
Teach-In für die 3D-Scan Akquise mit einem Roboter

Oliver STRUCKMEIER, Dorit BORRMANN und Andreas NÜCHTER

Zusammenfassung

Die 3D-Erfassung einer kompletten Umgebung mittels 3D-Laserscanner stellt abhängig von der zu scannenden Umgebung und dem Scanner einen hohen Zeitaufwand für einen menschlichen Operator dar. Der Scanner muss an die einzelnen Scanpositionen bewegt und dort verortet werden. Mit steigender Qualität der Messung nimmt zudem die Dauer der einzelnen Scanvorgänge zu. In diesem Beitrag wird ein Teach-In Ansatz vorgestellt und evaluiert, der den manuellen Vorgang verkürzt. Dabei führt ein Roboter die vom Vermesser geplanten und eingespeicherten zeitintensiven Schritte automatisch durch und entlastet somit den Bediener.

1 Einleitung

Die Entwicklung von Sensoren und Aktuatoren in den letzten Jahren ermöglicht es Robotern immer besser mit ihrer Umwelt zu interagieren, insbesondere ihre Umgebung genauer zu erfassen, präzisere Karten zu erstellen und sich in diesen zu lokalisieren. Die Erfassung einer Umgebung mit einem 3D-Laserscanner nimmt viel Zeit in Anspruch. Dabei müssen zuerst die Scanpositionen geplant werden, der Scanner an jede einzelne von ihnen positioniert werden und die mehrminütige Aufnahme pro Position durchgeführt werden. Während der Aufnahmezeit muss der Vermesser außerhalb des Sichtfeldes des Scanners abwarten.

In diesem Beitrag präsentieren wir eine Optimierung des Vorganges zur Erstellung eines 3D-Modelles einer Umgebung mit Hilfe eines mobilen Roboters mit 2D- und 3D-Laserscanner. Mittels Teach-In wird einem Roboter ein Pfad, der mehrere Scanpositionen verbindet, eingespeichert. Während des Teach-In Vorgangs fertigt der Roboter außerdem mit Simultaneous Localization and Mapping (SLAM) eine 2D-Karte der Umgebung zu seiner eigenen Orientierung an. Anschließend ist er in der Lage selbständig die abgespeicherten Wegpunkte abzufahren und alle Scans durchzuführen. Nach Abschluss werden die erzeugten Rohdaten mit dem 3D Toolkit (3DTK – <http://www.threedtk.de>) registriert und aufbereitet.

Auf diese Weise ist der Vermesser an der Planung der Scanpositionen beteiligt, die langwierige Aufnahme übernimmt der Roboter aber selbständig. Der Bediener kann sich langfristig aus dem Aufnahmegebiet des Laserscanners entfernen und sich anderen Aufgaben zuwenden. In diesem Beitrag wird das Verfahren vorgestellt und anhand von Testszenarien evaluiert.

2 Systemkomponenten

2.1 Hardware

Der Roboter Irma3D (Abk.: Intelligent Robot for Mapping Applications in 3D, Abb. 1) basiert auf dem Volksbot Konstruktions Kit des Fraunhofer Institutes für Intelligente Analyse- und Informationssysteme (IAIS) (FRAUNHOFER IAIS 2017), das ein flexibles und modulares System darstellt, das mit einer Vielzahl von Sensoren und Aktuatoren konfigurierbar ist.

1. Riegl VZ-400 3D-Laserscanner
2. Ladevorrichtung der Batterien des 3D-Scanners
3. Anschlüsse des 3D-Scanners
4. SICK LMS100 2D-Laserscanner
5. LAN-Kabel zum Notebook
6. LAN Switch für Anbindung der Laserscanner an den Computer
7. USB Anschluss Notebook an USB Hub
8. USB Hub für Anschluss des Notebooks
9. Notaus-Schalter
10. ON/OFF Schalter
11. ON/OFF Kontrollschalter

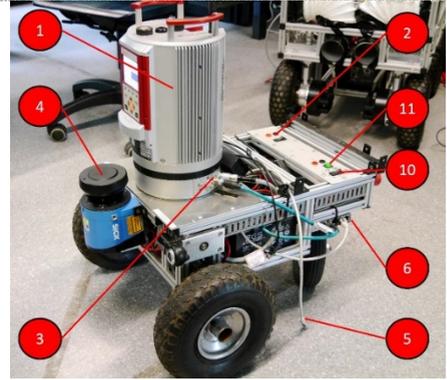


Abb. 1: Aufbau des Roboters Irma3D

Irma3D bewegt sich auf zwei aktiven Vorderrädern und einer passiven Lenkrolle vorwärts. Die beiden aktiven Räder werden von 150 Watt Maxon Motoren angetrieben. Zur Stromversorgung dienen vier 12 Volt Bleigel-Akkus mit 9 Amperestunden. In der Standardausführung verfügt der Roboter über zwei Akkus. Die Zusatzakkus bilden einen zweiten elektrischen Kreislauf, mit dem der 3D-Laserscanner versorgt wird.

Für die Aufnahme der 3D-Laserdaten wird der RIEGL VZ-400 verwendet. Die Scanzeit bei einer Auflösung von $0,04^\circ$ und dem maximalen Öffnungswinkel von 360° in horizontaler und 100° in vertikaler Richtung im High-Speed Modus bei einer Reichweite von 350 m beträgt ca. 3 Minuten.

Der an der Frontseite des Roboters angebrachte LMS100 2D-Laserscanner (4) ist ein für Indoor Anwendungen konzipierter Infrarots Scanner der Firma SICK. Der Laserscanner liefert mit einer Rate von 50 Hertz Entfernungsmessungen von Objekten in der Reichweite von 0,5 bis 20 Meter um den Roboter und hat einen Öffnungswinkel von 270° in Fahrtrichtung des Roboters. Die 2D-Laserdaten werden im Teach-In Prozess zum Kartieren der Umgebung benutzt. Später dienen die 2D-Laserdaten zur automatischen Lokalisierung in der Karte.

2.2 Kartierung und Lokalisierung

2.2.1 Occupancy Grid Maps

Ein Occupancy Grid (engl. Belegtheitsgitter) ist eine Darstellung der Umgebung als 2D-Karte zur Kartierung mit unpräzisen Sensordaten, wie 2D-Laserscannern (THRUN, BURGARD, & FOX 2005, S. 284). Die Karte wird dabei in kleine Quadrate unterteilt, von denen jedes Feld mit einem Wahrscheinlichkeitswert versehen ist, der aussagt, ob das Feld belegt oder frei ist. Abb. 2 stellt ein vereinfachtes Occupancy Grid dar. Das Sichtfeld des Roboters ist dargestellt durch die Linien, ausgehend von der Roboterposition auf einem Feld. Die weißen Felder markieren das freie Sichtfeld des Laserscanners, die schwarzen stellen ein Hindernis dar. Grau markierte Felder werden nicht erfasst.

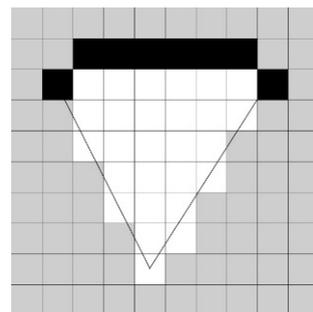


Abb. 2: Occupancy Grid

2.2.2 Kartierung mit TSD-SLAM

SLAM beschreibt in der Robotik das Problem der 2D-Kartierung mit zeitgleicher Lokalisierung des Roboters in der Karte. Dabei agiert ein Roboter in einer unbekanntem Umgebung und verfügt über Sensoren zur Umgebungserfassung, z.B. den in dieser Arbeit verwendeten 2D-Laserscanner. Die Ansätze zur Lösung des SLAM-Problems lassen sich in Online- und Offlinemethoden unterteilen. Onlinemethoden berechnen während der Roboterfahrt die Karte und lokalisieren den Roboter darin. Hierbei summieren sich lokale Fehler auf, so dass eine inkonsistente Karte entstehen kann. Offlinemethoden optimieren nach der Roboterfahrt die komplette Trajektorie, wodurch die optimierten Daten erst später zur Verfügung stehen.

In dieser Arbeit wird mit der TSD-SLAM Methode ein Onlineverfahren verwendet (MAY, KOCH, KOCH, PFITZNER, & NÜCHTER 2014). Das Lokalisierungsmodul verwendet den ICP (Iterative Closest Point) Algorithmus (BESL & MCKAY 1992). Dieser iterative Algorithmus bestimmt zu jedem Punkt aus der einen Punktwolke den nächstgelegenen Punkt aus der anderen Punktwolke und berechnet eine Transformation, die alle Abstände der Punktpaare minimiert. Diese Schritte wiederholen sich bis zur Konvergenz. Auf diese Weise wird die Roboterposition innerhalb einer teilweise bekannten Karte bestimmt. Mit dem Wissen über die Positionsänderung werden nun neue Lasermessungen in die Karte eingezeichnet. Während der Sichtbereich des Laserscanners über die Zellen des Occupancy Grids wandert, aktualisiert TSD-SLAM die Zustände abhängig von den Laserscans.

2.2.3 Lokalisierung mit AMCL

AMCL steht für Adaptive Monte Carlo Localization und ist ein Lokalisierungsalgorithmus für Roboter in 2D-Karten mit Hilfe eines Partikelfilters (GERKEY 2017). Monte Carlo Algorithmen sind randomisierte Algorithmen, die dann verwendet werden, wenn die exakte Berechnung eines Problems aufwändig ist und eine Lösung durch Stichproben effizienter erreicht werden kann (NÜCHTER, HERTZBERG, & LINGEMANN 2012). AMCL bestimmt mit Hilfe von 2D-Laserdaten eine beliebige Position des Roboters in der Karte. Zur Initialisierung werden zunächst Partikel gleichmäßig über die gesamte Karte verteilt. Jedem Partikel ist ein Gewicht zugewiesen, das die Wahrscheinlichkeit beschreibt, mit der sich der Roboter an dessen Position befindet. Die Partikelverteilung, der Belief X_{t-1} , aktualisiert sich an jedem Zeitpunkt t mit den Odometriedaten u_t (Bewegungsschätzung durch Radumdrehungen) des Roboters und den 2D-Laser Sensordaten z_t auf den neuen Zustand X_t . Anschließend fügt AMCL erneut Partikel in die Karte ein, mit höherer Wahrscheinlichkeit in der Nähe von Partikeln mit größerem Gewicht. Bei einer Bewegung des Roboters bewegen sich die Partikel in die entsprechende Richtung und werden anschließend unter Betrachtung der neuen Laserdaten neu gewichtet. Somit konzentrieren sich die Partikel über mehrere Durchläufe um die tatsächliche Position. Die geschätzte Position des Roboters liegt bei dem Partikel mit dem höchsten Gewicht. Durch die Aufrechterhaltung mehrerer Schätzungen in den Partikeln ist der Algorithmus robust gegenüber Sensorfehlern.

2.2.4 Der Regler zur Pfadverfolgung

Die Pfadverfolgung geschieht mittels des Reglers aus (INDIVERI & CORRADINI 2004), der auf der Arbeit von (DE WIT, KHENNOUF, SAMSON & SORDALEN 1993) basiert. Dabei wird der darin beschriebene nichtlineare Controller um eine Funktion erweitert, die den minimalen Wendekreis sowie die maximale Geschwindigkeit des Roboters berücksichtigt. Der nichtlineare Pfad wird durch Näherung in lineare Teilstücke zerlegt.

Abb. 3 skizziert die Pfadverfolgung. θ beschreibt den Winkel zwischen der x -Achse und der Längsachse des Roboters, θ_d den Winkel einer Tangente an den Pfad zur x -Achse. Aus $\tilde{\theta} = \theta - \theta_d$ ergibt sich der Winkel zwischen der Fahrtrichtung des Roboters und der Tangente an einen Punkt p auf dem Pfad. l ist der Abstand zwischen dem Rotationszentrum des Roboters und seiner orthogonalen Projektion p auf den Pfad.

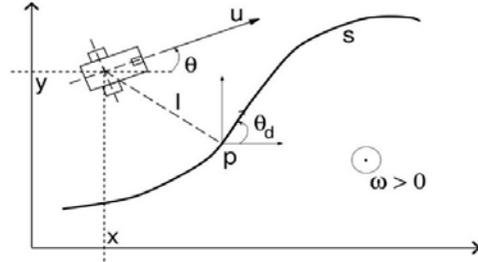


Abb. 3: Pfadverfolgung (INDIVERI & CORRADINI 2004)

$$\omega = \frac{uK(s)\cos\theta}{1-lK(s)} - hyl\frac{\sin\tilde{\theta}}{\tilde{\theta}} - \gamma\tilde{\theta} \quad : \quad h, y > 0 \quad (1)$$

berechnet die Winkelgeschwindigkeit ω mit u als linearer Geschwindigkeit des Roboters. γ und h sind Parameter des Reglers. $K(s)$ stellt die Krümmung der Kurve dar. Da der Algorithmus den Pfad auf lineare Teilstücke nähert wird $K(s) = 0 \forall s$ angenommen und der Abstand l des Roboters zum Pfad wird mit y ersetzt, da die x -Achse als globale Referenz dient:

$$\omega = -huy\frac{\sin\tilde{\theta}}{\tilde{\theta}} - \gamma\tilde{\theta} \quad : \quad h, y > 0. \quad (2)$$

2.3 Softwareumgebung

Das Robot Operating System (ROS - <http://www.ros.org>) ist ein Open-Source Softwareframework für die Programmierung von Robotern, das betriebssystem-ähnliche Funktionen zur Verfügung stellt. ROS verfügt über Tools und Bibliotheken zur einfachen Einbindung von Roboterkomponenten. Einzelne Module, sogenannte Nodes, kommunizieren nach dem Publisher/Subscriber Prinzip über die ROS Middleware miteinander. Neben der Nodes für die Ansteuerung der Hardware, also des Roboters und der Laserscanner, wurden für die Automatisierung der Pfadverfolgung mehrere Nodes zur Steuerung des Roboters und des 3D-Scanners im manuellen und automatischen Modus entwickelt. Die Pfaderzeugung und das Abarbeiten der Pfaddateien sowie die Verwendung des Reglers für die Aufgabe inklusive Hindernisvermeidung wurden implementiert.

Zur Verarbeitung der aufgenommenen 3D-Daten kommt das 3D Toolkit (3DTK) zum Einsatz. Ausgehend von den Poseschätzungen des Roboters berechnen der ICP Algorithmus und die global konsistente Optimierung von NÜCHTER, LINGEMANN, BORRMANN, ELSEBERG & BÖHM (2008) die optimale Pose der Scans im lokalen Koordinatensystem.

3 3D-Scan Akquise mit einem mobilen Roboter und Teach-In

Der Kartierungsprozess besteht aus dem manuellen Teach-In Vorgang und der anschließenden automatischen Scanakquise.

Teach-In: Der Roboter beginnt an einem Startpunkt, der das Projektkoordinatensystem definiert. Von dort steuert ein Benutzer per Joystick die einzelnen Scanpositionen an. Per Knopfdruck markiert er die Wegpunkte für die 3D-Scans. Während dieses Vorganges erstellt

der Roboter aus den Daten des 2D-Laserscanners mittels TSD-SLAM die Karte und zeichnet seinen zurückgelegten Pfad in einer Datei auf. Mit setzen eines Wegpunktes wird die Pfadaufzeichnung in einer neuen Datei fortgesetzt. Somit entstehen im Verlauf des Kartierungsprozesses mehrere aufsteigend nummerierte Dateien mit Pfadinformationen von Wegpunkt zu Wegpunkt. Abb. 4 zeigt das Ergebnis eines Teach-In Prozesses im Untergeschoss des Institutes für Informatik der Universität Würzburg. Zu sehen ist die angefertigte Karte des Gebäudes mit eingezeichnetem Pfad des Roboters in rot und Ausgangspunkt in schwarz.

Automatische Scan-Akquise: Mit der erzeugten Karte und den Pfaden zwischen den Wegpunkten beginnt das automatische Abfahren des Pfades und die Scan-Akquise. Mittels AMCL lokalisiert sich der Roboter in der Karte und liest den ersten Pfadabschnitt ein. Nach der Verfolgung eines Abschnittes mit dem Regler aus Abschnitt 2.2.4 hält der Roboter an und führt einen 3D-Scan durch. Anschließend folgt das Einlesen des nächsten Pfadabschnitts, bis alle Abschnitte abgearbeitet wurden.

4 Auswertung

Das Verfahren wurde in mehreren Szenarien getestet. Es folgt einer Auswertung anhand von Experimenten im Untergeschoss des Informatikgebäudes der Universität Würzburg. Zur Evaluation wurden sowohl während des Teach-Ins als auch bei der automatischen Pfadverfolgung Scans an den 7 Wegpunkten aufgenommen und mit 3DTK registriert.

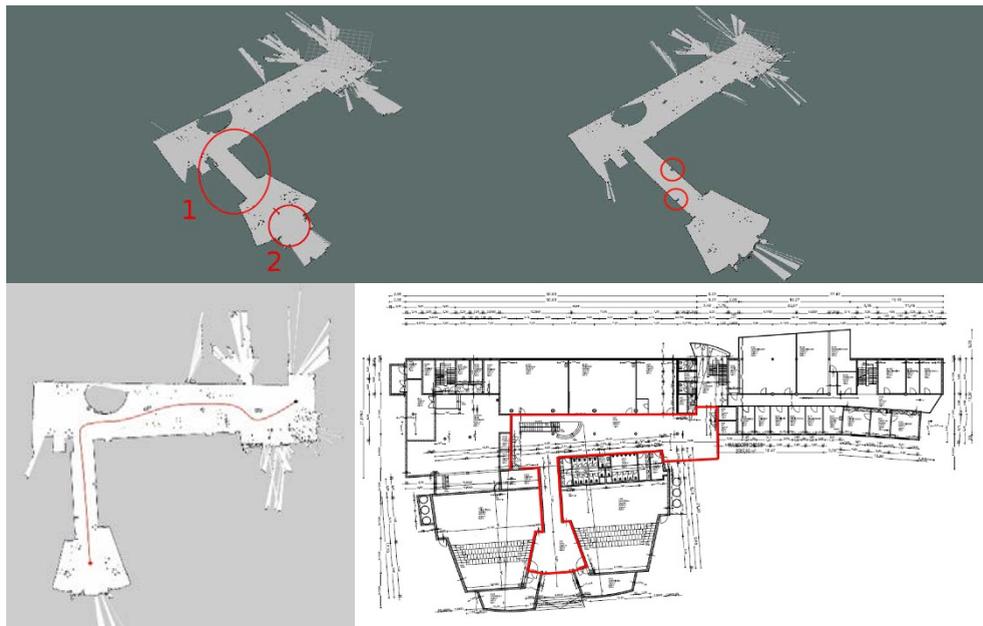


Abb. 4: Oben: Lokalisierungs- und Mappingfehler im Gang (links) behoben durch zusätzliche Objekte im Gang (rechts). Unten: Vergleich der 2D-Karte mit skizziertem Pfad mit dem offiziellen Grundriss des Untergeschosses des Informatikgebäudes der Universität Würzburg.



Abb. 5: Links: Die registrierten 3D-Scans aus Teach-In (blau) und Pfadverfolgung als Draufsicht (grün), eingefärbt über einen Höhenfarbverlauf und ab einer Höhe kurz unter der Decke abgeschnitten. Rechts: Die Scanpositionen in der registrierten finalen 3D-Punktwolke überlagert mit dem offiziellen Grundriss.

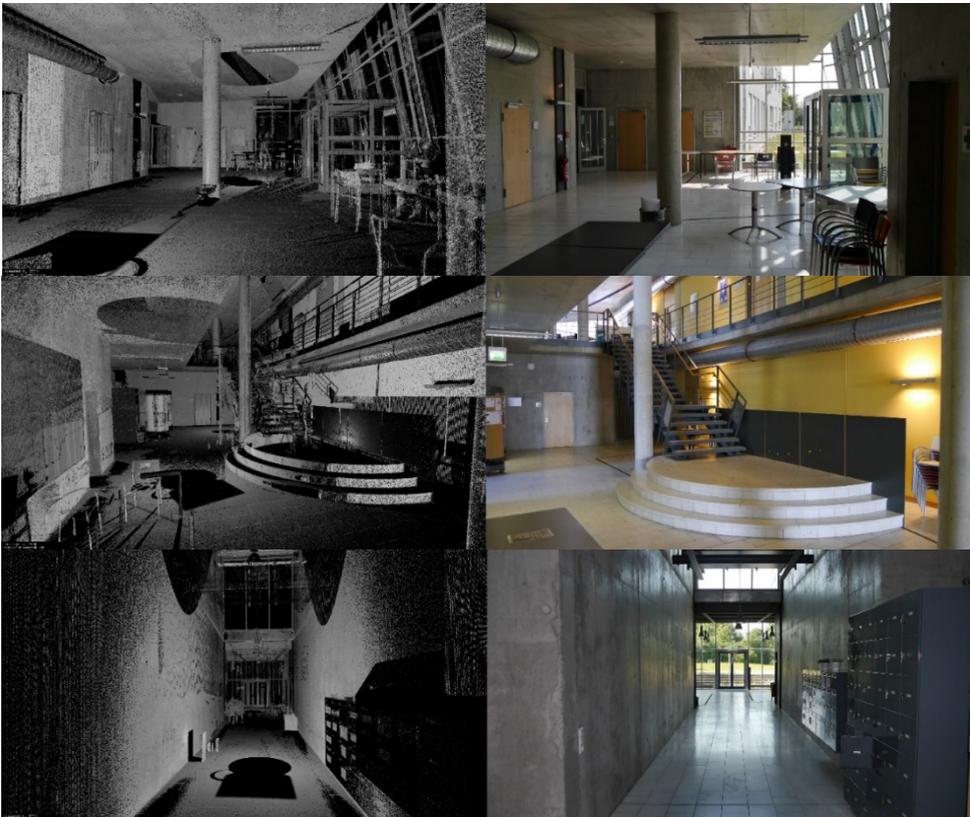


Abb. 6: Vergleich des registrierten 3D-Modells mit Fotos bei Scanposition 2, 3 und 5, eingefärbt mittels der Intensitätswerte. Der Hintergrund ist schwarz. Man erkennt deutlich die Glasfront am Ende des langen Ganges (unten) und die Scanpositionen.

4.1 Die 2D-Karte

Eine Betrachtung der erzeugten 2D-Karte offenbart Schwächen bei der Kartierung in Umgebungen ohne markante Umgebungsdetails. Abb. 4 zeigt den verkürzten Gang (1) in der Karte, da die Lokalisierung während der Kartierung einzig anhand der 2D-Laserscanpunkte der Umgebung erfolgt. Beim Abfahren des Ganges (1) erfasst der Laserscanner nur die Wände, die aufgrund der Enge nur durch wenige Punkte abgebildet werden. Die Glasscheiben in Fahrtrichtung (2) werden nicht erfasst. Somit ändern sich während der Fahrt die Laserdaten kaum, so dass der Kartierungsalgorithmus in seiner internen Karte fälschlicherweise keine Bewegung des Roboters verzeichnet. Der Gang wird dadurch verkürzt erfasst. Eine Lösung bietet das Aufstellen von Objekten im Gang (Abb. 4, rechts) als zusätzliche Umgebungsmerkmale. Durch diese Modifikation wird die Geometrie korrekt erfasst, wie ein Vergleich der Karte von TSD-SLAM mit dem offiziellen Grundriss der zu scannenden Umgebung verdeutlicht.

4.2 Bewertung des erzeugten 3D-Modells

Aus den einzelnen 3D-Scans wird mithilfe des 3D Toolkits ein zusammenhängendes 3D-Modell erzeugt, indem die Punktwolken unter Beachtung der Aufnahmeposition relativ zueinander rotiert und verschoben werden. Abb. 5 zeigt, dass sich die Scans von beiden Durchläufen gut registrieren lassen. Ein Vergleich mit dem Grundriss veranschaulicht auch hier die korrekte Geometrie. Eine Gegenüberstellung des 3D-Modells mit Fotos aus dem Informatikgebäude zeigt den Detailgrad des Modells (Abb. 6). Trotz der vielen Glasfassaden, sind im Modell keine Registrierungsfehler ersichtlich.

4.3 Auswertung des Roboterpfades

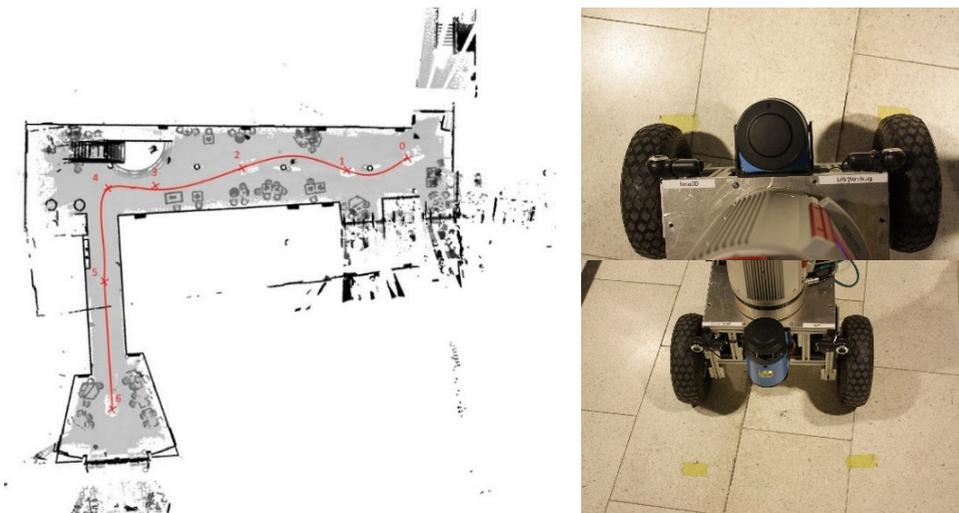


Abb. 7: Links: Pfad mit den Scanpositionen in einer Draufsicht (ohne Decke) der registrierten 3D-Scans. Die Scans sind über einen Farbverlauf entsprechend der Höheninformationen eingefärbt. Rechts: Abweichung (unten) von Sollposition (oben) am Ende der Pfadverfolgung.

Zur Erstellung des 3D-Modells legt der Benutzer Wegpunkte für 3D-Scans fest, die der Roboter im Pfadverfolgungsvorgang selbständig abfährt. Abb. 7 illustriert die verwendeten Wegpunkte aus dem Beispiel. Sie wurden ausgewählt mit dem Ziel verschiedene Blickwinkel für ein lückenloses 3D-Modell abzudecken, aber die Anzahl der Scans gering zu halten. Der Roboter fährt in zahlreichen Experimenten die Wegpunkte zuverlässig ab, jedoch zeigen sich an den einzelnen Wegpunkten kleine Fehler, wie in Abb. 7 beispielhaft dargestellt.

4.3.1 Translations- und Rotationsfehler der Scanpositionen

Für eine qualitative Bewertung der Fehler des Roboters an den einzelnen Wegpunkten nimmt der Roboter in einem Experiment bereits während des Teach-Ins Laserscans auf. Diese werden mit 3DTK registriert um die tatsächlichen Scanpositionen zu erhalten. Ebenso werden die Scans des Pfadverfolgungsdurchlaufs untereinander registriert, sowie zusätzlich im Koordinatensystem des Teach-In-Vorgangs verortet. Der Koordinatenursprung für die Referenz liegt im Wegpunkt 0 des Teach-In-Vorgangs. Für jeden Wegpunkt stehen somit vier Posen zur Verfügung. T0 bezeichnet die durch TSD-SLAM geschätzte Pose des Roboters während des Teach-Ins, T1 die tatsächliche Pose des Roboters während des Teach-Ins, die durch Registrierung mittels 3DTK bestimmt wurde. P0 ist die Pose aus AMCL während der Pfadverfolgung und schließlich P1 die tatsächliche Pose der aufgenommenen Scans.

Abb. 8 stellt die Translations- und Rotationsdifferenzen zwischen den einzelnen Schätzungen dar. Die Rotationsfehler berechnen sich durch Näherung der Distanz zweier Einheitsquaternionen (KUFFNER 2004). Betrachtet man den Gesamtfehler, so zeigt sich, dass die Position maximal 20 cm abweicht und der Rotationsfehler der mittleren Scanpositionen maximal 2° beträgt, zu Beginn und zum Ende jedoch höher liegt. Insgesamt sind diese Abweichungen tolerierbar. Dennoch sollen die Fehlerquellen genauer analysiert werden.

Der Fehler des Reglers liegt konstant bei ca. 25 cm und auch die hohen Rotationsfehler sind ausschließlich auf den Regler zurückzuführen. Die positionale Abweichung ist durch die Unterteilung des Pfades in lineare Teilstücke begründet. Sobald die Distanz - unabhängig von der Rotation - zum Ende des Teilstücks einen Schwellwert unterschreitet, wird das nächste Teilstück des Pfades verwendet. Abhilfe bietet hier eine strengere Grenze für den Wegpunkt. Die Differenz der Rotation ist aufgrund des 360° Sichtfeldes des Laserscanners ohne Bedeutung, ließe sich aber ebenfalls mit einer Sonderbehandlung des Wegpunktes korrigieren.

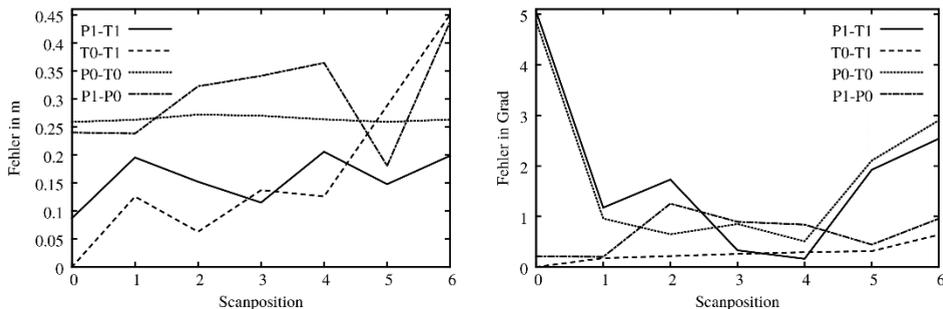


Abb. 8: Translationsfehler (links) und Rotationsfehler (rechts). Geplottet sind die Fehler der einzelnen Module, der Fehler des TSD-SLAM (T0-T1), des Regler (P0-T0), von AMCL (P1-P0), sowie der Gesamtfehler (P1-T1).

Durch Aufsummieren lokaler Fehler steigt der Translationsfehler während der Kartierung mit zunehmender Pfadlänge. Durch die Diskretisierung der Karte sind kleine Ungenauigkeiten unvermeidbar. Insbesondere bei den letzten Scanpositionen, wo der Roboter einen langen schmalen Gang durchfährt und schließlich vor einer Glasfassade stehen bleibt, ist die Orientierung mittels der 2D-Laserscandaten schwierig. Derselbe Effekt tritt auch bei der Lokalisierung mittels AMCL während der Pfadverfolgung auf. Die Positionsschätzung des Roboters weicht stark von der tatsächlichen Position ab. Die Lokalisierungsgenauigkeit ist hier jedoch auch von der Qualität der Karte abhängig. Dieser Effekt wirkt sich aber nur bedingt auf das Endergebnis aus. Insbesondere beim letzten Scanpunkt erkennt man, dass die Schätzungen aus der Kartierung und Lokalisierung beide identisch sind, die tatsächliche Abweichung der Scanposition zur Position des Teach-Ins jedoch nur bei 20 cm liegt. Eine Verbesserung ließe sich durch Berücksichtigung der Odometriemessungen bei der Kartierung erzielen. Die Rotationsschätzung ist sowohl bei TSD-SLAM als auch bei AMCL sehr gut.

4.4 Praxistest auf den Oldenburger 3D-Tagen 2017

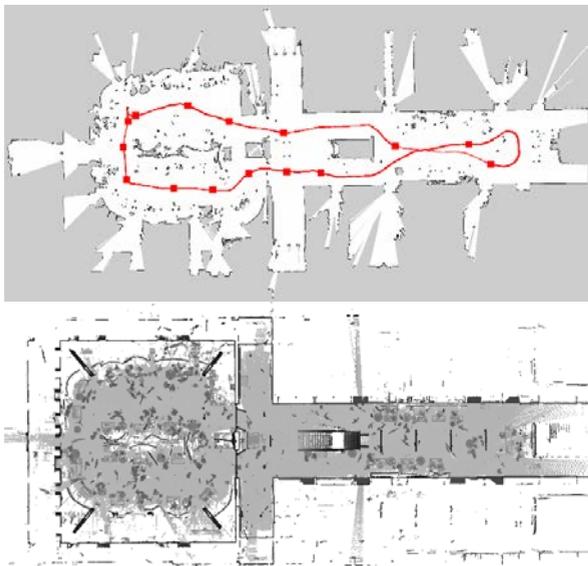


Abb. 9: Karte mit Pfad und Scanpositionen (oben) sowie Draufsicht der 3D-Scans von der Firmenausstellung auf den Oldenburger 3D-Tagen. Einfärbung der Scans erfolgt mit einem Höhenverlauf.

Auf den Oldenburger 3D-Tagen wurde das Verfahren einem Praxistest in dynamischen Umgebungen unterzogen. Zu Beginn der Veranstaltung wurden mehrere Pfade aufgenommen. Abb. 9 zeigt einen Pfad durch die gesamte Firmenausstellung mit 15 Scanpositionen. In den 3D-Daten erkennt man deutlich einige der Veranstaltungsteilnehmer, die in der Karte nicht erscheinen. Trotz der zahlreichen Veränderungen durch die Besucher der Firmenausstellung schaffte Irma3D es im Laufe der Veranstaltung immer wieder zuverlässig die vorgegebenen Pfade abzufahren. Sobald sich ein Hindernis in den Weg stellte, wartete der Roboter ab, bis der Pfad wieder frei war, um diesen fortzusetzen. Dieser Test demonstriert die Einsetzbarkeit des Systems in dynamischen Umgebungen.

5 Zusammenfassung und Ausblick

Der vorgestellte Vorgang verkürzt durch Teach-In die Anwesenheit eines Operators bei der zeitintensive 3D-Scan Akquise. Mit einem mobilen Roboter werden mittels 2D-SLAM eine 2D-Karte zur Orientierung sowie Wegpunkte und ein Pfad zwischen einzelnen Wegpunkten eingespeichert. In einem zweiten automatischen Vorgang fährt der Roboter mit den erzeugten Daten und einem Reglers zur Pfadverfolgung den Pfad ab und nimmt die 3D-Kartierung vor.

Die zuvor geplanten Scanpositionen werden präzise angefahren. Durch die zuverlässige Lokalisierung des Roboters ist eine automatische Verortung der 3D-Daten in einem Projektkoordinatensystem mit 3DTK möglich. Dabei entstehen detailgetreue vollständige 3D-Modelle. Die erzeugte 2D-Karte kann auch für andere Anwendungen wiederverwendet werden oder durch einen Grundriss ersetzt werden.

Zu beachten bleibt die Lokalisierungsschwierigkeit in Umgebungen ohne markante Punkte mittels TSD-SLAM, was durch Hinzufügen von zusätzlichen Orientierungspunkten umgangen werden kann. Softwareseitig ließe sich dieses Problem durch zusätzliches Abgleichen mit den Odometriedaten des Roboters beheben. Keine Probleme bereitet hingegen eine stark dynamische Umgebung, wie die Firmenausstellung auf der Oldenburger 3D-Tagen mit ihren Besuchern. Dies zeigt, dass die Pfaderstellung auch in belebten Umgebungen möglich ist. Als Erweiterung des Ansatzes ist geplant, das erstellte 3D-Modell auf Verschattungen zu überprüfen und automatisch zusätzliche Scanpunkte zu berechnen und anzufahren, die benötigt werden, um eine vollständige Erfassung der Umgebung zu ermöglichen.

6 Danksagung

Die Autoren danken Philipp Koch und Stefan May von der Technischen Hochschule Nürnberg für die Unterstützung bei der Verwendung der TSD-SLAM Methode.

Literatur

- Besl, P. J., & McKay, N. D. (1992). *A Method for Registration of 3-D Shapes*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2), S. 239-256.
- de Wit, C., Khennouf, H., Samson, C. & Sordalen, O. (1993). *Nonlinear Control Design for Mobile Robots*. Recent Trends in Mobile Robots, S. 121-156.
- Fraunhofer IAIS. (2017). *VolksBot*. Von <http://www.volksbot.de/index-en.php> abgerufen
- Gerkey, B. (2017). *ROS Wiki - AMCL*. Von <http://wiki.ros.org/amcl> abgerufen
- Indiveri, G., & Corradini, M. L. (2004). *Switching Linear Path Following For Bounded Curvature Car-like Vehicles*. Proc. Symp. on Intelligent Autonomous Vehicles.
- Kuffner, J. J. (2004). *Effective sampling and distance metrics for 3D rigid body path planning*. Proc. Intern. Conf. on Robotics and Automation (ICRA), S. 3993-3998).
- May, S., Koch, P., Koch, R., Pfitzner, C., & Nüchter, A. (2014). *A Generalized 2D and 3D Multi-Sensor Data Integration Approach based on Signed Distance Functions for Multi-Modal Robotic Mapping*. Proc. VMV: Vision, Modeling & Visualization.
- Nüchter, A., Hertzberg, J., & Lingemann, K. (2012). *Mobile Roboter*. Springer Verlag.
- Nüchter, A., Lingemann, K., Borrmann, D., Elseberg, J., & Böhm, J. (2008). *Global Konsistente 3D-Kartierung mit Scanmatching*. Photogrammetrie, Laserscanning, Optische 3D-Messtechnik, Oldenburger 3D-Tage, S. 194-201.
- Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics*. MIT Press.

Bitte geben Sie auf einer separaten Seite zu allen genannten Autoren folgende Informationen:

Struckmeier, Oliver
Julius-Maximilians-Universität Würzburg
oliver.struckmeier@stud-mail.uni-wuerzburg.de

Borrmann, Dorit
Julius-Maximilians-Universität Würzburg
borrmann@informatik.uni-wuerzburg.de

Nüchter, Andreas
Julius-Maximilians-Universität Würzburg
nuechter@informatik.uni-wuerzburg.de

Diese Angaben werden im Autorenverzeichnis am Ende des Tagungsbandes aufgeführt.