



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Anwendungen 1 - WiSe 2010/11

Ben Struss

2D/ 3D SLAM

in halbstatistischen Indoor Umgebungen

Inhaltsverzeichnis

1 Einführung	3
1.1 Motivation	4
1.2 Vorstellung der Plattform	4
2 Das SLAM Problem	5
2.1 Vorgehen bei der Kartierung	5
2.2 Darstellung des Weltmodells	6
2.3 Semantische Karten	9
2.4 SLAM in nicht-statischen Umgebungen	10
3 Zusammenfassung	12
3.1 Ausblick	12
3.2 Risiken	12
Literatur	14

1 Einführung

Waren Roboter lange Zeit der industriellen oder militärischen Anwendung vorbehalten, so verbreiten sie sich zunehmend auch in privaten und häuslichen Umfeldern. Einer der wohl bekanntesten autonomen Haushaltsroboter ist der, seit 2002 verfügbare, Roomba der Firma iRobot¹. Einmal vom Benutzer gestartet, fährt dieser anhand einfacher Algorithmen den Raum ab und versucht sicherzustellen, dass alle Stellen mindestens einmal erreicht werden. Da die Wegfindung jedoch vor allem aus simpler Kollisionserkennung und optionalen Infrarot-Schranken besteht, kann es verhältnismäßig lange dauern, bis mit hinreichend hoher Wahrscheinlichkeit der komplette Raum abgefahren wurde. Trotz dieser Einschränkungen konnte iRobot bis zum Jahr 2010 über 5 Millionen Haushaltsroboter absetzen.²

Dieses Beispiel liefert einen guten Eindruck der Anforderungen, die in privaten Umfeldern an die Eigenschaften von Robotern gestellt werden. So ist es zwar akzeptabel, dass zunächst initialer Aufwand für die Einrichtung und Programmierung des Roboters getrieben wird, es kann den Besitzern jedoch nicht zugemutet werden, die Programmierung des Roboters bei jeder kleinen Änderung der Umgebung anzupassen. Autonome Haushaltsroboter bewegen sich in dynamischen Umfeldern, in denen Menschen verkehren, Möbelstücke wie Stühle und Tische verrückt werden, sowie Gegenstände auf dem Boden liegen können. Um seinen Besitzern tatsächlich Arbeit abnehmen zu können, wird erwartet, dass sich der Roboter selbsttätig an die Veränderungen anpasst und weiterhin seine Aufgaben ausführen kann. Erst im Falle von grundlegenden Umstrukturierungen der Umgebung wie Anpassungen des Grundrisses, Verlegung von relevanten Räumen wie der Küche an andere Stellen oder einer komplett neuen Umgebung sollten manuelle Eingriffe von Menschenhand notwendig werden.

Im weiteren Verlauf des ersten Kapitels wird die konkrete Motivation der Arbeit, sowie die Zielplattform vorgestellt. Kapitel 2 gibt eine Einführung in die SLAM-Thematik und geht näher auf verschiedene, für diese Arbeit relevante Aspekte von SLAM ein. Abschließend gibt Kapitel 3 eine Zusammenfassung des Problems, eine Risikoanalyse sowie einen Ausblick auf das weitere Vorgehen.

¹<http://www.irobot.com/>

²iRobot History: <http://www.irobot.com/sp.cfm?pageid=203>

1.1 Motivation

An der HAW-Hamburg wird bereits seit 15 Jahren an der Entwicklung autonomer Roboter geforscht und eigens zu diesem Zweck ein Robotics Labor eingerichtet. In Kombination mit dem relativ neuen Forschungsbereich *Living Place Hamburg*³, der sich der Erforschung und Entwicklung von Technologien und Helfern im Wohnumfeld befasst, soll ein autonomer Service-roboter entwickelt werden, der einfache Hol- und Bringaufgaben ausführen kann. Dazu muss der Roboter die Fähigkeit haben seine Umgebung mittels verschiedener Sensoren wahrzunehmen, sich zu Lokalisieren, Wege zu planen und die relevanten Gegenstände zu identifizieren. Als Plattform für dieses Vorhaben wurde von der Hochschule ein SCITOS G5 angeschafft, der im folgenden Abschnitt kurz vorgestellt wird.

1.2 Vorstellung der Plattform

Der SCITOS G5 wurde von der Firma Metralabs⁴ als multifunktionale Industrie- und Forschungsplattform entwickelt und kann, in Abhängigkeit vom Einsatzgebiet, mit unterschiedlichen Aufbauten bestückt werden. Die Roboterbasis ist dabei mit 24 Ultraschallabstandssensoren und einem umlaufenden Bumper zur Kollisionserkennung, sowie einem integrierten Mini-ITX PC mit leistungsfähigem dualcore Mobilprozessor ausgerüstet. Außerdem sind direkt an den Rädern zwei Tickzähler integriert, die pro Radumdrehung 460 Signale geben und somit eine Abschätzung der zurückgelegten Strecke ermöglichen. Zusätzlich wurde an der Front ein 2D Laserscanner des Herstellers Leuze Electronis⁵ installiert, der präzise Entfernungsmessungen in der Ebene ermöglicht. Um als Serviceroboter agieren zu können, ist auf der Nutzplattform ein 5-Achs Gelenkarm der Firma SCHUNK⁶ montiert, der an seiner Spitze mit einem Greifer ausgestattet ist. Dieser ermöglicht es dem Roboter Gegenstände zu fassen und zu transportieren. Abbildung 1 zeigt den Roboter, wie er (Stand: Jan 2011) im Robotics Labor der HAW Hamburg steht.



Abbildung 1: SCITOS G5

³http://livingplace.informatik.haw-hamburg.de/blog/?page_id=47

⁴<http://www.metralabs.com/>

⁵<http://www.leuze.de/>, Modell ROTOSCAN RS4-2E

⁶<http://www.schunk.de/>

2 Das SLAM Problem

SLAM steht für *Simultaneous Localization and Mapping* und beschreibt eines der fundamentalen Probleme aller autonomen Roboter. Um tatsächlich autonom agieren zu können, benötigt ein Roboter sowohl Informationen über seine Umgebung (Mapping), als auch Kenntniss über seine gegenwärtige Position (Localization) innerhalb dieser Umgebung. Nur auf Basis beider Informationen kann der Roboter eine Entscheidung über die folgenden Aktionen treffen. Die Schwierigkeit liegt darin, dass keines von beidem ohne das jeweils andere ermittelt werden kann und daher beides zeitgleich auf Basis von Sensordaten geschätzt werden muss. Die Wissenschaft befasst sich seit gut 20 Jahren mit diesem Thema und hat seitdem eine Vielzahl verschiedener Ansätze hervorgebracht, die sich je nach Anwendungsfeld grundlegend voneinander unterscheiden. (Thrun und Leonard, 2008; Durrant-Whyte und Bailey, 2006)

Dieses Kapitel gibt zunächst einen Überblick über einige grundlegenden Bereiche aller SLAM-Verfahren und geht später mehr auf die besonderen Anforderungen von nicht-statischen Umgebungen ein. Hierbei wird der Fokus auf die für Serviceroboter relevanten Themen und Indooranwendungen gelegt.

2.1 Vorgehen bei der Kartierung

Die Ansätze für die Erstellung der Karte als Grundlage für die Lokalisierung werden in zwei wesentliche Kategorien eingeordnet. Dies sind zum einen *Offline*-Verfahren, bei denen die Karte zunächst in einem gesonderten Schritt erfasst wird und dem gegenüber die *Online*-Verfahren, wo der Roboter die Karte im autonomen Betrieb parallel zu seinen Aufgaben erstellt. (Frese u. a., 2010)

Bei der *Offline*-Erfassung der Karte wird der Roboter von einem Menschen ferngesteuert durch die Umgebung gelenkt und zeichnet bei dieser Fahrt sämtliche Sensordaten und Messwerte auf. Die eigentliche Erstellung der Karte geschieht hinterher auf Basis der aufgezeichneten Messdaten. Da dieser Schritt zumeist nicht zeitkritisch ist, können hier deutlich rechenaufwendigere und damit auch genauere Verfahren zum Einsatz kommen. Gerade feingranulare 3D-Messungen mit entsprechend vielen Messwerten erfordern große Mengen Speicherplatz und Rechenleistung, die der Roboter selber möglicherweise nicht bereitstellen kann. Die so entstandene Karte dient für den weiteren Betrieb als statische Grundlage für die Lokalisierung und bietet ein festes Koordinatensystem für alle auf dem Roboter laufenden Anwendungen. Zusätzlich können in dieser Karte semantische Bereiche und Orte definiert werden, die für den Roboter relevant sind oder die aus Sicherheitsgründen nicht befahren werden dürfen. Hierauf wird in Kapitel 2.3 näher eingegangen. In (Thrun u. a., 2000) wird ein solches Vorgehen anhand des autonomen Roboters „Minerva“ beschrieben, der Führungen durch das Smithsonian Museum of American History in Washington D.C. anbieten konnte.

Demgegenüber befassen sich *Online*-SLAM-Verfahren mit dem Problem, dass im autonomen Betrieb entweder noch gar keine Karte vorhanden, diese unvollständig und/oder ständiger Änderung unterworfen ist. Dieser Anwendungsfall ist der komplexeste und erfordert ein hohes Maß an Abstimmung aller Komponenten. Ausgehend von einem Startpunkt aus erkundet der Roboter die Umgebung und erzeugt aus den Sensormessungen ein Kartenmodell. Da die Messungen selber jedoch fehlerbehaftet sind und nur jeweils relativ zum derzeitigen Standort des Roboters erfasst werden können, welcher ebenso auf einer Schätzung des bisher zurückgelegten Weges basiert, entstehen mit zunehmender Strecke und Dauer teils gravierende Abweichungen in den absoluten Koordinaten der beobachteten Merkmale. Sollte der Roboter im späteren Verlauf eine Stelle ein zweites Mal passieren und dies feststellen, so kann er diese Information nutzen um die Karte zu verbessern indem er die Messungen übereinanderlegt und alle Wegpunkte die dazwischen zurückgelegt wurden entsprechend korrigiert. Dieser Vorgang wird als *Loop-Closing* bezeichnet und führt dazu, dass sich die Karte schlagartig ändert und mit ihr alle Koordinaten die sich innerhalb der Loop befinden. Sämtliche Anwendungen auf dem Roboter müssen sich dieses Falles bewusst sein und bei Eintritt informiert werden.

Für einen Serviceroboter, der sich typischerweise in einer relativ überschauberen und begrenzten Umgebung bewegt, bietet sich zunächst eine initiale Offline-Erfassung der Umgebung an. Da jedoch gerade in häuslichen Umfeldern mit häufigen Änderungen zu rechnen ist, müssen zusätzlich auch Online-Komponenten in das System integriert werden. Ansätze hierfür werden in Kapitel 2.4 vorgestellt.

2.2 Darstellung des Weltmodells

Die wohl wichtigste Entscheidung beim Entwurf eines SLAM-Verfahrens ist die Auswahl eines geeigneten Modells zur Abbildung der Umgebung. In der theoretischen Untersuchung des SLAM-Problems wird häufig die sogenannte *Landmark World* (auch *Feature Based SLAM*) verwendet, die auf eindeutig identifizierbaren Merkmalen basiert, die in einem 2- oder 3-dimensionalen Koordinatensystem angeordnet werden und anhand ihrer Beobachtung eine Lokalisierung ermöglichen. (Thrun und Leonard, 2008; Montemerlo u. a., 2002) Eine solche Anordnung von Merkmalen ist sehr effizient, da nur wenige Informationen gespeichert werden müssen. Davon abgesehen sind die Verwendungsmöglichkeiten abseits der Lokalisierung jedoch sehr eingeschränkt. Da keinerlei Informationen über freien Raum und Hindernisse gespeichert werden, ist eine Pfadplanung unmöglich. Es könnten ausschließlich bereits befahrene und gespeicherte Routen verwendet werden.

An dieser Stelle setzen die *Occupancy Grid Maps* an. Hierbei wird ein Grid aus Zellen beliebiger Kantenlänge (z.B. 1-5cm für Indoor-Umgebungen) generiert und für jede Zelle gespeichert, ob sie frei, belegt oder noch unbekannt ist. Abbildung 2 zeigt eine solche Karte, die mittels Überlagerung von 2D-Laserscans generiert wurde. Freie Bereiche sind weiß, Unbekanntes

grau und belegte Zellen dunkel gekennzeichnet. In rot ist der Pfad des erkundenden Roboters dargestellt. Aufgrund der guten Eignung für die Pfadplanung und der Möglichkeit relevante Objekte in dem festen Koordinatensystem der Karte zu verankern, findet dieses Verfahren heute eine weite Verbreitung. Siehe dazu auch: (Thrun u. a., 2000; Riaz u. a., 2010; Meyer-Delius u. a., 2010)

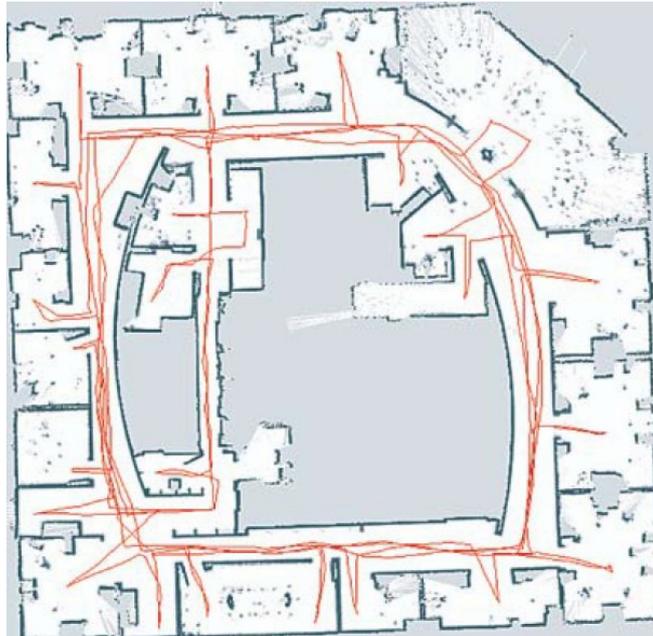


Abbildung 2: 2-dimensionale Occupancy Grid Map, (Thrun und Leonard, 2008)

Für unseren Serviceroboter mit seinem langem Gelenkarm ergibt sich jedoch in der Praxis das Problem, dass Hindernisse, die sich nicht in der Sensorebene befinden, auf einer reinen 2D-Karte unsichtbar sind. Dies können zum Beispiel Tischplatten, an den Wänden befestigte Schränke und Regale oder tief hängende Lampen sein, auf die man mit hoher Wahrscheinlichkeit in fast allen Wohnungen stößt. Als Lösung für dieses Problem können 2,5-dimensionale Karten generiert werden, die in ihren Zellen nicht nur den belegt-Zustand festhalten sondern, passende Sensorik vorausgesetzt, die Information über die geringste Höhe des erkannten Hindernisses speichern. Für eine reine Pfadplanung können bei diesem Verfahren alle Zellen deren Höhe größer als die Roboterhöhe ist, als frei angenommen werden. (Wurm u. a., 2010) Da in einer solchen Karte jedoch sämtliche Informationen über die Oberflächen von Objekten fehlen, ist es zum Beispiel nicht möglich auf ihrer Basis mit dem Gelenkarm des Roboters einen Gegenstand auf einem Tisch abzulegen. Generell wird für die Koordination des Arms ein relativ fein aufgelöstes, vollwertiges 3D-Modell der nahen Umgebung benötigt. Nur so kann sicher-

gestellt werden, dass der Arm in seiner Bewegung nirgendwo anstößt und die zu greifenden Objekte exakt lokalisiert werden können.

Zur Erfassung von 3D-Daten gibt es heutzutage eine ganze Reihe von Sensoren und Möglichkeiten. Angefangen bei stereoskopischen Kameras, über 3D-Laserscanner und verkippbare 2D-Laserscanner, bis hin zu Time-of-Flight Kameras und neuerdings auch das Kinect von Microsoft, welches mittels projiziertem IR-Muster und Triangulation arbeitet. Ihnen allen gemein ist jedoch, dass als Ausgabe eine mehr oder weniger hoch aufgelöste Punktwolke entsteht, siehe Abbildung 3. Diese Punkte können nun in Kombination mit der aktuellen Roboter-Pose (x,y,z-Koordinaten, sowie Rotation in allen 3 Richtungen) in die Weltkarte integriert werden.

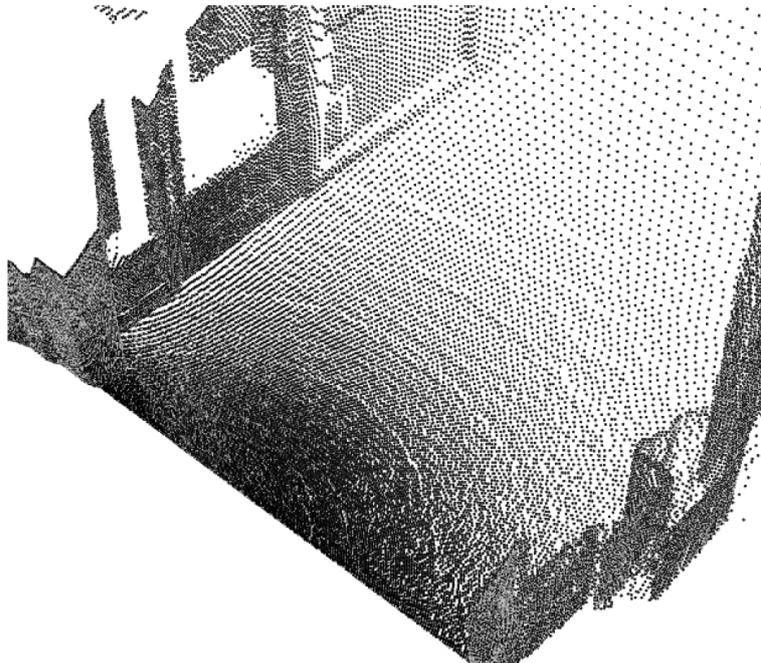


Abbildung 3: Punktwolke eines geneigten 2D-Laserscanners, (Wurm u. a., 2010)

Von Menschen können solche Karten sehr gut interpretiert werden, sie bereitem dem Computer jedoch einige Schwierigkeiten. In einem Scanvorgang können leicht mehrere hunderttausend Punkte entstehen, die schnell große Mengen Speicher belegen. Zudem sind viele Punkte von aufeinanderfolgenden Scans mehrfach redundant und für die gewünschte Auflösung nicht unbedingt notwendig. Auch für die Navigation ist es von Nachteil, dass keine direkte Aussage über freien Raum gemacht werden kann und keine zuverlässigen Informationen über geschlossene Flächen vorliegen.(Hertzberg u. a., 2010) Die naheliegende Lösung des Problems wäre das 2D Occupancy Grid um die dritte Dimension zu erweitern und den 3D Raum in Voxel⁷

⁷das volumetrische äquivalent zu Pixeln

zu zerlegen, deren Seitenlänge der gewünschten Auflösung entspricht. Für jeden Voxel würde dann analog der 2D-Variante der Belegungszustand gespeichert. Nachteil hierbei ist der hohe Speicherbedarf, der beim naiven Ansatz besonders der ineffizienten Speicherung großer zusammenhängender Bereiche gleichen Zustands geschuldet ist. Eine durchschnittlich große Wohnung von 10x10x2,5m würde bei einer Auflösung von 1cm bereits aus 250 Mio. Voxeln bestehen, welche jeweils mindestens 2 Bit zur Kodierung der 3 Zustände benötigen. In (Wurm u. a., 2010) wird mit „OctoMap“ eine Datenstruktur zur effizienten Speicherung der 3D Karte vorgestellt. Es handelt sich dabei um einen Achtfach-Baum in dem jeder Knoten ein Voxel repräsentiert und genau 8 Unterknoten mit jeweils halber Kantenlänge beinhaltet⁸. Pro Knoten wird nicht nur der aktuelle Belegungszustand sondern auch die zugehörige statistische Sicherheit gespeichert. Dies ist nötig, da jede Messung einen gewissen Rauschanteil beinhaltet und Ausreisser so über mehrere Messungen hinweg effektiv unterdrückt werden können. Neben den Vorteilen bei der Größe der Datenstruktur bietet sich auch die Möglichkeit, je nach Anwendungsfall, auf unterschiedlich fein aufgelöste Daten zuzugreifen (Abbildung 4). So stellt die Navigation typischerweise geringere Anforderungen an die Auflösung der Karte als beispielsweise die Steuerung des Greifarms bei der Erledigung einer Aufgabe.

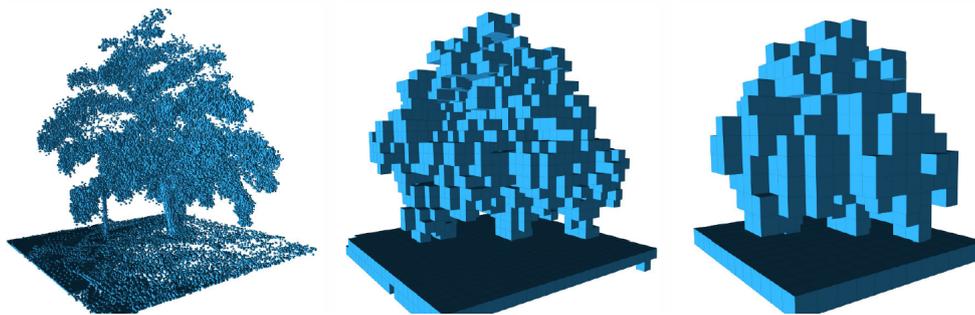


Abbildung 4: Ansichten der gleichen „OctoMap“ Karte in verschiedenen Auflösungen, (Wurm u. a., 2010)

2.3 Semantische Karten

Semantische Karten sind keine Alternative zur Darstellung des Weltmodells sondern vielmehr eine Anreicherung einer bestehenden Karte um semantische Informationen über Teile der Umgebung. Für einen autonomen Roboter ist es von Vorteil oder gar notwendig Teile seiner Umgebung zu erkennen und Wissen über die Eigenschaften und/oder Funktionen darüber zu besitzen. So kann ein entsprechend ausgestatteter Serviceroboter in die Lage versetzt werden

⁸Ein dreidimensionaler Binärbaum

geschlossene Türen selbständig zu öffnen, vorausgesetzt er erkennt sie als Tür und besitzt eine Wissensbasis, die Anweisungen darüber enthält, wie eine Tür zu öffnen ist.

Eine solche Wissensbasis für semantische Informationen muss immer zumindest eine Beschreibung des Objektes oder der Struktur enthalten. Das Kernproblem der semantischen Karten ist daher vor allem die zuverlässige Identifizierung von Objekten innerhalb der Sensor- oder Kartendaten. In (Nüchter und Hertzberg, 2008) wird hierfür ein Verfahren vorgestellt, welches in der Lage ist, in 3D-Punktwolken von Laserscans Umgebungsstrukturen und einzelne Objekte zu erkennen und zu klassifizieren. Dafür werden in der ersten Stufe großflächige, in einer Ebene liegende Bereiche gesucht und nach Eigenschaften wie Ausrichtung (horizontal oder vertikal), Höhe und Beziehung zu anderen Strukturen kategorisiert. Ergebnis ist die Kennzeichnung von Wänden, Böden, Decken und Türen innerhalb der Karte. Als zweite Stufe setzt darauf eine Objekterkennung auf, die aus 2D-Projektionen der 3D-Daten mit Hilfe klassischer Bildverarbeitungstechniken Konturen extrahiert und diese mit den Objekten der Wissensbasis vergleicht. Wird ein Treffer festgestellt, so wird der erkannte Bereich auf die 3D-Punkte zurücktransformiert und alle Punkte entsprechend ihrer Objektzugehörigkeit gekennzeichnet. Zusätzlich können alle Informationen über das Objekt in einer globalen Wissensbasis des Roboters abgelegt werden, so dass der Roboter schlussendlich in der Lage ist, auf Basis dieses Wissens zu agieren und Anweisungen auszuführen.

2.4 SLAM in nicht-statischen Umgebungen

Beinahe alle häuslichen Umgebungen sind ständigen, mehr oder weniger häufigen Änderungen unterworfen. Für uns Menschen ist dieser Fakt typischerweise kein größeres Problem, da wir intuitiv sehr gut zwischen sich ändernden Objekten und feststehenden Strukturen unterscheiden können. Einem Roboter jedoch, der nur ein begrenztes Wissen über die Eigenschaften von Objekten besitzt, kann es große Schwierigkeiten bereiten, sich in einer geänderten Umgebung zuverlässig zu lokalisieren. Um das Problem genauer einzugrenzen können Objekte in drei Dynamikklassen eingeteilt werden. (Zhou und Sakane, 2008)

- Dynamische Objekte sind solche, die ihre Position schnell genug ändern, so dass der Roboter diese Änderung während der Kartierung beobachten und sie direkt von der weiteren Bearbeitung ausschließen kann. Beispiele sind Menschen und Haustiere.
- Statische Objekte ändern ihren Standort während der gesamten Nutzungsdauer der Karte niemals. Typischerweise trifft dies auf Wände und andere gebäudetechnischen Struktur zu. Dies sind die Bestandteile der Karte, die eine zuverlässige Lokalisierung ermöglichen.
- Halbstatische Objekte sind die problematischen Fälle für SLAM-Algorithmen. Sie befinden sich oft relativ lange an einem festen Ort, so dass der Roboter sie bei der Kartierung

zunächst als statisch ansieht und ändern dann plötzlich, für den Roboter meist unbeobachtet, ihren Standort. Versucht der Roboter sich das nächste Mal an dieser Stelle zu lokalisieren, führt die veränderte Umgebung unter Umständen zu großen Fehlern und starken Abweichungen in der ermittelten Position.

Ein naiver Ansatz könnte sein, nach einem erfolgtem Mapping-Durchgang alle statischen Bereiche der Karte von einem Menschen markieren zu lassen und für die Lokalisierung zu nutzen. Da jedoch gerade in häuslichen Umfeldern ein Großteil aller Objekte in die Kategorie halbstatisch fällt und oft auch ganze Wände von solchen Objekten verdeckt werden können, gingen hierbei so viele Informationen verloren, dass eine zuverlässige und exakte Lokalisierung unmöglich werden könnte. Aufbauend auf diesem naiven Ansatz wird in (Meyer-Delius u. a., 2010) ein Verfahren vorgestellt, welches zusätzlich zur statischen Karte eine Reihe von temporären Karten erzeugt, sobald die Messwerte eine gewisse Grenze für die erlaubte Abweichung überschreiten. In dieser lokalen Umgebung werden dann solange die statische und temporäre Karte gemeinsam benutzt, bis sich die Messungen entweder wieder mit der statischen Karte allein decken oder sich die Umgebung erneut verändert und auch nicht mehr zur aktuellen temporären Karte passt. In diesem Fall wird eine neue temporäre Karte erstellt. Dies ist auch der größte Schwachpunkt des Verfahrens, da bereits kleine Änderungen, wie zum Beispiel ein stehengelassener Karton, dazu führen können, dass die komplette temporäre Karte mit allen halbstatischen Objekten darin neu erzeugt werden muss.

Einen besser geeigneten Ansatz zum (Roboter-)lebenslangen Lernen in Umgebungen mit vielen halb-statischen Objekten zeigen Biber und Duckett in (Biber und Duckett, 2005). Es wird hier zwischen einer aktuellen Karte die bei jeder Messung aktualisiert wird, dem „Kurzzeitgedächtnis“, und einer Reihe von Karten, die unterschiedliche Beobachtungszeiträume abdecken, dem „Langzeitgedächtnis“, unterschieden. Den Zeitraum den eine Karte dabei repräsentiert ergibt sich aus der Anzahl der gespeicherten Samples und der Updatehäufigkeit. Je weniger Samples und je häufiger die Updates, desto schneller passt sich die jeweilige Karte an Änderungen der Umgebung an. Dies führt dazu, dass zur Lokalisierung an einem beliebigen Punkt, auf Karten von verschiedenen Zeitpunkten zugegriffen und die am besten passende gewählt werden kann. Besonders an Punkten, an denen eine einzelne zuvor beobachtete Änderung wieder rückgängig gemacht wurde, kann mit einer älteren und präziseren (weil aus deutlich mehr Samples bestehenden) Karte, in der sich die kurzfristige Änderung noch nicht durchgesetzt hatte, gearbeitet werden.

3 Zusammenfassung

SLAM für statische Umgebungen gilt heutzutage als gut erforscht und ausreichend robust. Sobald jedoch dynamische Aspekte hinzukommen, die nicht nur als Messfehler interpretiert werden sollen, sondern in einer sinnvollen Art und Weise in das Weltbild eines Roboters integriert werden, stoßen die bisherigen Verfahren schnell an ihre Grenzen. Sehr ähnlich verhält es sich mit der Anreicherung der Umgebungsdaten um semantische Informationen. Hier betrifft es vor allem den Bereich der Objekterkennung, in dem es wünschenswert wäre, wenn ein Roboter irgendwann in der Lage ist neue Objekte selbständig zu segmentieren und dieses oder ähnliche Objekte zu einem späteren Zeitpunkt wiederzuerkennen. Im großen Bereich der Kartendarstellungen gibt es bereits heute eine Vielzahl verschiedenster Ansätze, Verfahren und Ideen für Optimierungen. Es gilt hier vor allem die Anforderungen genau zu definieren und die am besten passende Lösung als Basis für die eigene Entwicklung/Implementierung auszuwählen.

3.1 Ausblick

Als nächste Schritte soll ein möglichst genaues Anforderungsprofil erstellt und ein realistischer Rahmen für das Projekt abgesteckt werden. Hier muss auch mit dem Team, welches sich um die Kinematik und Steuerung des 5-Achs-Arms kümmert, abgestimmt werden, inwiefern eine integrierte 3D-Karte für alle Funktionsbereiche des Roboters sinnvoll ist. Auf Basis dieser Entscheidungen sollen bestehende SLAM-Ansätze näher untersucht und auf ihre Eignung hin überprüft werden. Hierbei muss insbesondere die Erweiterbarkeit für kommende Anwendungen im Bereich der semantischen Objekterkennung und dem Umgang mit nicht-statischen Umgebungen beachtet werden. Für die bestehende Plattform soll parallel dazu bereits eine einfache SLAM-Variante entwickelt und installiert werden, um erste praktische Erfahrung sammeln zu können. Da viele der skizzierten Ansätze im Bereich Kartierung und Semantik auf 3D-Daten basieren, muss außerdem die Integration entsprechender Sensorik in die Plattform vorangetrieben werden.

Angestrebt ist, dass bis zum Beginn des zweiten Projektteils die Basisplattform einsatzbereit ist und bereits einfache SLAM-Aufgaben ausgeführt werden können. Sollte die Zeit ausreichen, soll auch versucht werden, eine einfache autonome Navigation zwischen vorgegeben Punkten zu implementieren.

3.2 Risiken

Potentielle Schwierigkeiten in der Umsetzung der Vorhaben sind allem voran die bisher komplett fehlende, 3D-fähige Sensorik und ein eventuell noch nicht einsatzfähiges Steuerungs-

framework der Plattform. Während die Einschränkungen in der Sensorik zunächst noch verschmerzbar sind, da bereits eine Reihe von funktionsfähigen 2D-SLAM-Implementierungen auf Basis von Laserscannern verfügbar sind, könnte zuviel notwendige Entwicklungsarbeit am grundlegenden Framework der Plattform den Zeitplan stark beeinträchtigen. Auch würde dies vermutlich zunächst wieder einiges an Einarbeitung und eine Verschiebung des Fokus erforderlich machen. Hier müsste gegebenenfalls der Einsatz eines offenen, modularen und erweiterbaren Roboterframeworks wie beispielsweise *ROS*⁹ oder des herstellereigenen Frameworks *CogniDrive*¹⁰ geprüft werden.

⁹Robot Operating System, <http://www.ros.org/>

¹⁰http://metralabs.com/index.php?option=com_content&view=article&id=66&Itemid=68

Literatur

- [Biber und Duckett 2005] BIBER, Peter ; DUCKETT, Tom: Dynamic Maps for Long-Term Operation of Mobile Service Robots. In: *Robotics: Science and Systems*, 2005, S. 17–24
- [Durrant-Whyte und Bailey 2006] DURRANT-WHYTE, H. ; BAILEY, T.: Simultaneous localization and mapping: part I. In: *Robotics Automation Magazine, IEEE* 13 (2006), june, Nr. 2, S. 99–110. – ISSN 1070-9932
- [Frese u. a. 2010] FRESE, Udo ; WAGNER, René ; RÖFER, Thomas: A SLAM Overview from a User's Perspective. In: *KI - Künstliche Intelligenz* 24 (2010), S. 191–198. – URL <http://dx.doi.org/10.1007/s13218-010-0040-4>. – 10.1007/s13218-010-0040-4. – ISSN 0933-1875
- [Hertzberg u. a. 2010] HERTZBERG, Joachim ; LINGEMANN, Kai ; LÖRKEN, Christopher ; NÜCHTER, Andreas ; STIENE, Stefan ; WIEMANN, Thomas: 3D-Roboterkartenbau in Osnabrück. In: *KI - Künstliche Intelligenz* 24 (2010), S. 245–248. – URL <http://dx.doi.org/10.1007/s13218-010-0032-4>. – 10.1007/s13218-010-0032-4. – ISSN 0933-1875
- [Meyer-Delius u. a. 2010] MEYER-DELIUS, D. ; HESS, J. ; GRISETTI, G. ; BURGARD, W.: Temporary maps for robust localization in semi-static environments. In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, S. 5750–5755. – ISSN 2153-0858
- [Montemerlo u. a. 2002] MONTEMERLO, M. ; THRUN, S. ; KOLLER, D. ; WEGBREIT, B.: Fast-SLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In: *Proceedings of the AAAI National Conference on Artificial Intelligence*. Edmonton, Canada : AAAI, 2002
- [Nüchter und Hertzberg 2008] NÜCHTER, Andreas ; HERTZBERG, Joachim: Towards semantic maps for mobile robots. In: *Robot. Auton. Syst.* 56 (2008), November, S. 915–926. – URL <http://portal.acm.org/citation.cfm?id=1453261.1453481>. – ISSN 0921-8890
- [Riaz u. a. 2010] RIAZ, Z. ; PERVEZ, A. ; AHMER, M. ; IQBAL, J.: A fully autonomous indoor mobile robot using SLAM. In: *Information and Emerging Technologies (ICIET), 2010 International Conference on*, june 2010, S. 1–6
- [Thrun u. a. 2000] THRUN, S. ; BEETZ, M. ; BENNEWITZ, M. ; BURGARD, W. ; CREMERS, A. B. ; DELLAERT, F. ; FOX, D. ; HÄHNEL, D. ; ROSENBERG, C. ; ROY, N. ; SCHULTE, J. ; SCHULZ, D.: Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva. In: *The International Journal of Robotics Research* 19 (2000), Nr. 11, S. 972–999. – URL <http://ijr.sagepub.com/content/19/11/972.abstract>

- [Thrun und Leonard 2008] THRUN, Sebastian ; LEONARD, John J.: Simultaneous Localization and Mapping. In: SICILIANO, Bruno (Hrsg.) ; KHATIB, Oussama (Hrsg.): *Springer Handbook of Robotics*. Springer Berlin Heidelberg, 2008, S. 871–889. – URL http://dx.doi.org/10.1007/978-3-540-30301-5_38. – 10.1007/978-3-540-30301-5_38. – ISBN 978-3-540-30301-5
- [Wurm u.a. 2010] WURM, K.M. ; HORNING, 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*. Anchorage, USA, Mai 2010. – URL <http://ais.informatik.uni-freiburg.de/publications/papers/wurm10octomap.pdf>
- [Zhou und Sakane 2008] ZHOU, Hongjun ; SAKANE, S.: Localizing objects during robot SLAM in semi-dynamic environments. In: *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*, 2008, S. 595–601