik
A11	✓	Repräsentation über Regelgeometriekarte
A12	✓	Registrierung und Segmentierung robuster als Stand der Technik, Mergen in GK robust gegenüber Rauschen
A13	✓	Rauschfilter, Registrierung und Segmentierung robuster als Stand der Technik, Mergen in GK robust gegenüber Rauschen
A14	✓	Unterteilung des Verfahrens in abgeschlossene Einzelschritte

Tabelle 7.3: Überprüfung der Erfüllung der Anforderungen

In Tabelle 7.3 sind die Anforderungen den in dieser Arbeit ausgearbeiteten Maßnahmen gegenübergestellt. Es ist zu erkennen, dass alle Anforderungen durch das entwickelte Verfahren erfüllt werden können.

7.4 Validierung der Funktionalität

In diesem Abschnitt erfolgt die Validierung der Funktionalität des entwickelten Verfahrens zur 3-D-Umgebungserfassung anhand von typischen Haushaltsszenen in der Simulation und auf Care-O-bot[®] 3.

7.4.1 Validierung in der Simulation

In der Simulationsumgebung Gazebo¹ wird die Funktionalität des Systems anhand einer Küchenszene überprüft. Die Sensordaten sind dabei optimal, weisen also kein Sensorrauschen auf. Der Roboter führt zwei 180° Schwenks mit unterschiedlichem Neigungswinkel des Kopfes aus und scannt so die Umgebung. Abbildung 7.2 zeigt die PK zu drei unterschiedlichen Zeitpunkten. Es ist zu sehen, dass die Registrierung die Punktwolken optimal ausrichtet. Die GK ist in Abbildung 7.3 zu den selben Zeitpunkten zu sehen. Horizontale Ebenen sind grün eingefärbt, vertikale gelb bzw. oliv. Im Bild ist zu erkennen, dass fast alle Ebenen der Szene in der richtigen Lage und Größe abgebildet werden.

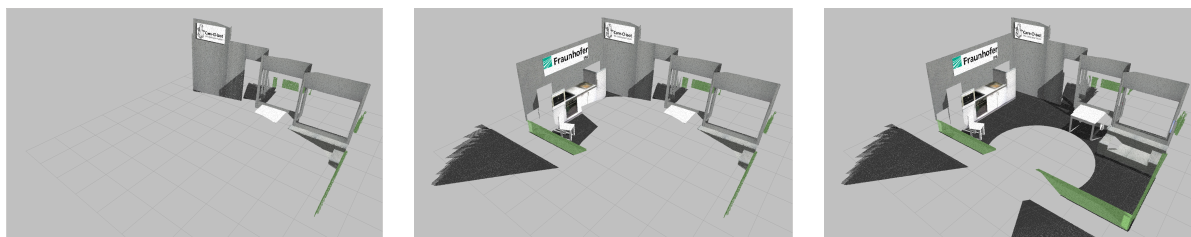


Abbildung 7.2: PK in der Simulation zu unterschiedlichen Zeitpunkten

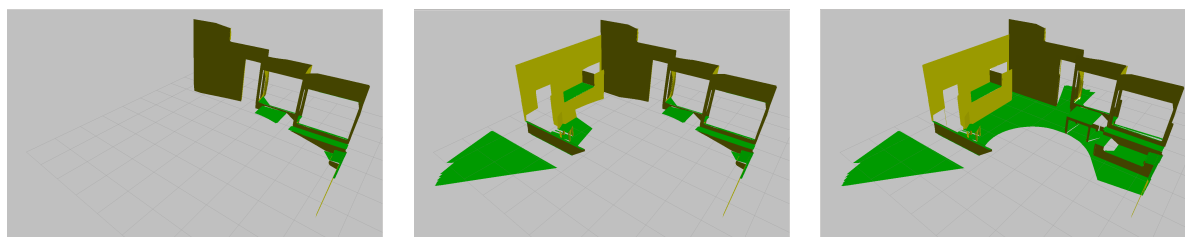


Abbildung 7.3: GK in der Simulation zu unterschiedlichen Zeitpunkten

7.4.2 Validierung auf Care-O-bot[®] 3

Zur Validierung der Funktionsweise auf dem Roboter werden zwei typische Haushaltsszenen betrachtet, eine Küchenzeile und ein Tisch mit Objekten (siehe Abbildung 7.4).

¹<http://www.ros.org/wiki/gazebo>



(a) Küchenszene

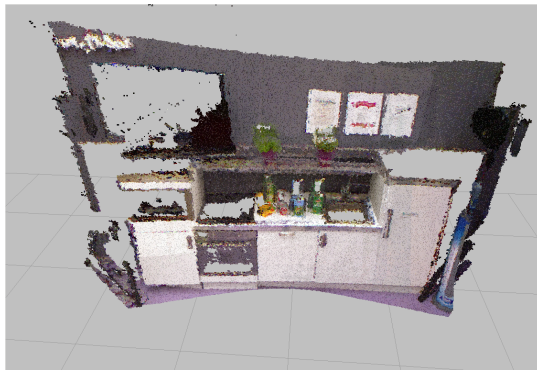


(b) Tischszene

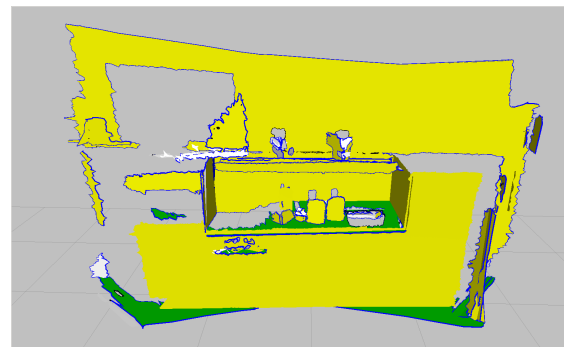
Abbildung 7.4: Zum Vergleich verwendete echte Szenen

Küchenszene

Anhand einer realen Küchenszene (Abbildung 7.4a) kann das Kartierungsergebnis mit echten Sensordaten überprüft werden. Der Roboter wird vor der Küchenzeile positioniert und schwenkt seinen Sensorkopf so, dass die gesamte Szene gescannt wird. In Abbildung 7.5 ist zu erkennen, dass sowohl die PK als auch die GK die Umgebung gut wiedergibt. Die meisten Flächen in der Szene werden durch konkave Ebenenpolygone in der GK abgebildet. Auch kleinere Objekte wie die Milchpackung werden noch separat beschrieben. Lediglich sehr kleine zylindrische Formen werden aufgrund von Sensorrauschen auch als Ebenen klassifiziert. Bei der geometrischen Karte sind wiederum horizontale Ebenen grün, vertikale gelb und sonstige weiß dargestellt.



(a) PK



(b) GK

Abbildung 7.5: Umgebungsmodell der Küchenszene

Tischszene

Weiterhin soll eine Tischszene untersucht werden, um insbesondere die Abbildung von Zylindern zu zeigen. In der gewählten Szene (Abbildung 7.4b) fährt der Roboter um einen Tisch, auf

dem sich zwei zylindrische Objekte befinden. In Abbildung 7.6 ist zu sehen, dass die PK die Szene korrekt wiedergibt. Allerdings entstehen kleinere Abweichungen insbesondere an Kanten. Dies hängt mit der großen Bewegung des Roboters zusammen, die durch die Registrierung nicht vollständig kompensiert werden kann. In der GK sind sowohl die Ebenen als auch die Zylinder fehlerfrei und die Szene insgesamt vollständig dargestellt. Für diese Szene wurden die Farbwerte der Punktwolke verwendet, um die Einträge der GK einzufärben.

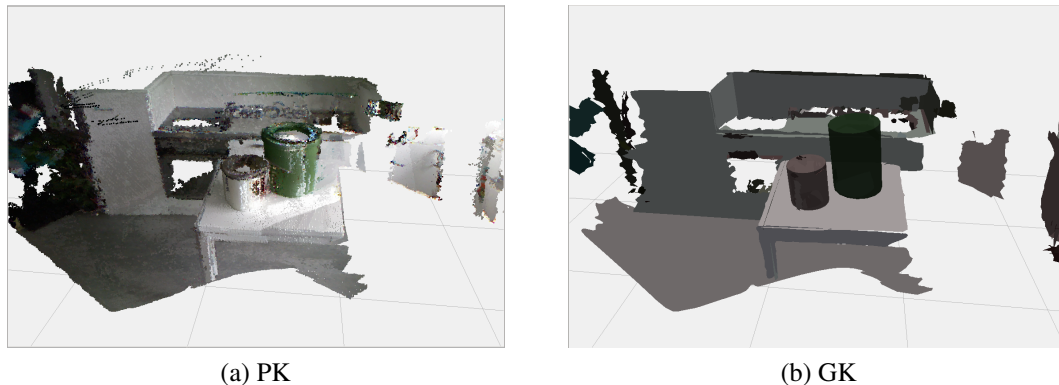


Abbildung 7.6: Umgebungsmodell für die Tischszene

7.5 Anwendungsbeispiel „Hol- und Bringdienst“

Abschließend soll die Funktionsweise des Verfahrens zur 3-D-Umgebungserfassung am Beispiel eines interaktiven „Hol- und Bringdienstes“ veranschaulicht werden. Die Lösung der Aufgabe verdeutlicht die Eignung des Verfahrens zur Manipulation, zur Visualisierung und Interaktion und zur Planung von Aktionen. Der Roboter soll in diesem Beispiel ein Objekt von einem Tisch holen (Abbildung 7.7a). Zur Lageerkennung der Objekte auf Tischen wird ein Verfahren eingesetzt, bei dem Objektmodelle eingelernt und im Bild wiedererkannt werden können. Dieses Verfahren ist auf eine Erkennungsreichweite von 1 – 2 m beschränkt.

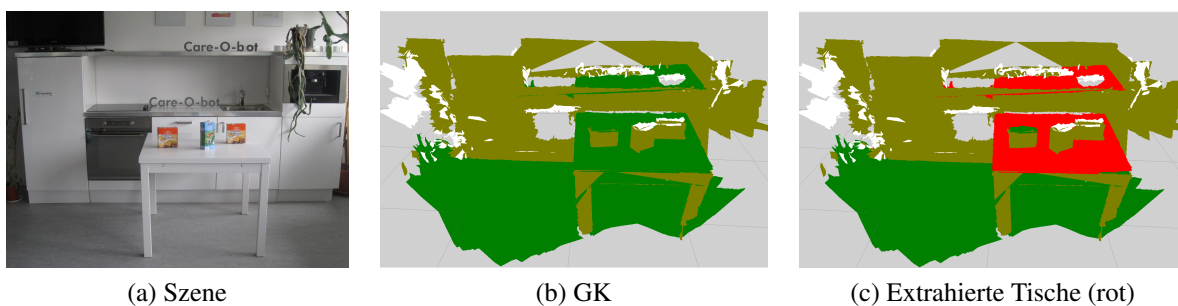


Abbildung 7.7: Erster Teil des Szenarios



Abbildung 7.8: Zweiter Teil des Szenarios

Die Position des Tisches ist ungefähr bekannt, es befinden sich aber mehrere Tische nahe dieser Position. Zunächst fährt Care-O-bot[®] 3 autonom zu einer Position, von der aus der Tisch im Blickfeld der Sensoren ist. Dann wird durch Bewegen des Torsos die Umgebung erfasst und eine GK erzeugt (Abbildung 7.7b). Im nächsten Schritt kann ein Tischerkennungsverfahren eingesetzt werden, um aus den in der Karte befindlichen horizontalen Ebenen die Maße und Position des Tisches zu bestimmen. Für eine präzise Erkennung der Objekte auf den Tischen ist der Roboter aber zu weit entfernt. Die erkannten Tische werden in Abbildung 7.7c als rote Ebenen eingeblendet. Aus der GK wird also semantische Information generiert, die zur Planung eingesetzt werden kann. Der Benutzer kann nun mit den visualisierten Karteneinträgen interagieren, indem er auf den richtigen Tisch klickt und somit dem Roboter den Befehl gibt, zum Tisch zu fahren (Abbildung 7.8a).

Nachdem Care-O-bot[®] 3 autonom zum Tisch navigiert hat, erzeugt er eine PK als Kollisionskarte der Umgebung (Abbildung 7.8b). Gleichzeitig werden auf dem Tisch befindliche Objekte erfasst und visualisiert (Abbildung 7.8c). Der Bediener kann nun überprüfen, ob die Kollisionskarte korrekt ist und klickt danach auf eines der Objekte. Dies teilt dem Roboter mit, welches Objekt gegriffen werden soll.

Ohne die Möglichkeit zum teil-autonomen Modus wäre dieses Szenario nur teilweise oder gar nicht lösbar. Zum einen gibt es mehrere Tische, der Roboter müsste also autonom entscheiden, zu welchem er sich bewegen soll. Dies würde entweder die Fähigkeit bedingen, Objekte bereits aus großer Entfernung zu erkennen. In diesem Fall wäre es nicht so wichtig, auf welchem Tisch das gewünschte Objekt steht. Oder der Roboter müsste über die nötige Information und Intelligenz verfügen, selbständig den richtigen Tisch zu wählen. Käme es bei der Erstellung der Kollisionskarte zu Fehlern, wäre der Roboter nicht in der Lage, das entsprechende Objekt kollisionsfrei zu greifen. Die einzige Möglichkeit wäre dann, die Karte in der Hoffnung, dass diesmal keine Fehler auftreten, neu zu erstellen oder das Szenario abubrechen. Mit der visuellen Kontrollmöglichkeit des Bedieners wird solch ein Fall verhindert. Schließlich kann es sein, dass sich mehrere ähnliche oder identische Objekte auf dem Tisch befinden. Mit der eingesetzten Objekterkennung wäre es nicht möglich, das korrekte Objekt zu bestimmen. In diesem Fall könnte Care-O-bot[®] 3 nur eines der Objekte zufällig auswählen. Ein menschlicher Bediener hingegen kann in dieser Situation das korrekte Objekt auswählen.

8 Zusammenfassung und Ausblick

Multifunktionale mobile Serviceroboter haben ein großes Potenzial zur autonomen Erfüllung vielfältiger Aufgaben in Haushalt und Gewerbe. Schon jetzt gibt es an spezielle Aufgaben angepasste Produkte, die diese zuverlässig erfüllen. Die Wahrnehmung und Analyse der Umgebung ist eine essentielle Technologie dafür, den Funktionsumfang mobiler Roboter deutlich zu erweitern und zur Produktreife zu bringen. Mit der Einführung kostengünstiger 3-D-Tiefenbildkameras sind die Voraussetzungen für eine leistungsfähige Umgebungserfassung, zumindest in Innenräumen, gegeben. Trotz aller Fortschritte in den Bereichen Hardware und Software werden mittelfristig die Fähigkeiten der Roboter jedoch nicht ausreichen, um alle Aufgaben und insbesondere Ausnahmesituationen erfolgreich und vor allem autonom zu meistern.

Aus diesen Gründen wurde im Rahmen dieser Arbeit ein Verfahren zur 3-D-Umgebungserfassung für teil-autonome mobile Roboter entwickelt. Die grundlegenden Eigenschaften dieses Verfahrens sind eine Eignung für eine kollisionsfreie Navigation und Manipulation, für die Visualisierung und Interaktion auf der Bedienerseite und für die Planung von Aktionen. Auf Basis von 3-D-Tiefenbildkameras werden 3-D-Karten von teil-dynamischen Innenraumumgebungen erstellt und in eine geeignete Kartenrepräsentation überführt.

Eine Betrachtung der Ausgangssituation zeigte, dass sich nicht alle Anwendungsgebiete für multifunktionale mobile Serviceroboter gleichermaßen gut als Schwerpunkt für das zu entwickelnde Verfahren eignen. Aufgrund der Eigenschaften auf existierenden Robotern eingesetzter Sensorik und den Umgebungsbedingungen der Anwendungsfelder wurde der Schwerpunkt der vorliegenden Arbeit auf ein Verfahren zur 3-D-Umgebungserfassung in Innenraumumgebungen unter Verwendung von 3-D-Tiefenbildkameras gelegt. Auf dieser Basis wurde der Stand der Technik mit Fokus auf ebensolche Kamertypen sowie bestehende Systeme und Verfahren in der Teleoperation von Robotern und in der Umgebungserfassung untersucht. Die Analyse zeigte vor allem Lücken bei wichtigen Einzeltechnologien für die Umgebungserfassung wie Registrierung und Segmentierung sowie bei der ganzheitlichen Repräsentation des Umgebungsmodells auf.

Mit den Ergebnissen der Recherche konnte ein Konzept für das Verfahren zur 3-D-Umgebungserfassung erarbeitet werden. Ein Kernpunkt war dabei die Ableitung einer für alle Anforderungen geeigneten Kartenrepräsentation, welche in ein hybrides Umgebungsmodell, bestehend aus einer Punkte- und einer Geometriekarte mündete. Weiterhin wurden die wichtigsten Schritte zum Aufbau der Repräsentation identifiziert. Nachfolgend wurden diese Schritte im Detail untersucht und wo nötig neue Methoden entwickelt. Das resultierende Verfahren zur 3-D-Umgebungserfassung besteht aus einer sequentiellen Verarbeitungskette zum Aufbau eines Umgebungsmodells. Zunächst erfolgt je nach eingesetztem Kamertyp eine Rauschfilterung der Daten, bei TOF-Kameras beispielsweise in einer Kaskade von Einzelfiltern. Der neu entwickelte Specklefilter ermöglicht dabei eine effiziente Entfernung von Gaußschem Rauschen. Die gefilterten Punktwolken werden im nächsten Schritt registriert, um konsistente Karten erzeugen zu können. Im Rahmen dieser Arbeit wurde ein Registrierungsverfahren konzipiert und implementiert, welches deutlich schneller und genauer rechnet als existierende Algorithmen. Aus den registrierten

Punktwolken werden nachfolgend geometrische Basisgeometrien extrahiert. Hierzu erfolgt die Unterteilung der Daten in homogene Segmente durch ein neues Verfahren, das graph-basiertes *Region Growing* über die Normalenvektoren durchführt und somit parametrisch beschreibbare Oberflächen extrahiert. Zur Berechnung der Normalenvektoren wurden effiziente Verfahren zur Nachbarschaftssuche und zur Bestimmung der Vektoren entwickelt, die die geordnete Struktur von Tiefenbildern ausnutzen. Der Segmentierungsalgorithmus hebt sich vor allem in Bezug auf Genauigkeit und mit der Möglichkeit zur Beschreibung gekrümmter Oberflächen von existierenden Verfahren ab. Zum Aufbau einer Geometriekarte werden die Segmente als Hüllpolygone beschrieben und mehrere Ansichten des gleichen Objekts über die Zeit fusioniert. Der Aufbau einer Geometriekarte wurde für Ebenen- und Zylindersegmente gezeigt.

Alle Einzelkomponenten wurden auf typischen Innenraumszenen sowohl im Bezug auf die Anforderungen als auch im Vergleich zu existierenden Verfahren evaluiert. Die Ergebnisse zeigen, dass die Neuentwicklungen zum einen bei den meisten Kriterien besser abschneiden als der Stand der Technik und zum anderen in der Lage sind, die Anforderungen, insbesondere in Bezug auf Genauigkeit und Rechenzeit zu erfüllen. Ebenfalls überprüft wurde das Übertragungsverhalten des Umgebungsmodells im Netzwerk mit dem Resultat, dass eine ausreichend schnelle Übertragung zum Bediener auch bei geringen Bandbreiten möglich ist. Schließlich wurde die Funktionsweise des entwickelten Verfahrens anhand typischer Haushaltsszenen validiert und durch ein Anwendungsbeispiel verdeutlicht.

Obwohl sich das entwickelte Verfahren als leistungsfähig herausgestellt hat, gibt es noch einige Punkte, mit denen sich nachfolgende Forschungsarbeiten befassen können. Zunächst wäre es sicher hilfreich, die bisher nicht geometrisch beschreibbaren komplexeren Formen zu behandeln. Dies könnte durch weitere Verarbeitungsschritte der komplexen Segmente geschehen. Dies steht in direktem Zusammenhang mit der Ableitung höherwertiger semantischer Information. Diese ist wichtig, um das Szenenverständnis des Roboters zu erhöhen. Je mehr geometrische Objekte in der Karte identifiziert werden können, desto besser kann der Roboter seine Aufgaben erfüllen. Die semantische Information könnte direkt mit einem abstrakten Planungssystem für Roboteraktionen verknüpft werden. Dadurch wäre es dem Roboter möglich, von einfachen Befehlen wie „Hole die Milch“ darauf zu schließen, dass die Milch wahrscheinlich im Kühlschrank in der Küche ist. Dieses Wissen könnte durch Information aus dem Umgebungsmodell erweitert werden.

Weitere Forschungsarbeiten könnten sich mit der Kartierung von hochdynamischen Szenen befassen, wie sie in manchen Manipulationsaufgaben auftreten. Eine Erweiterung des Verfahrens für die Behandlung von schnellen Änderungen, z. B. wenn sich Menschen in der unmittelbaren Umgebung des Roboters aufhalten, wäre sinnvoll und würde die Leistungsfähigkeit erhöhen. Viele Teilkomponenten wie die Registrierung und die Segmentierung sind schon dafür geeignet. Jedoch müsste die Kartenverwaltung angepasst werden. Eine zusätzliche Weiterentwicklung wäre der Aufbau eines 3-D-SLAM Systems aus einigen Einzelkomponenten. Damit wäre es zum einen möglich, auf die Lokalisierung per Laserscanner zu verzichten und dadurch das Verfahren auch auf Robotern einzusetzen, die nur eine Tiefenbildkamera als Sensor aufweisen. Zum anderen könnten auch größere Areale konsistent kartiert und durch Erkennung des Kreisschlusses eine globale Optimierung der Karten durchgeführt werden.

A Evaluierung von Klassifizierungsproblemen

Für die Evaluierung von Klassifizierungsproblemen werden für gewöhnlich die Erwartungswerte mit den vorhergesagten Werten verglichen. Um Erwartungswerte zu erhalten, werden die Klassen in einem Datensatz manuell bestimmt und diese dann als Grundwahrheit verwendet. Nachfolgend wird das zu evaluierende Verfahren auf den Datensatz angewandt und für jede Klasse gezählt, welche Instanzen richtig und welche falsch klassifiziert wurden. Dies resultiert dann in einer Aufstellung von *wahr Positiven* (*true positives*, TP), *falsch Positiven* (*false positives*, FP), *wahr Negativen* (*true negatives*, TN) und *falsch Negativen* (*false negatives*, FN) für jede Klasse (Olson u. Delen, 2008). TP liegen beispielsweise dann vor, wenn für einen Wert die vorhergesagte der tatsächlichen Klasse entspricht. Die Werte lassen sich in einer Konfusionsmatrix eintragen, wie sie in Abbildung A.1 für zwei Klassen dargestellt ist. Eine Konfusionsmatrix gibt zwar einen guten Eindruck der Klassifizierungsgenauigkeit, es ist jedoch schwierig, auf diesem Wege unterschiedliche Verfahren zu vergleichen. Deswegen können aus den Einträgen der Matrix die Werte

- *precision* P als das Verhältnis von korrekt positiv bestimmter zu allen als positiv klassifizierten Instanzen (d. h. $P = 1,0$ bedeutet, dass alle vorhergesagten Positiven ein „hit“ sind),
- *recall* R als das Verhältnis korrekt positiv bestimmter zu allen positiv erwarteten Instanzen (d. h. $R = 1,0$ bedeutet, dass alle erwartet Positiven gefunden wurden) und
- *F-score* F_β als das harmonische Mittel aus P und R mit dem Gewichtungsfaktor β zwischen P und R

berechnet werden, welche sich unmittelbar vergleichen lassen.

Für Klassifizierungsprobleme mit mehr als zwei Klassen reichen die eben beschriebenen Werte aber nicht aus. Für die Evaluierung solcher Probleme bietet sich die Aufstellung der *micro* und *macro average* Werte an, die die Einträge in der Konfusionsmatrix A_{ij} für mehrere Klassen zusammenfassen (Manning u. a., 2008). Diese Werte werden wieder für *precision*, *recall* und *F-score* gebildet. Demnach ergeben sich als *micro average* Werte für k Klassen

$$R_{\text{mic}} = P_{\text{mic}} = F_{\text{mic}} = \frac{\sum_i^k A_{ii}}{\sum_i^k \sum_j^k A_{ij}} = \frac{\sum_i^k TP_i}{\sum_i^k (TP_i + FN_i)} \quad (\text{A.1})$$

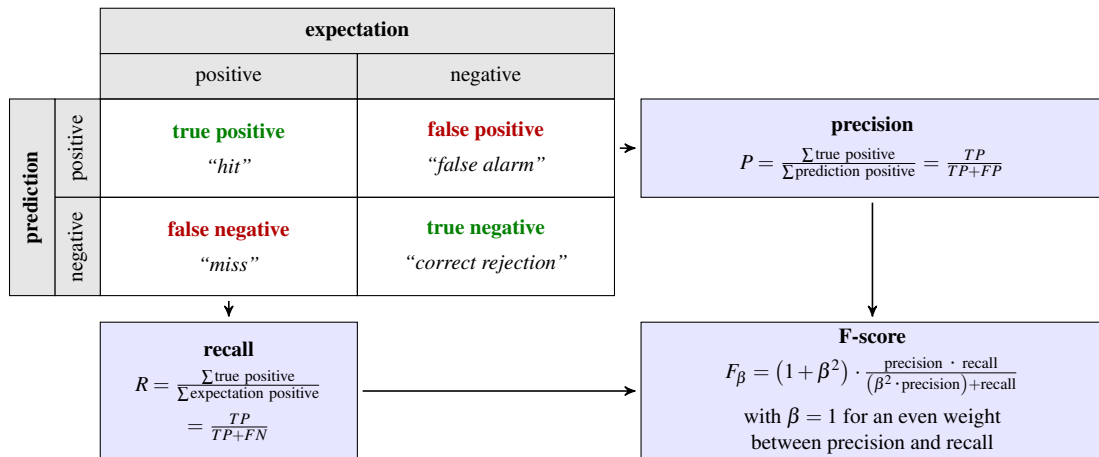


Abbildung A.1: Maße für die Evaluierung von Zweiklassenproblemen

und als *macro average* Werte

$$R_{\text{mac}} = \frac{1}{k} \sum_i^k \frac{A_{ii}}{\sum_j^k A_{ij}} = \frac{1}{k} \sum_i^k \frac{TP_i}{TP_i + FN_i} \quad (\text{A.2})$$

$$P_{\text{mac}} = \frac{1}{k} \sum_i^k \frac{A_{ii}}{\sum_j^k A_{ji}} = \frac{1}{k} \sum_i^k \frac{TP_i}{TP_i + FP_i} \quad (\text{A.3})$$

$$F_{\text{mac}} = (1 + \beta^2) \cdot \frac{P_{\text{mac}} \cdot R_{\text{mac}}}{(\beta^2 \cdot P_{\text{mac}}) + R_{\text{mac}}} \quad (\text{A.4})$$

Für ein optimales Klassifizierungsergebnis würden alle Größen den Wert 1 annehmen. Anschaulich gibt der *micro average* klassenunabhängige Werte wieder, wohingegen der *macro average* den Mittelwert der Einzelwerte jeder Klasse berechnet.

B Datensatz für die Evaluierung

Für die Evaluierung von Merkmalsextraktion und Segmentierung wurde ein Datensatz aus acht Innenraumszenen verwendet. Es gibt vier Setups, nämlich Küche (*kitchen*), Tisch (*table*), Büro (*office*) und Regal (*cupboard*). Jede dieser Anordnungen wurde aus der Nähe (*close*) und aus der Ferne (*far*) aufgenommen. Somit ergeben sich die acht Szenen, wie sie in Abbildung B.2 zu sehen sind. Für jede dieser Szenen wurde die aufgenommene Punktwolke manuell nach Oberflächentypen (Ebene, Kante, Zylinder, Kugel, Ecke und undefiniert) klassifiziert. Da es sich bei den aufgenommenen Daten um Tiefenbilder handelt, können diese in ein Grafikbearbeitungsprogramm geladen und dort pixel- oder regionsweise farblich klassifiziert werden. Die Farbcodes für die Oberflächentypen sind in Tabelle B.1 aufgelistet.

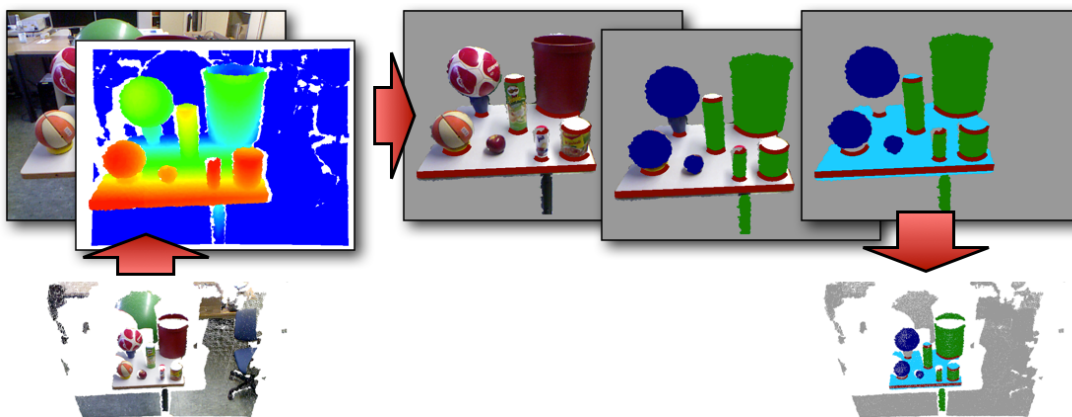


Abbildung B.1: Vorgehensweise beim manuellen Klassifizieren

Abbildung B.1 zeigt die Vorgehensweise beispielhaft. Zunächst wird das Tiefenbild in Tiefenrichtung farbkodiert. Somit kann der Hintergrund und Bereiche mit ungünstigen Tiefenwerten

Klasse	Farbe
Ebene	■
Kante	■
Zylinder	■
Kugel	■
Ecke	■
undefiniert	■

Tabelle B.1: Farbcodes für die Oberflächentypen

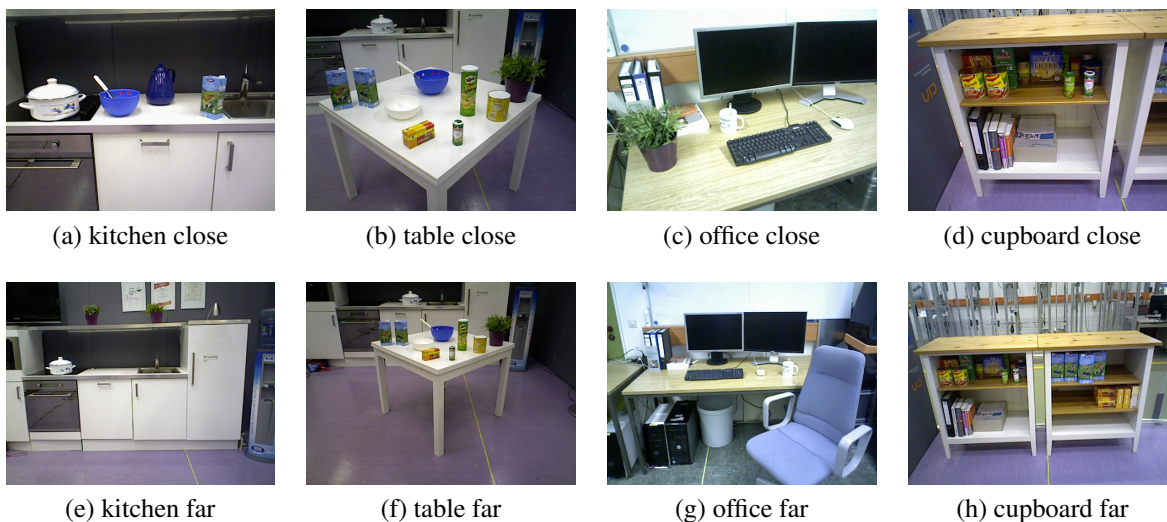


Abbildung B.2: Farbbilder der Szenen

leicht entfernt werden. Danach werden alle Kanten rot und alle Ecken gelb eingefärbt. Die Tiefeninformation wird anschließend zusätzlich mit einbezogen, um Objektgrenzen von Kugeln und Zylindern zu finden. Schließlich werden alle verbleibenden Regionen als Ebenen markiert. Die Klassifizierung wird im Farbkanal des Tiefenbilds abgelegt. Abbildung B.3 zeigt die klassifizierte Szenen.

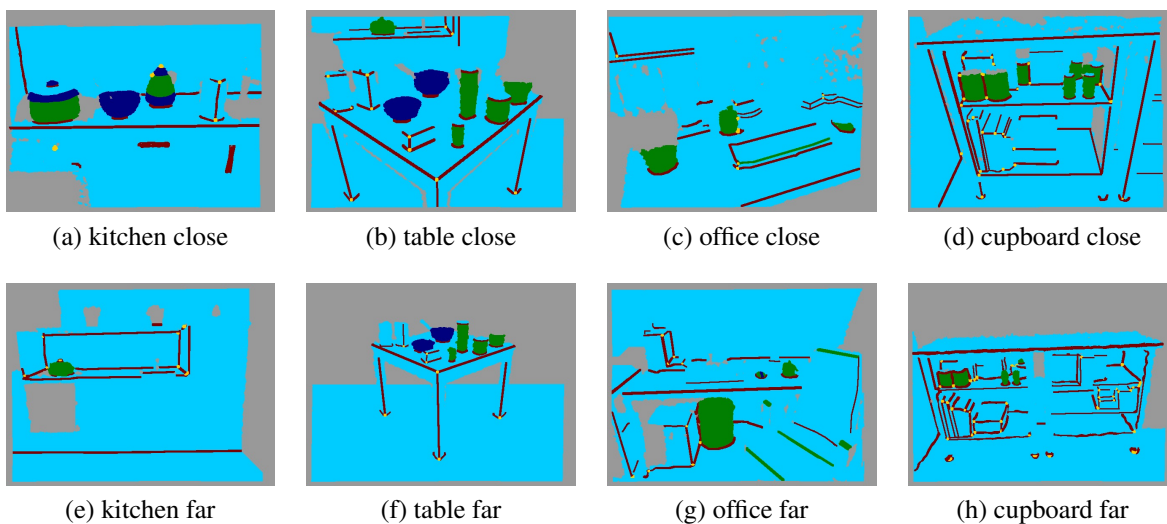


Abbildung B.3: Manuell klassifizierte Szenen

C Principal Component Analysis (PCA)

Die Hauptkomponentenanalyse (*Principal Component Analysis*, PCA) wird im Rahmen dieser Arbeit an verschiedenen Stellen benötigt, sei es bei der Normalenberechnung, bei der Merkmalsextraktion oder bei der Segmentierung. Deswegen sollen hier die Grundlagen dieser ursprünglich in (Pearson, 1901) veröffentlichten Methode beschrieben werden.

Ziel einer PCA ist es, die Datenpunkte eines Datensatzes so in einen Unterraum zu projizieren, dass dabei wenig vorliegende Information verloren geht und vorliegende Redundanz in Form von Korrelation in den Datenpunkten zusammengefasst wird. Verwendet wird dabei eine Hauptachsentransformation zur Minimierung der Korrelation. Die Hauptachsentransformation bewirkt, dass die Kovarianzmatrix \mathbf{C} des Datensatzes diagonalisiert wird. Der erste Schritt der Vorgehensweise liegt darin, die vorliegenden Daten mit der Dimension d auf ihren Mittelwert zu beziehen. Das bedeutet, für alle Punkte \mathbf{p}_i eines Datensatzes (z. B. einer Punktwolke) mit n Instanzen wird der Mittelwert

$$\bar{\mathbf{p}} = \frac{\sum_{i=1}^n \mathbf{p}_i}{n} \quad (\text{C.1})$$

berechnet und dann die Differenz $\mathbf{p}_i - \bar{\mathbf{p}}$ gebildet. Anschließend kann die Kovarianzmatrix des Datensatzes zu

$$\mathbf{C} = \frac{1}{n} \sum_{i=1}^n (\mathbf{p}_i - \bar{\mathbf{p}})^T \cdot (\mathbf{p}_i - \bar{\mathbf{p}}) \quad (\text{C.2})$$

berechnet werden. Im nächsten Schritt können die d Eigenwerte λ_j und die zugehörigen Eigenvektoren \mathbf{v}_j durch Lösen der Gleichung

$$\mathbf{C} \cdot \mathbf{v}_j = \lambda_j \mathbf{v}_j \quad (\text{C.3})$$

bestimmt werden. Die Berechnung der Eigenwerte erfolgt normalerweise über die Bestimmung der Determinante $\det(\mathbf{C} - \lambda \mathbf{I})$ und Lösen des resultierenden Gleichungssystems. Die Eigenvektoren können dann durch Einsetzen der Eigenwerte in (C.3) abgeleitet werden. Der kleinste Eigenwert korrespondiert mit der kleinsten Varianz des Datensatzes. Der zugehörige Eigenvektor definiert die Richtung der Varianz.

Diese Beziehungen werden im Rahmen dieser Arbeit meist verwendet, um geometrische Schlüsse über den Datensatz zu ziehen. Soll beispielsweise der Normalenvektor einer Ebene gefunden werden, so liegt dieser in Richtung der kleinsten Varianz der Punktwolke, da eine Ebene ihre Ausdehnung senkrecht zum Normalenvektor hat. Folglich können über eine PCA zunächst die Eigenwerte berechnet werden. Der zum kleinsten Eigenwert gehörige Eigenvektor definiert dann den Normalenvektor der Ebene.

D Basisgeometrien

Die in der GK enthaltenen Basisgeometrien müssen eindeutig beschrieben werden, damit eine konsistente Karte entstehen kann. Eine erfolgreiche Fusion von Segmenten ist nur dann möglich, wenn diese im selben Format vorliegen. Weiterhin muss die Beschreibung vollständig sein. Da mit dem entwickelten Verfahren Karten, welche aus Ebenenpolygonen und Zylindern bestehen, erzeugt werden können, soll die Beschreibung für diese zwei Geometrien im Folgenden definiert werden.

D.1 Ebene

Eine Ebene e kann über die Parameterform

$$e : ax + by + cz + d = 0 \quad (\text{D.1})$$

oder über die Normalform

$$e_n : \mathbf{n} \cdot (\mathbf{x} - \mathbf{a}) = 0 \quad (\text{D.2})$$

mit dem beliebigen Punkt auf der Ebene \mathbf{a} beschrieben werden. Für den Normalenvektor \mathbf{n} gilt dabei

$$\mathbf{n} = (a \ b \ c)^T. \quad (\text{D.3})$$

In dieser Arbeit wird meist die Parameterform verwendet. Die vier Parameter beschreiben eine Ebene vollständig. Allerdings besitzen die Segmente Grenzen, die sich über ein Konturpolygon ausdrücken lassen. Das Konturpolygon C lässt sich durch eine geordnete Liste von Punkten \mathbf{p} beschreiben. Die Kontur kann konkav oder konvex sein. Weiterhin kann das Polygon Löcher besitzen, was durch zusätzliche negative Konturen beschrieben wird. Zusätzlich ist häufig noch der Schwerpunkt \mathbf{c} des Ebenenpolygons von Bedeutung, welcher sich aus der Kontur mit n gleichmäßig verteilten Punkten gemäß

$$\mathbf{c} = \frac{1}{n} \sum_{i=1}^n \mathbf{p}_i \quad (\text{D.4})$$

berechnen lässt. Abbildung D.1 zeigt die Definition des Ebenenpolygons.

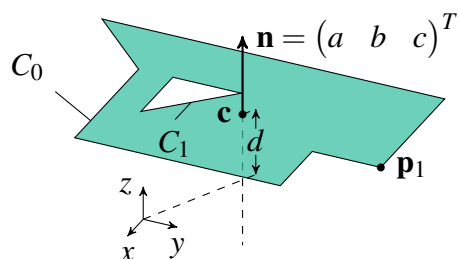


Abbildung D.1: Definition eines Ebenenpolygons

D.2 Zylinder

Ähnlich wie die Ebene lässt sich auch ein Zylinder über Parameter beschreiben, wobei es jedoch nicht möglich ist, eine geschlossene Beschreibung in Form einer Gleichung anzugeben. Die Zylinderparameter sind der Ursprung \mathbf{o} , der Radius r , die Symmetrieachse \mathbf{s} und die Höhe h . Genauso wie bei der Ebene liegen die Zylinder häufig nur unvollständig vor. Diesem Aspekt wird wiederum die Definition einer Kontur gerecht. Diese ist jedoch anders als bei der Ebene dreidimensional. Der Ursprung liegt auf Höhe des Konturschwerpunkts \mathbf{c} . Die Beziehungen sind in Abbildung D.2 dargestellt.

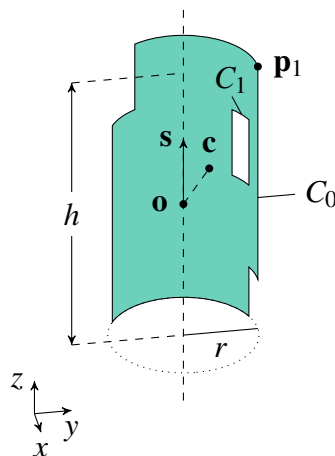


Abbildung D.2: Definition eines Zylinders

Literaturverzeichnis

- [Akca 2007] AKCA, Devrim: Matching of 3D surfaces and their intensities. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 62 (2007), Nr. 2, S. 112–121. – DOI 10.1016/j.isprsjprs.2006.06.001
- [Alexa u. a. 2003] ALEXA, M. ; BEHR, J. ; COHEN-OR, D. ; FLEISHMAN, S. ; LEVIN, D. ; SILVA, C.T.: Computing and rendering point set surfaces. In: *IEEE Transactions on visualization and computer graphics* (2003), S. 3–15
- [Aracil u. a. 2007] ARACIL, Rafael ; BUSS, Martin ; COBOS, Salvador ; FERRE, Manuel ; HIRCHE, Sandra ; KUSCHEL, Martin ; PEER, Angelika: The Human Role in Telerobotics. In: FERRE, Manuel (Hrsg.) ; BUSS, Martin (Hrsg.) ; ARACIL, Rafael (Hrsg.) ; MELCHIORRI, Claudio (Hrsg.) ; BALAGUER, Carlos (Hrsg.): *Advances in Telerobotics*. Berlin, Heidelberg : Springer, 2007 (Springer Tracts in Advanced Robotics 31), S. 11–24. – DOI 10.1007/978-3-540-71364-7_2
- [Arbeiter u. a. 2012a] ARBEITER, Georg ; BORMANN, Richard ; HÄGELE, Martin ; VERL, Alexander: Towards Geometric Mapping for Semi-autonomous Mobile Robots. In: STACHNISS, Cyrill (Hrsg.) ; SCHILL, Kerstin (Hrsg.) ; UTTAL, David (Hrsg.): *Spatial Cognition VIII*. Springer, 2012 (Lecture Notes in Artificial Intelligence 7463), S. 114–127
- [Arbeiter u. a. 2010] ARBEITER, Georg ; BUBECK, Alexander ; FISCHER, Jan ; GRAF, Birgit: Teleoperierte mobile Roboter warten Prozessanlagen : Niedrigere Betriebskosten und höhere Arbeitssicherheit. In: *atp-edition* 52 (2010), Nr. 5, S. 44–51
- [Arbeiter u. a. 2012b] ARBEITER, Georg ; FUCHS, Steffen ; BORMANN, Richard ; FISCHER, Jan ; VERL, Alexander: Evaluation of 3D feature descriptors for classification of surface geometries in point clouds. In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. Piscataway, NJ, USA : IEEE, 2012, S. 1644–1650. – DOI 10.1109/IROS.2012.6385552
- [Arbeiter u. a. 2011] ARBEITER, Georg ; HÄGELE, Martin ; VERL, Alexander: Field of view dependent registration of point clouds and incremental extraction of table-tops using time-of-flight cameras : held at the ICRA Workshop: Semantic Perception, Mapping and Exploration (SPME). In: *IEEE / Robotics and Automation Society: 2011 IEEE International Conference on Robotics and Automation ICRA : Better Robots, Better Life. May 9-13, 2011, Shanghai, China*. Piscataway, NJ, USA : IEEE Press, 2011
- [Arun u. a. 1987] ARUN, K. S. ; HUANG, T. S. ; BLOSTEIN, S. D.: Least-Squares Fitting of Two 3-D Point Sets. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-9* (1987), Nr. 5, S. 698–700. – DOI 10.1109/TPAMI.1987.4767965

- [Arya u. Mount 1993] ARYA, S. ; MOUNT, D. M.: Approximate nearest neighbor queries in fixed dimensions. In: *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, 1993, S. 271–280
- [ASUSTek 2012] ASUSTEK: *ASUSTeK Computer Inc. - Multimedia- ASUS Xtion PRO LIVE*. URL http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO_LIVE, Abruf: 5.1.2012
- [Balaguer u. a. 2007] BALAGUER, Carlos ; GIMÉNEZ, Antonio ; JARDÓN, Alberto ; CORREAL, Raúl ; MARTÍNEZ, Santiago ; SABATINI, Angelo ; GENOVESE, Vincenzo: Proprio and Teleoperation of a Robotic System for Disabled Persons' Assistance in Domestic Environments. In: FERRE, Manuel (Hrsg.) ; BUSS, Martin (Hrsg.) ; ARACIL, Rafael (Hrsg.) ; MELCHIORRI, Claudio (Hrsg.) ; BALAGUER, Carlos (Hrsg.): *Advances in Telerobotics*. Berlin, Heidelberg : Springer, 2007 (Springer Tracts in Advanced Robotics 31), S. 415–427. – DOI 10.1007/978-3-540-71364-7_25
- [Barber u. a. 1996] BARBER, C. B. ; DOBKIN, David P. ; HUHDANPAA, Hannu: The quickhull algorithm for convex hulls. In: *ACM Trans. Math. Softw.* 22 (1996), S. 469–483. – DOI 10.1145/235815.235821
- [Bauernschmitt u. a. 2009] BAUERNSCHMITT, R. ; BRAUN, E.U. ; BUSS, M. ; FROHLICH, F. ; HIRCHE, S. ; HIRZINGER, G. ; KAMMERL, J. ; KNOLL, A. ; KONIETSCHKE, R. ; KUBLER, B. u. a.: On the role of multimodal communication in telesurgery systems. In: *2009 IEEE International Workshop on Multimedia Signal Processing (MMSP 2009)*. Piscataway, NJ, USA : IEEE Press, 2009, S. 1–6
- [Bell 2003] BELL, J. F.: Mars Exploration Rover Athena Panoramic Camera (Pancam) investigation. In: *Journal of Geophysical Research* 108 (2003), Nr. E12. – DOI 10.1029/2003JE002070
- [Bernardini u. Bajaj 1997] BERNARDINI, F. ; BAJAJ, C.: Sampling and reconstructing manifolds using alpha-shapes. In: *Proc. of the Ninth Canadian Conference on Computational Geometry*. Ottawa : Carleton University Press, 1997, S. 193–198
- [Besl u. McKay 1992] BESL, P.J. ; MCKAY, H.D.: A method for registration of 3-D shapes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992), Nr. 2, S. 239–256. – DOI 10.1109/34.121791
- [Biswas u. Veloso 2012] BISWAS, J. ; VELOSO, M.: Planar polygon extraction and merging from depth images. In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. Piscataway, NJ, USA : IEEE Press, 2012, S. 3859–3864. – DOI 10.1109/IROS.2012.6385841
- [Bo You u. a. 2010] BO YOU ; YU ZOU ; WANZHE XIAO ; JINYU WANG: Telerobot control system based on dual-virtual model and virtual force. In: *Strategic Technology (IFOST), 2010 International Forum on*. Piscataway, NJ, USA : IEEE Press, 2010, S. 246–250. – DOI 10.1109/IFOST.2010.5667968

- [Borrmann u. a. 2011] BORRMANN, Dorit ; ELSEBERG, Jan ; LINGEMANN, Kai ; NÜCHTER, Andreas: The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design. In: *3D Research* 2 (2011), Nr. 2, S. 1–13. – DOI 10.1007/3DRes.02(2011)3
- [Bronstein u. Semendjajew 2001] BRONSTEIN, Ilja N. ; SEMENDJAJEW, Konstantin A.: *Taschenbuch der Mathematik*. Deutsch, 2001. – ISBN 978–3817120055
- [Bultheel u. Cools 2008] BULTHEEL, A. ; COOLS, R.: Extraction of Closed Feature Lines from Point Clouds Based on Graph Theory. In: *Nieuw Archief voor Wiskunde* 5 (2008), Nr. 1, S. 24–28
- [Buss u. a. 2009] BUSS, M. ; PEER, A. ; SCHAUSS, T. ; STEFANOV, N. ; UNTERHINNINGHOFFEN, U. ; BEHRENDT, S. ; LEUPOLD, J. ; DURKOVIC, M. ; SARKIS, M.: Development of a Multi-modal Multi-user Telepresence and Teleaction System. In: *The International Journal of Robotics Research* 29 (2009), Nr. 10, S. 1298–1316. – DOI 10.1177/0278364909351756
- [Canny 1986] CANNY, John: A Computational Approach to Edge Detection. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8 (1986), Nr. 6, S. 679–698. – DOI 10.1109/TPAMI.1986.4767851
- [Captaindistance 2008] CAPTAINDISTANCE: *File:TOF Kamera.jpg - Wikipedia, the free encyclopedia*. URL http://en.wikipedia.org/wiki/File:TOF_Kamera.jpg, Abruf: 18.9.2012
- [Carff u. a. 2009] CARFF, J. ; JOHNSON, M. ; EL-SHEIKH, E.M. ; PRATT, J.E.: Human-robot team navigation in visually complex environments. In: *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. Piscataway, NJ, USA : IEEE Press, 2009, S. 3043–3050
- [Chen u. a. 2003] CHEN, Z. ; BODENHEIMER, B. ; BARNES, J. F.: Robust transmission of 3D geometry over lossy networks. In: *Proceedings of the eighth international conference on 3D Web technology*. New York, NY, USA : ACM, 2003, S. 161–172. – DOI 10.1145/636593.636618
- [Cherubini u. a. 2007] CHERUBINI, A. ; ORIOLO, G. ; MACRI, F. ; ALOISE, F. ; BABILONI, F. ; CINCOTTI, F. ; MATTIA, D.: Development of a multimode navigation system for an assistive robotics project. In: *Robotics and Automation, 2007 IEEE International Conference on*. Piscataway, NJ, USA : IEEE Press, 2007, S. 2336–2342. – DOI 10.1109/ROBOT.2007.363668
- [Chiabrando 2010] CHIABRANDO, F.: SR-4000 TOF Camera: Further experimental tests and first applications to metric surveys. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII* (2010), Nr. 5, S. 149–154
- [Chipalkatty u. a. 2011] CHIPALKATTY, R. ; DAEPP, H. ; EGERSTEDT, M. ; BOOK, W.: Human-in-the-loop: MPC for shared control of a quadruped rescue robot. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. Piscataway, NJ, USA : IEEE Press, 2011, S. 4556–4561

- [Cole u. Newman 2006] COLE, D.M. ; NEWMAN, P.M.: Using laser range data for 3D SLAM in outdoor environments. In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on.* Piscataway, NJ, USA : IEEE Press, 2006, S. 1556–1563. – DOI 10.1109/ROBOT.2006.1641929
- [Coltin u. a. 2012] COLTIN, Brian ; BISWAS, Joydeep ; POMERLEAU, Dean ; VELOSO, Manuela: Effective Semi-autonomous Telepresence. In: RÖFER, Thomas (Hrsg.) ; MAYER, N.Michael (Hrsg.) ; SAVAGE, Jesus (Hrsg.) ; SARANLI, Uluc (Hrsg.): *RoboCup 2011: Robot Soccer World Cup XV* Bd. 7416. Berlin, Heidelberg : Springer, 2012, S. 365–376. – DOI 10.1007/978-3-642-32060-6_31
- [Connette u. a. 2012] CONNETTE, Christian ; ARBEITER, Georg ; MESSMER, Felix ; HÄGELE, Martin ; VERL, Alexander ; NOTHEIS, Simon ; MENDE, Michael ; HEIN, Björn ; WÖRN, Heinz: Efficient Monitoring of Process Plants by Telepresence and Attention Guidance. In: *ROBOTIK 2012 - 7th Conference on Robotics.* Berlin, Offenbach : VDE Verlag, 2012, S. 269–272
- [Connor u. Kumar 2010] CONNOR, Michael ; KUMAR, Piyush: Fast Construction of k-Nearest Neighbor Graphs for Point Clouds. In: *IEEE Transactions on Visualization and Computer Graphics* 16 (2010), Nr. 4, S. 599–608. – DOI 10.1109/TVCG.2010.9
- [DLR - Institute of Robotics and Mechatronics 2012] DLR - INSTITUTE OF ROBOTICS AND MECHATRONICS: *Rollin' Justin.* URL <http://www.dlr.de/rm/en/desktopdefault.aspx/tabid-5471>, Abruf: 20.8.2012
- [Droeschel u. a. 2010] DROESCHEL, D. ; HOLZ, D. ; BEHNKE, S.: Probabilistic Phase Unwrapping for Time-of-Flight Cameras. In: *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK).* Berlin, Offenbach : VDE Verlag, 2010, S. 318–324
- [Dupuis u. a. 2005] DUPUIS, E. ; ALLARD, P. ; BAKAMBU, J. ; LAMARCHE, T. ; ZHU, W. ; REKLEITIS, I.: Towards autonomous long-range navigation. In: *Proceedings 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space.* Noordwijk : ESA Publications, 2005
- [Elseberg u. a. 2012] ELSEBERG, Jan ; MAGNENAT, Stephane ; SIEGWART, Roland ; NÜCHTER, Andreas: Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration. In: *Journal of Software Engineering for Robotics (JOSE)* 3 (2012), Nr. 1, S. 2–12
- [Engelhard u. a. 2011] ENGELHARD, Nikolas ; ENDRES, Felix ; HESS, Jürgen ; STURM, Jürgen ; BURGARD, Wolfram: Real-time 3D visual SLAM with a hand-held RGB-D camera. In: *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, 2011*
- [Evan-Amos 2011] EVAN-AMOS: *File:Xbox-360-Kinect-Standalone.png - Wikipedia, the free encyclopedia.* URL <http://en.wikipedia.org/wiki/File:Xbox-360-Kinect-Standalone.png>, Abruf: 18.9.2012

- [Fairfield u. a. 2007] FAIRFIELD, Nathaniel ; KANTOR, George ; WETTERGREEN, David: Real-Time SLAM with Octree Evidence Grids for Exploration in Underwater Tunnels. In: *Journal of Field Robotics* 24 (2007), Nr. 1-2, S. 03–21. – DOI 10.1002/rob.20165
- [FareEastGizmos 2011] FAREEASTGIZMOS: *Upgraded Quince robots ready for gathering data inside the Fukushima No.1 nuclear power plant*. URL <http://www.fareastgizmos.com/robotic>, Abruf: 3.1.2013
- [Feisst u. Christ 2004] FEISST, M. ; CHRIST, A.: Dynamically optimised 3D (virtual reality) data transmission for mobile devices. In: *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*. Piscataway, NJ, USA : IEEE Press, 2004, S. 270–274. – DOI 10.1109/TDPVT.2004.1335205
- [Foix u. a. 2011] FOIX, S. ; ALENYA, G. ; TORRAS, C.: Lock-in Time-of-Flight (ToF) Cameras: A Survey. In: *Sensors Journal, IEEE* 11 (2011), Nr. 9, S. 1917 –1926. – DOI 10.1109/JSEN.2010.2101060
- [Forsyth u. Ponce 2003] FORSYTH, David A. ; PONCE, Jean: *Computer vision : a modern approach*. Upper Saddle River, NJ, USA : Prentice Hall, 2003. – ISBN 978–0130851987
- [Frank u. a. 2009] FRANK, M. ; PLAUE, M. ; RAPP, H. ; KÖTHE, U. ; JÄHNE, B. ; HAMPRECHT, F. A.: Theoretical and experimental error analysis of continuous-wave time-of-flight range cameras. In: *Optical Engineering* 48 (2009), S. 013602
- [Fraunhofer IPA 2012] FRAUNHOFER IPA: *Care-O-bot*. URL <http://www.care-o-bot.de>, Abruf: 18.9.2012
- [Freedman u. a.] Schutzrecht WO/2008/120217 (2008)
- [Fuchs u. Hirzinger 2008] FUCHS, S. ; HIRZINGER, G.: Extrinsic and depth calibration of ToF-cameras. In: *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*. Piscataway, NJ, USA : IEEE Press, 2008, S. 1–6
- [Fuchs u. May 2008] FUCHS, Stefan ; MAY, Stefan: Calibration and registration for precise surface reconstruction with Time-Of-Flight cameras. In: *International Journal of Intelligent Systems Technologies and Applications* 5 (2008), Nr. 3/4, S. 274–284. – DOI 10.1504/IJIS-TA.2008.021290
- [Fuchs 2011] FUCHS, Steffen: *Evaluation of Algorithms for 3D Feature Estimation in Point Clouds*, Fachhochschule Würzburg-Schweinfurt, Diplomarbeit, 2011
- [Georgiev u. a. 2011] GEORGIEV, Kristiyan ; CREED, Ross T. ; LAKAEMPER, Rolf: Fast plane extraction in 3D range data based on line segments. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. Piscataway, NJ, USA : IEEE Press, 2011, S. 3808–3815. – DOI 10.1109/IROS.2011.6094916
- [Gonzalez u. Woods 2002] GONZALEZ, Rafael C. ; WOODS, Richard E.: *Digital image processing*. Upper Saddle River, NJ, USA : Prentice Hall, 2002. – ISBN 978–0201180756

- [Goodfellow u. a. 2010] GOODFELLOW, I.J. ; KOENIG, N. ; MUJA, M. ; PANTOFARU, C. ; SOROKIN, A. ; TAKAYAMA, L.: Help me help you: interfaces for personal robots. In: *Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction*. Piscataway, NJ, USA : IEEE Press, 2010, S. 187–188
- [Graf u. a. 2009] GRAF, Birgit ; PARLITZ, Christopher ; HÄGELE, Martin: Robotic Home Assistant Care-O-bot[®] 3 Product Vision and Innovation Platform. In: *HCI (2)*, 2009, S. 312–320
- [Grisetti u. a. 2010] GRISETTI, G. ; KÜMMERLE, R. ; STACHNISS, C. ; BURGARD, W.: A Tutorial on Graph-Based SLAM. In: *Intelligent Transportation Systems Magazine, IEEE 2* (2010), Nr. 4, S. 31–43
- [Groenda u. a. 2005] GROENDA, Henning ; NOWAK, Fabian ; RÖSSLER, Patrick ; HANEBECK, Uwe D.: Telepresence Techniques for Controlling Avatar Motion in First Person Games. In: MAYBURY, Mark (Hrsg.) ; STOCK, Oliviero (Hrsg.) ; WAHLSTER, Wolfgang (Hrsg.): *Intelligent Technologies for Interactive Entertainment* Bd. 3814. Berlin, Heidelberg : Springer, 2005, S. 44–53. – DOI 10.1007/11590323_5
- [Gudmundsson u. a. 2007] GUDMUNDSSON, S. A. ; AANAES, H. ; LARSEN, R.: Environmental effects on measurement uncertainties of time-of-flight cameras. In: *International Symposium on Signals, Circuits and Systems, 2007. ISSCS 2007*. Piscataway, NJ, USA : IEEE Press, 2007
- [Gumhold u. a. 2005] GUMHOLD, S. ; KAMI, Z. ; ISENBURG, M. ; SEIDEL, H. P.: Predictive point-cloud compression. In: *ACM SIGGRAPH 2005 Sketches*. New York, NY, USA : ACM, 2005, S. 137
- [Hampp 2012] HAMPP, Joshua: *Evaluierung von 3D Registrierungsverfahren in Indoor-Umgebungen*. Isny, nta Isny, Diplomarbeit, Februar 2012
- [Hasunuma u. a. 1999] HASUNUMA, H. ; KAGAYA, H. ; TAKATORI, M. ; FUJIMORI, J. ; MIFUNE, F. ; SHIKODA, S. ; KOBAYASI, M. ; ITOKO, T. ; TACHI, S.: Development of teleoperation master system with a kinesthetic sensation of presence. In: *Proc. of the International Conference on Artificial reality and Telexistence, Waseda University*. Piscataway, NJ, USA : IEEE Press, 1999
- [Henry u. a. 2012] HENRY, P. ; KRAININ, M. ; HERBST, E. ; REN, X. ; FOX, D.: RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. In: *The International Journal of Robotics Research* 31 (2012), Nr. 5, S. 647–663. – DOI 10.1177/0278364911434148
- [Herbert u. a. 1989] HERBERT, M. ; CAILLAS, C. ; KROTKOV, E. ; KWEON, I.S. ; KANADE, T.: Terrain mapping for a roving planetary explorer. In: *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*. Piscataway, NJ, USA : IEEE Press, 1989, S. 997–1002. – DOI 10.1109/ROBOT.1989.100111
- [Holwerda 1998] HOLWERDA, Klaas: *Complete Boolean Description*. URL <http://boolean.klaasholwerda.nl/algdoc/top.html>, Abruf: 15.9.2011

- [Holz u. a. 2012] HOLZ, Dirk ; HOLZER, Stefan ; RUSU, Radu B. ; BEHNKE, Sven: Real-time plane segmentation using RGB-D cameras. In: RÖFER, Thomas (Hrsg.) ; MAYER, N. Michael (Hrsg.) ; SAVAGE, Jesus (Hrsg.) ; SARANLI, Uluc (Hrsg.): *RoboCup 2011: Robot Soccer World Cup XV* Bd. 7416. Berlin, Heidelberg : Springer, 2012, S. 306–317
- [Holz u. a. 2011] HOLZ, Dirk ; SCHNABEL, Ruwen ; DROESCHEL, David ; STÜCKLER, Jörg ; BEHNKE, Sven: Towards Semantic Scene Analysis with Time-of-Flight Cameras. In: SOLAR, Javier Ruiz-del (Hrsg.) ; CHOWN, Eric (Hrsg.) ; PLÖGER, Paul G. (Hrsg.): *RoboCup 2010: Robot Soccer World Cup XIV* Bd. 6556. Berlin, Heidelberg : Springer, 2011, S. 121–132. – DOI 10.1007/978-3-642-20217-9_11
- [Hoppe u. a. 1992] HOPPE, Hugues ; DEROSE, Tony ; DUCHAMP, Tom ; McDONALD, John ; STUETZLE, Werner: Surface reconstruction from unorganized points. In: *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM, 1992, S. 71–78. – DOI 10.1145/133994.134011
- [Hsiao u. a. 2005] HSIAO, Y.T. ; CHUANG, C.L. ; JIANG, J.A. ; CHIEN, C.C.: A contour based image segmentation algorithm using morphological edge detection. In: *Systems, Man and Cybernetics, 2005 IEEE International Conference on*. Piscataway, NJ, USA : IEEE Press, 2005, S. 2962–2967
- [Hägele u. a. 2011] HÄGELE, Martin (Hrsg.) ; BENGEL, Matthias (Hrsg.) ; BLÜMLEIN, Nikolaus (Hrsg.) ; CONNETTE, Christian P. (Hrsg.) ; FISCHER, Jan (Hrsg.) ; GRAF, Birgit (Hrsg.) ; HÖPF, Michael (Hrsg.) ; JACOBS, Theo (Hrsg.) ; KLEINE, Oliver (Hrsg.) ; PFEIFFER, Kai (Hrsg.) ; WÖLTJE, Kay (Hrsg.) ; ROST, Arne (Hrsg.): *Wirtschaftlichkeitsanalysen neuartiger Servicerobotik - Anwendungen und ihre Bedeutung für die Robotik-Entwicklung : EFFIROB-Studie. Eine Analyse der Fraunhofer-Institute IPA und ISI im Auftrag des BMBF (Kennzeichen 01M09001) zwischen dem 1. Dezember 2009 und dem 30. November 2010*. 2011. – 371 S.
- [ifm electronic 2012] IFM ELECTRONIC: *Automation - Sensorik, Kommunikations- und Steuerungssysteme - ifm electronic*. URL <http://www.ifm.com/ifmde/web/home.htm>, Abruf: 3.1.2012
- [Iocchi u. Pellegrini 2007] IOCCHI, L. ; PELLEGRINI, S.: Building 3d maps with semantic elements integrating 2d laser, stereo vision and imu on a mobile robot. In: *Proceedings of the 2nd ISPRS International Workshop 3D-ARCH 2007, 3D Virtual Reconstruction and Visualization of Complex Architectures*. ETH Zurich, Switzerland : ISPRS, 2007
- [iRobot 2012] IROBOT: *iRobot: Robots for Defense and Public Safety*. URL <http://www.irobot.com/en/us/robots/defense.aspx>, Abruf: 27.7.2012
- [Izadi u. a. 2011] IZADI, S. ; KIM, D. ; HILLIGES, O. ; MOLYNEAUX, D. ; NEWCOMBE, R. ; KOHLI, P. ; SHOTTON, J. ; HODGES, S. ; FREEMAN, D. ; DAVISON, A. ; OTHERS: KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In: *Proceedings of the 24th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM, 2011, S. 559–568. – DOI 10.1145/2047196.2047270

- [Jain u. Kemp 2010] JAIN, Advait ; KEMP, Charles C.: EL-E: an assistive mobile manipulator that autonomously fetches objects from flat surfaces. In: *Autonomous Robots* 28 (2010), Nr. 1, S. 45–64
- [Jiang u. a. 2006] JIANG, Z. ; XIE, Z. ; WANG, B. ; WANG, J. ; HUANG, J. ; LIU, H.: A teleprogramming method for internet-based teleoperation. In: *Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on*. Piscataway, NJ, USA : IEEE Press, 2006, S. 422–426
- [Johnson u. Kang 1997] JOHNSON, A. E. ; KANG, S. B.: Registration and integration of textured 3-D data. In: *3-D Digital Imaging and Modeling, 1997. Proceedings., International Conference on Recent Advances in*. Piscataway, NJ, USA : IEEE Press, 1997, S. 234–241. – DOI 10.1109/IM.1997.603871
- [Johnson u. Hebert 1999] JOHNSON, A.E. ; HEBERT, M.: Using spin images for efficient object recognition in cluttered 3D scenes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (1999), Nr. 5, S. 433–449. – DOI 10.1109/34.765655
- [Johnson-Roberson u. a. 2011] JOHNSON-ROBERSON, M. ; BOHG, J. ; SKANTZE, G. ; GUSTAFSON, J. ; CARLSON, R. ; RASOLZADEH, B. ; KRAGIC, D.: Enhanced visual scene understanding through human-robot dialog. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. Piscataway, NJ, USA : IEEE Press, 2011, S. 3342–3348
- [Kaess u. a. 2003] KAESS, Michael ; ARKIN, Ronald C. ; ROSSIGNAC, Jarek: Compact Encoding of Robot-Generated 3D Maps for Efficient Wireless Transmission. In: *IEEE Intl. Conf. on Advanced Robotics, ICAR*. Piscataway, NJ, USA : IEEE Press, 2003, S. 324–331
- [Kahlmann u. Ingensand 2008] KAHLMANN, Timo ; INGENSAND, Hilmar: Calibration and development for increased accuracy of 3D range imaging cameras. In: *Journal of Applied Geodesy* 2 (2008), Nr. 1, S. 1–11. – DOI 10.1515/JAG.2008.001
- [Kheddar u. a. 2007] KHEDDAR, Abderrahmane ; NEO, Ee-Sian ; TADAKUMA, Riichiro ; YOKOI, Kazuhito: Enhanced Teleoperation Through Virtual Reality Techniques. In: FERRE, Manuel (Hrsg.) ; BUSS, Martin (Hrsg.) ; ARACIL, Rafael (Hrsg.) ; MELCHIORRI, Claudio (Hrsg.) ; BALAGUER, Carlos (Hrsg.): *Advances in Telerobotics*. Springer Berlin / Heidelberg, 2007 (Springer Tracts in Advanced Robotics 31), S. 139–159. – DOI 10.1007/978-3-540-71364-7_10
- [Khoshelham 2011] KHOSHELHAM, K.: Accuracy analysis of kinect depth data. In: *ISPRS workshop laser scanning 2011, Calgary, Canada, 29-31 August 2011*, 2011, S. 133–138
- [Koster u. Spann 2000] KOSTER, K. ; SPANN, M.: MIR: an approach to robust clustering-application to range image segmentation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000), Nr. 5, S. 430–444. – DOI 10.1109/34.857001
- [Kunz u. a. 2010] KUNZ, T ; REISER, U ; STILMAN, M ; VERL, A: Real-time path planning for a robot arm in changing environments. In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. Piscataway, NJ, USA : IEEE Press, 2010, S. 5906–5911. – DOI 10.1109/IROS.2010.5653275

- [Lange 2000] LANGE, R.: *3D time-of-flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology*, Siegen, Diss., 2000
- [Lazebnik u. a. 2005] LAZEBNIK, Svetlana ; SCHMID, Cordelia ; PONCE, Jean: A sparse texture representation using local affine regions. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 27 (2005), Nr. 8, S. 1265–1278
- [Lee u. Schachter 1980] LEE, D T. ; SCHACHTER, B J.: Two algorithms for constructing a Delaunay triangulation. In: *International Journal of Computer and Information Sciences* 9 (1980), Nr. 3, S. 219–242
- [Leonov 1998] LEONOV, Michael: *PolyBoolean . Comparison...* URL <http://www.complex-a5.ru/polyboolean/comp.html>, Abruf: 15.9.2011
- [Li u. a. 2003] LI, Jie ; GAO, Shuming ; ZHOU, Xun: Direct incremental transmission of boundary representation. In: *Proceedings of the eighth ACM symposium on Solid modeling and applications*. New York, NY, USA : ACM, 2003, S. 298–303. – DOI 10.1145/781606.781655
- [Lowe 2004] LOWE, David G.: Distinctive Image Features from Scale-Invariant Keypoints. In: *International Journal of Computer Vision* 60 (2004), Nr. 2, S. 91–110. – DOI 10.1023/B:VISI.0000029664.99615.94
- [Luck u. a. 2000] LUCK, J. ; LITTLE, C. ; HOFF, W.: Registration of range data using a hybrid simulated annealing and iterative closest point algorithm. In: *IEEE International Conference on Robotics and Automation*. Piscataway, NJ, USA : IEEE Press, 2000, S. 3739–3744
- [Ma 2004] MA, Yi: *An invitation to 3-D vision: from images to geometric models*. New York : Springer, 2004 (Texts in applied mathematics). – ISBN 0387008934
- [Magnusson u. a. 2007] MAGNUSSON, Martin ; LILIENTHAL, Achim ; DUCKETT, Tom: Scan registration for autonomous mining vehicles using 3D-NDT. In: *Journal of Field Robotics* 24 (2007), Nr. 10, S. 803–827. – DOI 10.1002/rob.20204
- [Malysz u. Sirouspour 2011] MALYSZ, P. ; SIROUSPOUR, S.: A task-space weighting matrix approach to semi-autonomous teleoperation control. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. Piscataway, NJ, USA : IEEE Press, 2011, S. 645–652
- [Manning u. a. 2008] MANNING, Christopher D. ; RAGHAVAN, Prabhakar ; SCHÜTZE, Hinrich: *Introduction to information retrieval*. New York : Cambridge University Press, 2008. – ISBN 9780521865715
- [Marsiske 2011] MARSISKE, H.-A.: Roboterspezialisten nutzen Erfahrungen aus Fukushima. In: *VDI Nachrichten* (2011), September, Nr. 38
- [Marton u. a. 2010] MARTON, Z ; PANGERCIC, D ; BLODOW, N ; KLEINEHELLEFORT, J ; BEETZ, M: General 3D modelling of novel objects from a single view. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Piscataway, NJ, USA : IEEE Press, 2010, S. 3700–3705. – DOI 10.1109/IROS.2010.5650434

- [Marton u. a. 2009] MARTON, Z. C. ; RUSU, R. B. ; BEETZ, M.: On fast surface reconstruction methods for large and noisy point clouds. In: *Proceedings of the International Conference on Robotics and Automation (ICRA), Kobe, Japan*. Piscataway, NJ, USA : IEEE Press, 2009, S. 3218–3223. – DOI 10.1109/ROBOT.2009.5152628
- [Mast u. a. 2012] MAST, Marcus ; BURMESTER, Michael ; KRÜGER, Katja ; FATIKOW, Sascha ; ARBEITER, Georg ; GRAF, Birgit ; KRONREIF, Gernot ; PIGINI, Lucia ; FACAL, David ; QIU, Renxi: User-Centered Design of a Dynamic-Autonomy Remote Interaction Concept for Manipulation-Capable Robots to Assist Elderly People in the Home. In: *Journal of Human-Robot Interaction* 1 (2012), Nr. 1, S. 96–118. – DOI 10.5898/JHRI.1.1.Mast
- [May u. a. 2008] MAY, S. ; DROESCHEL, D. ; HOLZ, D. ; WIESEN, C. ; FUCHS, S.: 3D Pose Estimation and Mapping with Time-of-Flight Cameras. In: *International Conference on Intelligent Robots and Systems (IROS), 3D Mapping workshop, Nice, France, 2008*
- [May u. a. 2009] MAY, S. ; FUCHS, S. ; DROESCHEL, D. ; HOLZ, D. ; NÜCHTER, A.: Robust 3D-mapping with time-of-flight cameras. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ, USA : IEEE Press, 2009
- [Meagher 1982] MEAGHER, D: Geometric modeling using octree encoding. In: *Computer Graphics and Image Processing* 19 (1982), Nr. 2, S. 129–147. – DOI 10.1016/0146–664X(82)90104–6
- [Mesa Imaging AG 2012] MESA IMAGING AG: *Mesa Imaging AG - SwissRanger SR4000 - miniature 3D time-of-flight range camera*. URL <http://www.mesa-imaging.ch/prodview4k.php?cat=3D%20Camera>, Abruf: 3.1.2012
- [Microsoft 2012] MICROSOFT: *Kinect für Xbox 360 - Xbox.com*. URL <http://www.xbox.com/de-DE/kinect>, Abruf: 5.1.2012
- [Moravec 1988] MORAVEC, H. P.: Sensor fusion in certainty grids for mobile robots. In: *AI magazine* 9 (1988), Nr. 2, S. 61–74
- [Mozos 2010] MOZOS, Oscar: *Semantic labeling of places with mobile robots*. Berlin, Heidelberg : Springer, 2010. – ISBN 978–3642112096
- [Nagatani u. a. 2009] NAGATANI, K. ; OKADA, Y. ; TOKUNAGA, N. ; YOSHIDA, K. ; KIRIBAYASHI, S. ; OHNO, K. ; TAKEUCHI, E. ; TADOKORO, S. ; AKIYAMA, H. ; NODA, I. ; OTHERS: Multi-robot exploration for search and rescue missions: A report of map building in RoboCupRescue 2009. In: *Safety, Security & Rescue Robotics (SSRR), 2009 IEEE International Workshop on*. Piscataway, NJ, USA : IEEE Press, 2009, S. 1–6
- [Nagatani u. a. 2010] NAGATANI, K. ; TOKUNAGA, N. ; OKADA, Y. ; YOSHIDA, K. ; HADA, Y. ; YOSHIDA, T. ; KOYANAGI, E.: Teleoperation of all-terrain robot using continuous acquisition of three-dimensional environment under time-delayed narrow bandwidth communication. In: *Safety, Security & Rescue Robotics (SSRR), 2009 IEEE International Workshop on*. Piscataway, NJ, USA : IEEE Press, 2010, S. 1–6

- [Nagatani u. a. 2011] NAGATANI, Keiji ; KIRIBAYASHI, Seiga ; OKADA, Yoshito ; TADOKORO, Satoshi ; NISHIMURA, Takeshi ; YOSHIDA, Tomoaki ; KOYANAGI, Eiji ; HADA, Yasushi: Redesign of rescue mobile robot Quince. In: *Proceedings of the 2011 IEEE Int'l Workshop on Safety, Security and Rescue Robotics*. Piscataway, NJ, USA : IEEE Press, 2011, S. 13–18
- [Nagatani u. a. 2008] NAGATANI, Keiji ; TOKUNAGA, Naoki ; OKADA, Yoshito ; YOSHIDA, Kazuya: Continuous Acquisition of Three-Dimensional Environment Information for Tracked Vehicles on Uneven Terrain. In: *Safety, Security and Rescue Robotics, 2008. SSRR 2008. IEEE International Workshop on*. Piscataway, NJ, USA : IEEE Press, 2008, S. 25–30. – DOI 10.1109/SSRR.2008.4745872
- [NASA 2011] NASA: *File:Mars Science Laboratory Curiosity rover.jpg - Wikipedia, the free encyclopedia*. URL http://en.wikipedia.org/wiki/File:Mars_Science_Laboratory_Curiosity_rover.jpg, Abruf: 18.9.2012
- [NASA - Mars Science Laboratory 2012] NASA - MARS SCIENCE LABORATORY: *Curiosity: NASA's Next Mars Rover*. URL http://www.nasa.gov/mission_pages/msl/index.html, Abruf: 20.8.2012
- [Newcombe u. a. 2011] NEWCOMBE, R.A. ; IZADI, S. ; HILLIGES, O. ; MOLYNEAUX, D. ; KIM, D. ; DAVISON, A.J. ; KOHLI, P. ; SHOTTON, J. ; HODGES, S. ; FITZGIBBON, A.: KinectFusion: Real-time dense surface mapping and tracking. In: *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*. Piscataway, NJ, USA : IEEE Press, 2011, S. 127–136
- [Notheis u. a. 2010] NOTHEIS, S ; MILIGHETTI, G ; HEIN, B ; WÖRN, H ; BEYERER, J: Skill-based telemanipulation by means of intelligent robots. In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. Piscataway, NJ, USA : IEEE Press, Oktober 2010, S. 5258–5263. – DOI 10.1109/IROS.2010.5650731
- [Nüchter u. Hertzberg 2008] NÜCHTER, A. ; HERTZBERG, J.: Towards semantic maps for mobile robots. In: *Robotics and Autonomous Systems* 56 (2008), Nr. 11, S. 915–926
- [Nüchter 2009] NÜCHTER, Andreas: *Springer tracts in advanced robotics*. Bd. 52: *3D robotic mapping: The simultaneous localization and mapping problem with six degrees of freedom*. Berlin, Heidelberg : Springer, 2009. – ISBN 9783540898832
- [Nüchter u. a. 2007a] NÜCHTER, Andreas ; LINGEMANN, Kai ; HERTZBERG, Joachim: Cached k-d tree search for ICP algorithms. In: *3-D Digital Imaging and Modeling, 2007. 3DIM '07. Sixth International Conference on*. Piscataway, NJ, USA : IEEE Press, 2007, S. 419–426. – DOI 10.1109/3DIM.2007.15
- [Nüchter u. a. 2007b] NÜCHTER, Andreas ; LINGEMANN, Kai ; HERTZBERG, Joachim ; SURMANN, Hartmut: 6D SLAM—3D mapping outdoor environments. In: *Journal of Field Robotics* 24 (2007), Nr. 8-9, S. 699–722. – DOI 10.1002/rob.20209
- [Oggier u. a. 2004] OGGIER, Thierry ; LEHMANN, Michael ; KAUFMANN, Rolf ; SCHWEIZER, Matthias ; RICHTER, Michael ; METZLER, Peter ; LANG, Graham ; LUSTENBERGER, Felix ;

- BLANC, Nicolas: An all-solid-state optical range camera for 3D real-time imaging with sub-centimeter depth resolution (SwissRanger). In: *Proc. SPIE 5249* (2004), S. 534–545. – DOI 10.1117/12.513307
- [Ohroku u. Nonami 2009] OHROKU, H. ; NONAMI, K.: Omni-directional Vision and 3D Animation Based Teleoperation of Hydraulically Actuated Hexapod Robot COMET-IV. In: *ICGST International Journal on Automation, Robotics and Autonomous Systems* 9 (2009), Nr. 1, S. 17–24
- [Olson u. Delen 2008] OLSON, David L. ; DELEN, Dursun: *Advanced data mining techniques*. Berlin, Heidelberg : Springer, 2008. – ISBN 978–3540769163
- [Pearson 1901] PEARSON, Karl: On lines and planes of closest fit to systems of points in space. In: *Philosophical Magazine* 2 (1901), Nr. 6, S. 559–572
- [Pfister u. a. 2000] PFISTER, H. ; ZWICKER, M. ; VAN BAAR, J. ; GROSS, M.: Surfels: Surface elements as rendering primitives. In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM, 2000, S. 335–342
- [Pitzer u. a. 2011] PITZER, B. ; STYER, M. ; BERSCH, C. ; DUHADWAY, C. ; BECKER, J.: Towards Perceptual Shared Autonomy for Robotic Mobile Manipulation. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. Piscataway, NJ, USA : IEEE Press, 2011, S. 6245–6251
- [PMDTec 2012] PMDTEC: *PMDTec: Leading chip supplier for 3D CMOS imaging*. URL <http://www.pmdtec.com/>, Abruf: 3.1.2012
- [Poppinga u. a. 2008] POPPINGA, Jann ; VASKEVICIUS, Narunas ; BIRK, Andreas ; PATHAK, Kaustubh: Fast plane detection and polygonalization in noisy 3D range images. In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. Piscataway, NJ, USA : IEEE Press, 2008, S. 3378–3383. – DOI 10.1109/IROS.2008.4650729
- [Pronobis 2011] PRONOBIS, A.: *Semantic Mapping with Mobile Robots*, PhD thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, Diss., 2011
- [Qiu u. a. 2012] QIU, Renxi ; NOYVIRT, Alexandre ; JI, Ze ; SOROKA, Anthony ; LI, Dayou ; LIU, Beisheng ; ARBEITER, Georg ; WEISSHARDT, Florian ; XU, Shuo: Integration of Symbolic Task Planning into Operations within an Unstructured Environment. In: *International Journal of Intelligent Mechatronics and Robotics* 2 (2012), Nr. 3, S. 38–57. – DOI 10.4018/ijmr.2012070104
- [Rabbani u. a. 2006] RABBANI, T. ; HEUVEL, F. van d. ; VOSSELMANN, G.: Segmentation of point clouds using smoothness constraint. In: *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 36 (2006), Nr. 5, S. 248–253
- [Rusinkiewicz u. Levoy 2001] RUSINKIEWICZ, S. ; LEVOY, M.: Efficient variants of the ICP algorithm. In: *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. Piscataway, NJ, USA : IEEE Press, 2001, S. 145–152

- [Rusu u. a. 2007] RUSU, R. B. ; BLOWOW, N. ; MARTON, Z. ; SOOS, A. ; BEETZ, M.: Towards 3D object maps for autonomous household robots. In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. Piscataway, NJ, USA : IEEE Press, 2007, S. 3191–3198
- [Rusu u. a. 2009a] RUSU, R. B. ; BLOWOW, N. ; MARTON, Z. C. ; BEETZ, M.: Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation in domestic environments. In: *Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*. Piscataway, NJ, USA : IEEE Press, 2009, S. 1–6
- [Rusu u. a. 2008a] RUSU, R. B. ; MARTON, Z. C. ; BLOWOW, N. ; DOLHA, M. ; BEETZ, M.: Towards 3D Point cloud based object maps for household environments. In: *Robotics and Autonomous Systems* 56 (2008), S. 927–941
- [Rusu 2009] RUSU, Radu B.: *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. 2009
- [Rusu u. a. 2009b] RUSU, Radu B. ; BLOWOW, Nico ; BEETZ, Michael: Fast point feature histograms (FPFH) for 3D registration. In: *2009 IEEE International Conference on Robotics and Automation*, IEEE, 2009, S. 1848–1853
- [Rusu u. a. 2008b] RUSU, Radu B. ; BLOWOW, Nico ; MARTON, Zoltan C.: Aligning Point Cloud Views using Persistent Feature Histograms. In: *Proceedings of the 21st IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ, USA : IEEE Press, 2008
- [Rusu u. a. 2008c] RUSU, Radu B. ; MARTON, Zoltan C. ; BLOWOW, Nico ; NICO ; MICHAEL BEETZ, Michael: Persistent Point Feature Histograms for 3D Point Clouds. In: *Proceedings of the 21st IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ, USA : IEEE Press, 2008
- [Rusu u. Cousins 2011] RUSU, R.B. ; COUSINS, S.: 3d is here: Point cloud library (pcl). In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. Piscataway, NJ, USA : IEEE Press, 2011, S. 1–4
- [Sadjadi u. Hall 1980] SADJADI, Firooz A. ; HALL, Ernest L.: Three-Dimensional Moment Invariants. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-2 (1980), Nr. 2, S. 127–136. – DOI 10.1109/TPAMI.1980.4766990
- [Saitoh u. a. 2006] SAITOH, Kensaku ; MACHIDA, Takashi ; KIYOKAWA, Kiyoshi ; TAKEMURA, Haruo: A 2D-3D integrated interface for mobile robot control using omnidirectional images and 3D geometric models. In: *ISMAR '06 Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*. Piscataway, NJ, USA : IEEE Press, 2006, S. 173–176. – DOI 10.1109/ISMAR.2006.297810
- [Samet u. Tamminen 1988] SAMET, H. ; TAMMINEN, M.: Efficient component labeling of images of arbitrary dimension represented by linear bintrees. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10 (1988), Juli, Nr. 4, S. 579–586. – DOI 10.1109/34.3918

- [Sappa u. Devy 2001] SAPPA, A. ; DEVY, M.: Fast range image segmentation by an edge detection strategy. In: *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. Piscataway, NJ, USA : IEEE Press, 2001, S. 292–299
- [Schnabel u. a. 2007] SCHNABEL, R. ; WAHL, R. ; KLEIN, R.: Efficient ransac for point-cloud shape detection. In: *Computer Graphics Forum* 26 (2007), Nr. 2, S. 214–226. – DOI 10.1111/j.1467–8659.2007.01016.x
- [Schraft u. Volz 1996] SCHRAFT, Rolf D. ; VOLZ, Hansjörg: *Serviceroboter : innovative Technik in Dienstleistung und Versorgung*. Berlin, Heidelberg : Springer, 1996. – ISBN 978–3540593591
- [Segal u. a. 2009] SEGAL, A. ; HAEHNEL, D. ; THRUN, S.: Generalized-ICP. In: *Proc. of Robotics: Science and Systems (RSS)*. Cambridge, MA, USA : MIT Press, 2009
- [Sequeira u. a. 1999] SEQUEIRA, V. ; NG, K. ; WOLFART, E. ; GONÇALVES, J.G.M. ; HOGG, D.: Automated reconstruction of 3D models from real environments. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 54 (1999), Nr. 1, S. 1–22. – DOI 10.1016/S0924–2716(98)00026–4
- [Sheh u. a. 2006] SHEH, R. ; KADOUS, M. W. ; SAMMUT, C.: On building 3D maps using a Range camera: Applications to Rescue Robotics / The University of New South Wales. 2006. – Technischer Report
- [Showcase 2012] SHOWCASE, Young T.: Sensabot: A Safe and Cost-Effective Inspection Solution. In: *Journal of Petroleum Technology* (2012), Oktober, S. 32–34
- [Showstack 2012] SHOWSTACK, Randy: After dramatic landing, Mars Curiosity rover will soon search for signs of habitability. In: *Eos, Transactions American Geophysical Union* 93 (2012), August, Nr. 33, S. 318. – DOI 10.1029/2012EO330002
- [Siegwart 2004] SIEGWART, Roland: *Introduction to autonomous mobile robots*. Cambridge, MA, USA : MIT Press, 2004. – ISBN 9780262256995
- [Skelly u. Sclaroff 2007] SKELLY, Luke J. ; SCLAROFF, Stan: Improved feature descriptors for 3D surface matching. In: *Proc. SPIE* 6762 (2007), S. 67620A. – DOI 10.1117/12.753263
- [Song u. Lee 2004] SONG, Youngjae ; LEE, Kunwoo: Incremental transmission of B-Rep models through the network. In: *Computer-Aided Design and Applications* 1 (2004), Nr. 1-4, S. 523–530
- [Spectrum 2011] SPECTRUM: *Fukushima Robot Operator Writes Tell-All Blog - IEEE Spectrum*. URL <http://spectrum.ieee.org/automaton/robotics/industrial-robots/fukushima-robot-operator-diaries>, Abruf: 30.11.2011
- [Srinivasa u. a. 2008] SRINIVASA, Siddhartha ; FERGUSON, David ; VANDEWEGHE, J. M. ; DIANKOV, Rosen ; BERENSON, Dmitry ; HELFRICH, Casey ; STRASDAT, Hauke: The Robotic Busboy: Steps Towards Developing a Mobile Robotic Home Assistant. In: *International Conference on Intelligent Autonomous Systems*. Amsterdam : Ios Press, Juli 2008, S. 651–656

- [Steder u. a. 2011] STEDER, Bastian ; RUSU, Radu B. ; KONOLIGE, Kurt ; BURGARD, Wolfram: Point feature extraction on 3D range scans taking into account object boundaries. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. Piscataway, NJ, USA : IEEE Press, 2011, S. 2601–2608. – DOI 10.1109/ICRA.2011.5980187
- [Strasdat u. a. 2010] STRASDAT, H. ; MONTIEL, J. M. M. ; DAVISON, A. J.: Scale Drift-Aware Large Scale Monocular SLAM. In: *Proceedings of Robotics: Science and Systems*. Cambridge, MA, USA : MIT Press, 2010
- [Straszheim 2011] STRASZHEIM, Troy: *PR2 robot reads the Mythical Man-Month.jpg* - *Wikimedia Commons*. URL http://commons.wikimedia.org/wiki/File:PR2_robot_reads_the_Mythical_Man-Month.jpg, Abruf: 18.9.2012
- [Sturm u. a. 2012] STURM, J. ; ENGELHARD, N. ; ENDRES, F. ; BURGARD, W. ; CREMERS, D.: A Benchmark for the Evaluation of RGB-D SLAM Systems. In: *Proc. of the International Conference on Intelligent Robot Systems (IROS)*. Piscataway, NJ, USA : IEEE Press, 2012, S. 573 – 580. – DOI 10.1109/IROS.2012.6385773
- [Sturm u. a. 2011] STURM, J. ; MAGNENAT, S. ; ENGELHARD, N. ; POMERLEAU, F. ; COLAS, F. ; CREMERS, D. ; SIEGWART, R. ; BURGARD, W.: Towards a benchmark for RGB-D SLAM evaluation. In: *Proc. of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conf.(RSS), Los Angeles, USA, 2011*, S. 3
- [Sucan u. a. 2010] SUCAN, I. A. ; KALAKRISHNAN, M. ; CHITTA, S.: Combining planning techniques for manipulation using realtime perception. In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. Piscataway, NJ, USA : IEEE Press, 2010, S. 2895–2901
- [Sugimoto u. a. 2005] SUGIMOTO, M. ; KAGOTANI, G. ; NII, H. ; SHIROMA, N. ; MATSUNO, F. ; INAMI, M.: Time Follower's Vision: a teleoperation interface with past images. In: *IEEE Computer Graphics and Applications* 25 (2005), Nr. 1, S. 54–63. – DOI 10.1109/MCG.2005.23
- [Sun u. a. 2011] SUN, Jinyuan ; ZHU, Xiaoyan ; ZHANG, Chi ; FANG, Yuguang: RescueMe: Location-Based Secure and Dependable VANETs for Disaster Rescue. In: *IEEE Journal on Selected Areas in Communications* 29 (2011), März, Nr. 3, S. 659–669. – DOI 10.1109/JSAC.2011.110314
- [Suzuki 1985] SUZUKI, S.: Topological structural analysis of digitized binary images by border following. In: *Computer Vision, Graphics, and Image Processing* 30 (1985), Nr. 1, S. 32–46
- [Szeliski 2011] SZELISKI, Richard: *Computer vision algorithms and applications*. London; New York : Springer, 2011. – ISBN 9781848829343
- [Tachi u. a. 2011] TACHI, S. ; WATANABE, K. ; TAKESHITA, K. ; MINAMIZAWA, K. ; YOSHIDA, T. ; SATO, K.: Mutual Telexistence Surrogate System: TELESAR4-telexistence in real environments using autostereoscopic immersive display. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. Piscataway, NJ, USA : IEEE Press, 2011, S. 157–162

- [Tari u. a. 2005] TARI, B. ; YEMEZ, Y. ; ÖZKASAP, Ö ; CIVANLAR, R.: Progressive view-dependent Transmission of 3D Models over lossy Networks. In: *SIGNAL PROCESSING CONFERENCE. EUROPEAN. 13TH 2005. (EUSIPCO 2005)*. Red Hook, NY, USA : Curran Associates, Inc., 2005, S. 1479–1482
- [Taylor u. Cowley 2011] TAYLOR, Camillo J. ; COWLEY, Anthony: Fast Scene Analysis Using Image and Range Data. In: *IEEE International Conference on Robotics and Automation*. Piscataway, NJ, USA : IEEE Press, 2011
- [TEPCO 2013] TEPCO: *TOKYO ELECTRIC POWER COMPANY - Photos and Videos Library*. http://photo.tepco.co.jp/en/date/2011/201111-e/111105_01e.html. URL http://photo.tepco.co.jp/en/date/2011/201111-e/111105_01e.html, Abruf: 20.02.2013
- [Thrun u. a. 2003] THRUN, S. ; HAHNEL, D. ; FERGUSON, D. ; MONTEMERLO, M. ; TRIEBEL, R. ; BURGARD, W. ; BAKER, C. ; OMOHUNDRO, Z. ; THAYER, S. ; WHITTAKER, W.: A system for volumetric robotic mapping of abandoned mines. In: *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*. Piscataway, NJ, USA : IEEE Press, 2003, S. 4270–4275
- [Thrun u. a. 2005] THRUN, Sebastian ; BURGARD, Wolfram ; FOX, Dieter: *Probabilistic robotics*. Cambridge, MA, USA : MIT Press, 2005 (Intelligent robotics and autonomous agents). – ISBN 978–0262201629
- [Tian u. AlRegib 2006] TIAN, D. ; ALREGIB, G.: On-demand transmission of 3D models over lossy networks. In: *Signal Processing: Image Communication* 21 (2006), Nr. 5, S. 396–415
- [ToFExpert 2010a] TOFEXPERT: *File:PMD CamBoard.png - Wikipedia, the free encyclopedia*. URL http://en.wikipedia.org/wiki/File:PMD_CamBoard.png, Abruf: 18.9.2012
- [ToFExpert 2010b] TOFEXPERT: *File:PMDCamCube.jpg - Wikipedia, the free encyclopedia*. URL <http://en.wikipedia.org/wiki/File:PMDCamCube.jpg>, Abruf: 18.9.2012
- [Tombari u. a. 2011] TOMBARI, F. ; SALTI, S. ; DI STEFANO, L.: A combined texture-shape descriptor for enhanced 3d feature matching. In: *Image Processing (ICIP), 2011 18th IEEE International Conference on*. Piscataway, NJ, USA : IEEE Press, 2011, S. 809–812
- [Triebel u. a. 2006] TRIEBEL, Rudolph ; PFAFF, Patrick ; BURGARD, Wolfram: Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing. In: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. Piscataway, NJ, USA : IEEE Press, 2006, S. 2276–2282. – DOI 10.1109/IROS.2006.282632
- [University of Bonn 2012] UNIVERSITY OF BONN: *NimbRo*. URL <http://www.nimbrot.net>, Abruf: 20.8.2012
- [Vatti 1992] VATTI, Bala R.: A generic solution to polygon clipping. In: *Commun. ACM* 35 (1992), Nr. 7, S. 56–63. – DOI 10.1145/129902.129906

- [Waschbüsch u. a. 2004] WASCHBÜSCH, M. ; GROSS, M. ; EBERHARD, F. ; LAMBORAY, E. ; WÜRMLIN, S.: Progressive compression of point-sampled models. In: *Proceedings of the eurographics symposium on point-based graphics*. Aire-la-Ville, Schweiz : Eurographics Association, 2004, S. 95–103
- [Willow Garage 2012] WILLOW GARAGE: *PR2 Overview*. URL <http://www.willowgarage.com/pages/pr2/overview>, Abruf: 27.7.2012
- [Wurm u. a. 2011] WURM, Kai ; HENNES, Daniel ; HOLZ, Dirk ; RUSU, Radu B. ; STACHNISS, Cyrill ; BURGARD, Wolfram: Hierarchies of Octrees for Efficient 3D Mapping. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. Piscataway, NJ, USA : IEEE Press, 2011, S. 4249–4255
- [Wurm u. a. 2010] WURM, K.M. ; HORNUNG, A. ; BENNEWITZ, M. ; STACHNISS, C. ; BURGARD, W.: OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In: *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*. Piscataway, NJ, USA : IEEE Press, 2010
- [Yang u. Lee 1999] YANG, M. ; LEE, E.: Segmentation of measured point data using a parametric quadric surface approximation. In: *Computer-Aided Design* 31 (1999), Nr. 7, S. 449–457
- [Yang u. a. 2004] YANG, S. ; KIM, C.-S. ; KUO, C.-C.J.: A Progressive View-Dependent Technique for Interactive 3-D Mesh Transmission. In: *IEEE Transactions on Circuits and Systems for Video Technology* 14 (2004), Nr. 11, S. 1249–1264. – DOI 10.1109/TCSVT.2004.835153
- [Zalik u. a. 1998] ZALIK, B. ; GOMBOSI, M. ; PODGORELEC, D.: A Quick Intersection Algorithm for Arbitrary Polygons. In: *Spring Conference on Computer Graphics (SCCG98)*. Budmerice, Slowakei : Comenius University, 1998, S. 195–204
- [Zhan u. a. 2010] ZHAN, Qingming ; LIANG YU ; LIANG, Yubing: A point cloud segmentation method based on vector estimation and color clustering. In: *Information Science and Engineering (ICISE), 2010 2nd International Conference on*. Piscataway, NJ, USA : IEEE Press, 2010, S. 3463–3466. – DOI 10.1109/ICISE.2010.5691038

Der teil-autonome Roboterbetrieb kann mittelfristig die Leistungsfähigkeit und Bedienbarkeit von mobilen Servicerobotern deutlich erhöhen. Dabei erfüllt der Roboter im Normalbetrieb seine Aufgaben autonom. In Ausnahmefällen kann ein menschlicher Bediener eingreifen und den Roboter fernsteuern. Ein wichtiger Aspekt bei der Umsetzung des teil-autonomen Betriebs ist dabei die Weiterentwicklung der Wahrnehmungsfähigkeiten. Diese Arbeit behandelt die Entwicklung eines Verfahrens zur 3-D-Umgebungserfassung für teil-autonome Roboter. Das erzeugte Umgebungsmodell unterstützt sowohl den Bediener bei der Fernsteuerung des Roboters als auch den autonomen Roboterbetrieb.

ISBN 978-3-8396-0674-2



FRAUNHOFER VERLAG