

Design of a mobile platform for the evaluation of Localization and Mapping Algorithms

Systementwurf für die Lokalisierung und Kartierung mobiler Plattformen

Stefan Hensel*, Marin B. Marinov**, Christoh Kehret*

* University of Applied Sciences Offenburg, Department for Electrical Engineering, Badstraße 24, D-77652 Offenburg, Germany, stefan.hensel@hs-offenburg.de

** Technical University of Sofia, Faculty of Electronic Engineering and Technologies, Department of Electronics, 8, Kliment Ohridski Blvd., BG-1756 Sofia, Bulgaria, mbm@tu-sofia.bg

Abstract — In this thesis various algorithms of simultaneous localization and mapping (SLAM) are evaluated. The methods use different sensor combinations to determine the current position and to create a map. Two approaches are compared. Firstly, the position is determined exclusively on the basis of consecutive scans of a LiDAR system and the resulting point cloud of the environment is made available as a three-dimensional map. On the other hand, the combination of laser scan data, odometry and inertial measurements is carried out on the basis of an online SLAM procedure in which the position relative to the map is determined and the map is simultaneously created and updated taking into account the current position estimate. In this thesis, an experimental platform has been developed which is suitable for evaluation under real conditions.

Zusammenfassung — In dieser Arbeit werden diverse Algorithmen der simultanen Lokalisierung und Kartierung (SLAM) evaluiert. Die Verfahren nutzen für die Aufgabe unterschiedliche Sensorkombinationen um die aktuelle Position zu bestimmen und eine Karte zu erstellen. Es werden zwei Ansätze verglichen. Zum einen wird die Position ausschließlich auf Basis konsekutiver Scans eines LiDAR-Systems bestimmt und die resultierende Punktwolke der Umgebung als dreidimensionale Karte zur Verfügung gestellt. Zum anderen erfolgt die Kombination von Laserscandaten, Odometrie und Inertialmessungen auf Basis eines online SLAM-Verfahrens, in dem die Position relativ zur Karte bestimmt und die Karte gleichzeitig unter Berücksichtigung der aktuellen Positionsschätzung erstellt und aktualisiert wird. In der Arbeit wurde eine experimentelle Plattform aufgebaut, die für eine Evaluation unter realen Bedingungen geeignet ist.

I. EINFÜHRUNG

Die Navigation und Ortsbestimmung in einer unbekanntem Umgebung stellt seit Beginn der mobilen Robotik die fundamentale Herausforderung für den Einsatz autonomer Systeme im praktischen Einsatz dar. Während die ersten Ansätze militärischen Systemen und kostenintensiven Fluganwendungen vorbehalten waren, erlaubt die kontinuierliche Weiterentwicklung der Algorithmen und eingesetzten Sensorsystemen mittlerweile die Anwendung in der zivilen Nutzung. Durch den stetig steigenden Grad der Komplexität für die detaillierte Umwelterfassung, die beispielsweise im innerstädtischen autonomen Fahren benötigt wird, und die geforderte Flexibilität der Einsatzmöglichkeiten mobiler Plattformen haben sich bestimmte Sensorsysteme und deren Kombination als außerordentlich erfolgreich herausgestellt [1]. Die hohe Informationsdichte optischer Systeme und die geforderte Güte für die genutzten Karten stellen zudem die Mindestanforderungen an die genutzten Kartierungs- und Lokalisierungsalgorithmen. Die Familie der simultanen Lokalisierungs- und Kartierungsalgorithmen (SLAM) haben sich hierbei als besonders erfolgreich erwiesen [2].

II. MATERIALEN UND METHODEN

Dieses Kapitel gibt einen Überblick über die eingesetzten Sensortypen und -systeme. Das als Sensorplattform verwendete mobile System und die durchgeführten Anpassungen werden danach erläutert, bevor detailliert auf die verwendete Software eingegangen wird, die sich aus der Middleware des Robot Operating Systems (ROS) und den letztendlich evaluierten Algorithmen zusammensetzt.

A. Hardwarekomponenten

1) 3D-Lasersensor

In der präzisen und ortsgenauen Vermessung von Objekten mit einer hohen Messrate haben sich zum aktuellen Zeitpunkt Laserscanner gegenüber optischen und radarbasierten Systemen durchgesetzt [3]. Die in anspruchsvollen Aufgaben eingesetzten 3D-Laserscanner werden hierbei als LiDAR (engl. Light Detection And Ranging) bezeichnet.

2) LiDAR Velodyne VLP-16

In der Arbeit wird der VLP-16 der Firma Velodyne eingesetzt. Er besteht aus 16 vertikal angeordneten Lasern, die sich in X° Intervallen auffächern und einen Öffnungswinkel von Y° aufweisen. Durch die Rotation mit 10 Hz werden 300 000 Laserpunkte gewonnen, die Objekte bis zu einer Distanz von 100 Metern und einem horizontalen Sichtfeld von 360° detektieren können. Das System arbeitet mit dem Pulslaufzeitverfahren und bestimmt die Entfernung des reflektierenden Objekts mit einer Abweichung von 10 cm. Durch die Miniaturisierung des Systems konnte der Preis in den vierstelligen Bereich verringert werden, was den Sensor zum de facto Standard im autonomen Fahren macht [3].

3) Inertiale Messeinheit (IMU)

Während optische Sensorsysteme wie Laser und Kameras die Positionsbestimmung bezüglich der Umwelt erlauben, werden sogenannte inertielle Sensoren dazu genutzt, die relative Bewegung einer Plattform aus deren Bewegung zu bestimmen.

Eine inertielle Messeinheit (IMU) besteht aus drei Beschleunigungssensoren, drei Drehratensensoren (Gyros) und für gewöhnlich drei Magnetfeldsensoren, die jeweils als Triaden orthogonal zueinander angeordnet werden und auf Basis der Massenträgheit die aktuelle Orientierung im Raum bestimmen.

Die Sensordatenfusion der Einzelsensoren erlaubt hierbei die Kompensation vorhandener Drift, welche durch die Integration der Drehraten und Beschleunigungen unweigerlich auftritt [4]. Die Lage im Raum wird durch die Drehratensensor Triade bestimmt, deren Drift um die Nick- und Wankachse durch die mit Hilfe der Beschleunigungs-sensortriade gemessenen Erdanziehung kompensiert wird. Die Drift des Gierwinkels wird durch die Bestimmung der Nordrichtung kompensiert, die wiederum mit der Hallsensortriade aus dem Erdmagnetfeld bestimmt wird. Die letztendliche Schätzung der Lage wird entweder mit Hilfe nichtlinearer Zustandsbeobachter, z.B. dem extended Kalman Filter [5], oder orthogonalen Filtern, beispielsweise in [6] beschrieben, durchgeführt.

4) XSens IMU

In dieser Arbeit wurde die IMU MTi-300 von XSens verwendet. Diese besteht aus den beschriebenen Sensortriaden und schätzt die Lage mit Hilfe eines Kalman Filters. Die Unsicherheit des Gierwinkels liegt unter 1°, die Nick- und Rollwinkel werden mit einer Genauigkeit von 0,2° bestimmt. Vorteile des Systems sind die IP67 Schutzklasse und die Verfügbarkeit der Treiber, die eine Integration auf der Roboterplattform erleichtern und einen Einsatz im Außenbereich erlauben.

5) Mobile Plattform – Clearpath Robotics Husky

Als Sensorträger kommt die mobile Roboterplattform Husky von Clearpath Robotics zum Einsatz. Es handelt sich um eine mobile Basis, welche durch die Traglast von 75 kg mit einer Vielzahl von Sensoren, Rechnern und Batterien für die Energieversorgung ausgerüstet werden kann. Durch die kompakten Abmaße, den Radstand, gegebene Radhöhe sowie den Skid-Steer-Antrieb kann ein Einsatz im Innen- und Außenbereich erfolgen, was die Evaluation der Kartierung für strukturierte und natürliche Umgebungen erlaubt. Die Eigenschaften des Systems sind zur Übersicht in Tabelle I aufgeführt, in Abbildung 1 ist die Plattform im Rahmen des Gesamtsystems zu sehen.

TABELLE I. KENNDATEN DES HUSKY ROBOTERS

Eigenschaften	Husky
Größe [mm]	990x670x390
Gewicht [kg]	50
Max. Traglast [kg]	75
Max. Geschwindigkeit [m/s]	1.0
Laufzeit [h]	3
Treiber und APIS	ROS, C++, Mathworks

Die Steuerung der Einheit erfolgt über einen Rechner mit x64 Architektur, der mit einer Intel Core-i5 CPU, 8 GByte RAM und dem Betriebssystem Linux Ubuntu sowie ROS ausgestattet ist.

B. Softwarekomponenten

1) ROS – Robot operating system

Für die Implementierung der Algorithmen und Sensoren wurde die Middleware ROS [7] verwendet. Unter einer Middleware versteht man eine Verteilungsplattform, welche im abstrakten Sinn die Kommunikation zwischen Prozessen

erlaubt und regelt. Die vorgestellten Sensoren verfügen ebenso wie der Husky Roboter über Treibersoftware für ROS, was die Integration im Gesamtsystem deutlich vereinfacht. ROS bietet zudem eine Vielzahl von existierenden Programmibliotheken, die die Bereiche Visualisierung, Datenaufzeichnung, Simulation und Systemüberwachung beinhalten. Selbst erstellte Module können in den Programmiersprachen C++, Python oder mit Matlab/Simulink erstellt werden.

III. VEFAHREN ZUR KARTENERSTELLUNG

A. Laser Odometry and Mapping – LOAM

Laser Odometry and Mapping ist eine echtzeitfähige Methode, um mit Hilfe von 3D LiDAR-Daten die aktuelle Position zu schätzen. Die gefahrene Distanz zwischen zwei konsekutiven Laserscans wird mit Hilfe eines Point-Matching-Verfahrens bestimmt und die aktuelle Position aus der Integration bestimmt. Das Verfahren ist hierbei zweistufig und besteht aus einer schnellen lokalen Positionsschätzung und einem langsameren Korrekturschritt, der aus den aufgezeichneten Messungen eine Punktwolke mit Hilfe des Blockbündelausgleichs erstellt. Die Punktwolke fungiert als Karte, welche wiederum als Bezugssystem für die erneute Positionsbestimmung genutzt werden kann. Diese Parallele Struktur erlaubt eine hohe Berechnungsgeschwindigkeit bei gleichzeitig hoher Präzision. Detaillierte Informationen finden sich in [8] und [9].

Das System nutzt explizit keine zusätzlichen Sensoren, so dass das Verfahren in die Klasse der Scan-Matching basierten SLAM-Methoden zu zählen ist. Für die Evaluation ausgewählt wurde es aufgrund seiner geringen Rechenlast, was die Echtzeitfähigkeit auf handelsüblichen CPUs erlaubt, zudem sind Varianten des Algorithmus auf den vordersten Rängen neutraler Leistungsbenchmarks, z.B. der KITTI-Datenbank [10], zu finden.

B. Cartographer SLAM

Methoden der simultanen Lokalisierung und Kartierung haben eine lange Historie in der Robotik [1] und können grob in zwei relevante Klassen eingeteilt werden: Probabilistische Verfahren basierend auf Zustandsschätzern (Extended-Kalman-Filter-SLAM) und Verfahren, die einen Graphen aus den vorhandenen Posen (Position und Orientierung) und den Sensorbeobachtungen der Umgebung erstellen und diesen optimieren (engl. Pose-Graph-SLAM).

Bei Cartographer handelt es sich um ein Pose-Graph-SLAM-Verfahren, basierend auf LiDAR-Sensoren [11]. Eine Erweiterung zu den klassischen SLAM-Verfahren stellt die Unterteilung in einen lokalen, echtzeitfähigen, SLAM-Prozess, der mit einer globalen Optimierungsstufe verknüpft wird.

Der lokale Prozess weist eine starke Ähnlichkeit zu LOAM auf und erzeugt aus konsekutiven Laserscandaten eine konsistente Darstellung der Umgebung die ebenfalls für die Berechnung der zurückgelegten Strecke genutzt werden kann.

Die globale Optimierung beruht auf einem Blockbündelausgleichsverfahren. Diese Optimierung von dünnbesetzten Matrizen beruht auf photogrammetrischen Algorithmen und korrigiert die Ausrichtung der Ursprungspunkte der lokalen Karten, was sich über diese wiederum auf die geschätzte Position des Roboters auswirkt.

Die verwendete Implementierung von Cartographer-SLAM stellt zwei Optionen zur Verfügung, die eine nutzt den 3D-Laserscan und erzeugt eine zweidimensionale Karte, die auf den horizontalen Laserstrahlen beruht, die andere erzeugt eine interne 3D-Karte mit allen Laserstrahlen, die allerdings auf eine 2D-Darstellung zurückprojiziert wird.

C. Aufnahmearten des Systems

In Abbildung 2 ist der Aufbau des Roboters dargestellt. Die Kommunikation der einzelnen Komponenten erfolgt über ROS, die Bewegung des Roboters erfolgt über einen Controller, so dass keine autonome Exploration erfolgt, sondern ausschließlich die Kartierung und Lokalisierung erprobt wird.

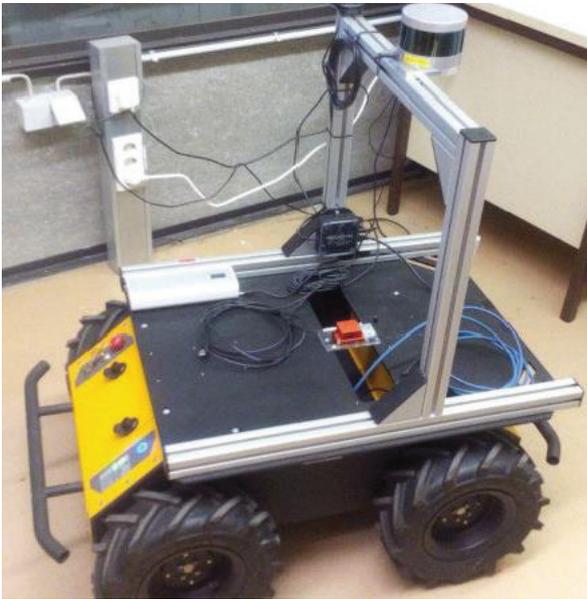
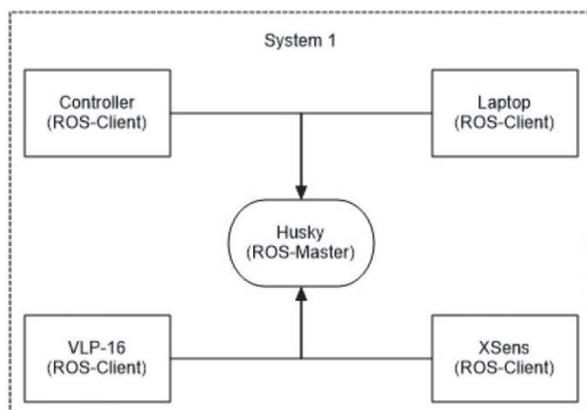


Abb. 1. Aufbau des vollen Systems

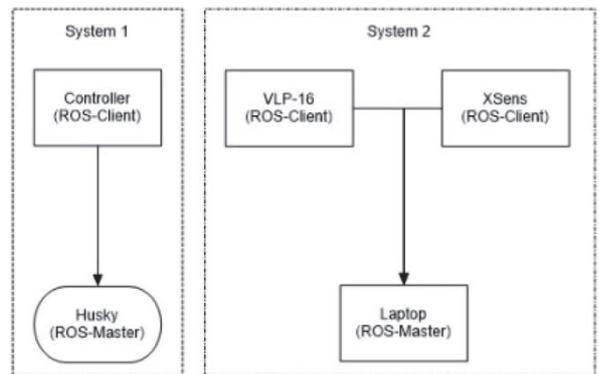
Für die Evaluation wurden zwei Arten der Datenakquise realisiert, welche in Abbildung 3 dargestellt sind. Die erste Möglichkeit erlaubt es, die Sensoren über den eingebauten Rechner zu steuern und mit Hilfe eines externen Laptops die Daten über eine SSH-Schnittstelle (LAN oder WLAN) aufzuzeichnen. Dieser Ansatz bietet den Vorteil, neben den Sensordaten auch direkt die fusionierte Odometrie des Huskys sowie die Steuerdaten des Controllers aufnehmen zu können.

Die zweite Methode trennt die Datenaufnahme, die mit Hilfe eines zweiten Laptops erfolgt, von der Steuerung des Roboters.

Obschon die erste Methode den direkten Einsatz und online-SLAM-Berechnung ermöglicht, wurde sich im Rahmen der Veröffentlichung für den zweiten Ansatz entschieden, da hier eine höhere Datenrate und lückenlose Aufzeichnung, im Speziellen die Laserscan-Daten betreffend erreicht werden konnte. Die Erstellung der Karte und Berechnung der Position kann danach offline mit Hilfe des ROS-Frameworks simuliert werden, um eine Reproduzierbarkeit der Parametereinstellungen sicherzustellen.



(a) Indirekte Aufnahme



(b) Direkte Aufnahme

Abb. 2. Aufnahmearten

IV. ERGEBNISSE UND AUSWERTUNG

Mit dem beschriebenen System wurden zuerst die qualitativen und danach die quantitativen Aspekte untersucht und beurteilt.

A. Qualitative und quantitative Aspekte bei der Untersuchung

1) Qualitative Auswertung

In der qualitativen Auswertung werden folgende Punkte beleuchtet:

- Kartentyp: Hierbei handelt es sich um den Typ der Karte, entweder die Karte wird in 2D oder 3D dargestellt.
- Ressourcen: Ressourcen bezeichnet die Hardwareanforderungen. Dabei bedeutet schlecht, dass eine hohe Rechenleistung benötigt wird.
- Ansicht: Hiermit wird die Auflösung der Karte bewertet. Je höher die Auflösung ist, desto besser können Menschen Informationen aus der Karte entnehmen.
- Navigation: Unter dem Aspekt Navigation wird angegeben, ob sich eine spätere Navigation mithilfe der erzeugten Karte bewerkstelligen lässt.
- Robustheit: Mit der Robustheit wird eine qualitative Aussage darüber geschlossen, wie robust der Algorithmus arbeitet.
- Zubehör: Hierdurch wird der benötigte Aufwand beschrieben. In dieses Kriterium geht vor allem die benötigte Zahl an Sensoren ein.

2) Quantitative Auswertung

Die quantitative Auswertung erfolgt auf Basis der Start- und End-Pose des Roboters, hierbei gelten folgende Definitionen:

- Absoluter Fehler: Beim absoluten Fehler handelt es sich, um die gemessenen Abweichungen der Endposition mit der Anfangsposition. Die Abweichung wird einerseits in X-Richtung und andererseits in Y-Richtung ermittelt. Die Einheit ist Meter.
- Relativer Fehler: Der relative Fehler bezeichnet den prozentualen Fehler. Dieser gibt an, um wie viel Prozent der Algorithmus falsch liegt in Abhängigkeit von der gefahrenen Gesamtstrecke. Die Formel für den relativen Fehler lautet:

$$Rel. Fehler = \frac{abs. Fehler}{Strecke} 100\%. \quad (1)$$

Der Fehler gibt an, wie genau und robust die Berechnungen sind. Dies ist die wichtigste Eigenschaft und deswegen liegt hierbei das Augenmerk und die Relevanz der Arbeit.

B. Exemplarische Experimentelle Ergebnisse

Die Evaluation der Verfahren erfolgte anhand mehrere typischen Sequenzen für mobile Roboter. Als Beispiel ist in Abb. 3 eine industrielle Anlage mit hohem Strukturanteil dargestellt. Es ist jeweils ein Beispielfotobild, die erstellte LOAM-Punktswolke und die in Cartographer erstellte 2D Karte gezeigt. Zusätzlich erfolgte die Auswertung mit unterschiedlichen Streckenlängen und zusätzlichen Szenarien, einmal in Bürogebäuden mit Korridoren und einmal in Außenbereichen mit Sträuchern und Bäumen.

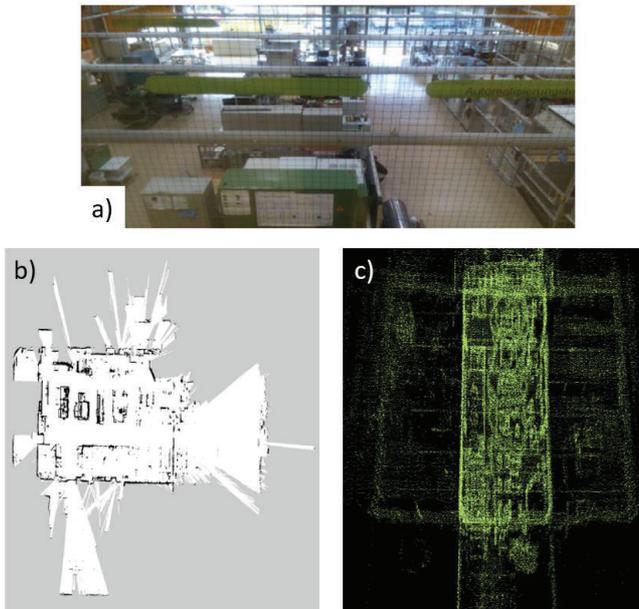


Abb. 3. Exemplarisches Ergebnis der Auswertung mit a) Videobild, b) Cartographer "D-Karte und c) LOAM 3D-Punktswolkenkarte

Das quantitative Ergebnis der Testfahrten (Abb. 3 findet sich in Industry Double wieder) für den Cartographer-SLAM ist in Tabelle II dargestellt, welche auch für die anderen Sequenzen die Ergebnisse als relativen Fehler der Lokalisierungs-genauigkeit in x- und y-Koordinatenrichtung auflistet.

TABELLE II. QUANTITATIVE AUSWERTUNG DES CARTOGRAPHER-2D-VERFAHRENS

Umgebung	Gesamtlänge [m]	rel. Fehler X [%]	rel. Fehler Y [%]
IndoorShort	61	0,033	0,016
IndoorShort3	183	0	0
IndoorMiddle	152	1,7	0,2
IndoorMiddle3	456	0,4	0,02
IndoorBig	475	0,48	0,18
IndoorBigLoop	500	0,41	0,06
IndustryLoop	50	0,6	0,04
IndustryReverse	100	0,06	0,02
IndustryDouble	115	0,18	0,02
OutdoorShort	75	0,64	0,65
OutdoorShort3	225	0,64	0,65
OutdoorMiddle	150	0,28	0,12
OutdoorBig	500	0,24	0,14

Zum Vergleich sind in Tabelle III die Ergebnisse für den 3D-LOAM-SLAM aufgeführt. Die Ergebnisse zeigen deutlich, dass Cartographer-SLAM dem LOAM-Verfahren in puncto Genauigkeit überlegen ist. Dies war durch den zusätzlichen Optimierungsschritt auch zu erwarten. Der Vorteil des LOAM-Verfahrens zeigt sich eher in der hohen Bearbeitungs-geschwindigkeit und der Möglichkeit auf einfache Weise eine relative Lokalisierung zu erhalten.

Der systematisch größte Vorteil von Cartographer-SLAM besteht in der Erstellung der Rasterkarte, welche als Eingang für die spätere Roboternavigation dient. Die 3D-Punktswolke des LOAM-Verfahrens stellt dagegen eine visuell ansprechende Darstellung des befahrenen Gebietes dar, welche auch von Menschen sehr gut interpretiert werden kann.

TABELLE III. QUANTITATIVE AUSWERTUNG DES LOAM-ALGORITHMUS

Umgebung	Gesamtlänge [m]	rel. Fehler X [%]	rel. Fehler Y [%]
IndoorShort	61	0,08	0,43
IndoorShort3	183	0,47	0,01
IndoorMiddle	152	1,66	0,68
IndoorMiddle3	456	-	-
IndoorBig	475	0,18	0,24
IndoorBigLoop	500	0,07	0,27
IndustryLoop	50	1,22	0,14
IndustryReverse	100	0,54	0,04
IndustryDouble	115	0,24	0,04
OutdoorShort	75	0,31	0,11
OutdoorShort3	225	0,35	0,03
OutdoorMiddle	150	0,46	0,15
OutdoorBig	500	0,03	0,10

V. ZUSAMMENFASSUNG UND AUSBLICK

In dem vorliegenden Aufsatz konnte das System für die Implementierung und Evaluation komplexer Lokalisierungs- und Kartierungsverfahren vorgestellt werden. Beide Verfahren eignen sich grundlegend für eine robuste Lokalisierung in fordernden Szenarien wie Pick and Place Anwendungen in der Industrie 4.0, Erkundung und Überwachung in der Feldrobotik oder Anwendungen der Servicerobotik. Komplexe SLAM-Ansätze zeigen sich hier systemisch überlegen, da alle Informationen einer Vielzahl an Sensoren kombiniert und genutzt werden kann.

Nächste Schritte bestehen nun in der Nutzung der erstellten Karten, was sich im Falle der präzisen 3D-Punktswolke als komplexe Aufgabe erweist.

LITERATURVERZEICHNISS

- [1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid and J. Leonard, "Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age," vol. 32, no. 6, pp. 1309-1332, December 2016.
- [2] Sebastian Thrun, Wolfram Burgard, Dieter Fox, *Probabilistic Robotics*, ISBN:978-0-262-20162-9, Cambridge : MIT Press, 2005.
- [3] Andreas Nüchter, *3D Robotic Mapping - The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom*, Berlin, Heidelberg: Springer Science & Business Media, 2009.
- [4] J. Wendel, *Integrierte Navigationssysteme: Sensordatenfusion, GPS und Inertiale Navigation*, 2009.
- [5] P. Corke, Robotics, *Vision and Control*, 2nd. Edition, Springer, 2017.
- [6] R. Mahony, T. Hamel and J.-M. Pflimlin, "Nonlinear Complementary Filters on the Special Orthogonal Group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203-1217, 2008.
- [7] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.
- [8] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," *Robotics Science and Systems*, 2014.
- [9] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots*, vol. 41, no. 2, pp. 401-416, 2017.
- [10] A. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? The kitti vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [11] W. Hess, D. Kohler, H. Rapp and D. Andor, "Real-Time Loop Closure in 2D LIDAR SLAM," in *Robotics and Automation (ICRA)*, pp. 1271-1278, 2016.