Institut für Informatik
AG Autonome Robotik

Masterarbeit

Loop Closure in TSDF basiertem SLAM

Patrick Hoffmann

Oktober 2022

Erstgutachter: Prof. Dr. Mario Porrmann
Zweitgutachter: Alexander Mock

Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit der Implementierung und Konzeption einer Lösung zur Detektion von **Schleifenschlüssen** in einem auf **Truncated Signed Distance Funtion (TSDF)** basierenden **Simultaneaous Localization and Mapping (SLAM)** Ansatz und der nachfolgenden Optimierung der Robotertrajektorie und TSDF-Karte. Zur Optimierung der Trajektorie des Robters nach der Identifikation eines Schleifenschlusses kommt die Bibliothek **GTSAM** zum Einsatz. Es wird evaluiert ob und unter welchen Voraussetzungen eine Nachbearbeitung auf Basis einer fertigen TSDF Karte mit zugehöriger initialer Trajektorie möglich ist und zusätzlich der Einsatz in einem inkrementellen SLAM Ansatz geprüft. Darüber wird untersucht, inwiefern ein Teilupdate der TSDF basierten Karte möglich ist.

Abstract

This paper deals with the implementation and design of a solution for the detection of **loop closures** in a **Truncated Signed Distance Function (TSDF)** based **Simultaneaous Localization and Mapping (SLAM)** approach and the subsequent optimization of the robot trajectory and TSDF map. The **GTSAM** library is used to optimize the robot's trajectory after identifying a loop closure. It is evaluated whether and under which conditions a postprocessing based on a finished TSDF map with associated initial trajectory is possible and additionally the use in an incremental SLAM approach is examined. Furthermore, it will be investigated to what extent a partial update of the TSDF based map is possible.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Diese Arbeit beschäftigt sich mit der Konzeption und Implementation einer Lösung zum *Loop Closure (Schleifenschluss)* Problem in *Truncated Signed Distance Fields* (TSDF) basiertem *Simulatneous Localization and Mapping* (SLAM). Diese Arbeit setzt auf den Implementationen und Konzepten von Eisoldt et al. [**?**] auf, die einen TSDF basierten, hardware-beschleunigten SLAM Ansatz (*HATSDF-SLAM*) implementiert und dessen Effizienz und Funktionalität verfiziert haben.

Unter SLAM versteht man den Prozess der Erstellung einer Karte einer unbekannten Umgebung, bei gleichzeitiger Positionsbestimmung innerhalb dieser unbekannten Umgebung. Das SLAM Problem ist ein *Henne-Ei-Problem*, da eine vollständige Karte benötigt wird, um die Pose eines Roboters akkurat bestimmen zu könnnen, während auf der anderen Seite eine akkurate Pose benötigt wird, um eine vollständige Karte aufbauen zu können. Für die Lösung dieses Problems gibt es sowohl in 2D als auch 3D verschiedene Lösungsansätze, die in Abschnitt 1.1 grob dargestellt sind. Die in diesem Exposé vorgestellte Arbeit beschränkt sich auf 6D SLAM. Unter einem Schleifenschluss versteht man in der mobilen Robotik den Prozess, geschlossene Kreise in abgelaufen Pfaden durch die im SLAM-Prozess erschlossene, zunächst unbekannte Umgebung, zu identifizieren. Nach der Identifikation dieser Kreise können Fehler, die sich im Kartierungsprozess akkumuliert haben, nachträglich berichtigt werden. Dies sorgt für eine konsistenteren Pfad und damit verbunden für eine akkuratere Karte. Als Basis für dieses Problem dient der GraphSLAM Ansatz von Borrmann et al. [**?**], die einen Pose-Graphen erstellen, in den aufeinanderfolgende Positionen während der Kartierung nacheinander eingefügt werden. Zur Registierung aufeinanderfolgender Datenscans verwenden Borrmann et al. den *Pairwise Iterative Closest Point Algorithm* (Pairwise ICP). Zur Detektion eines Schleifenschlusses wird eine einfache Distanz-Heuristik verwendet. Wichtige Voraussetzung für die Optimierung der Posen im Graphen ist eine Assoziation der Posen mit den zugehörigen Umgebungsdaten. Im Falle von Borrmann et al. sind dies jeweils die zum Zeitpunkt aufgenommenen Punktwolken. Diese Assozation ist bei TSDF basiertem SLAM, das eine diskretisierte TSDF-Karte verwendet, nicht möglich, solange nicht zusätzlich die entsprechenden Punktdaten abgespeichert werden. Wird ein Schleifenschluss detektiert, müsste dann jedoch die gesamte Karte neu erstellt werden.

Dies ist speicher- und zeitaufwändig. Dieses Problem markiert den Hauptbeitrag dieser Arbeit. Es gilt zu bestimmen, welcher Teil der TSDF Karte mit welcher Pose im Graph assoziiert werden kann, um anschließend die bereits existierende Karte zu modifizieren und zu optimieren.

1.1 Stand der Forschung

Simulataneous Localization and Mapping (SLAM) ist eines der größten Forschungsgebiete in der mobilen Robotik. Die Forschungen zu diesem Problem reichen zurück bis vor die Jahrtausendwende. Einer der bekanntesten Lösungen zur Registierung dreidimensionaler Daten ist der ICP Algorithmus nach Besl & McKay [**?**], der in einer Abwandlung ebenfalls von Borrmann et al. im GraphSLAM [**?**] verwendet wird. Bis heute gibt es zahlreiche Implementationen dieses Algorithmus.

Erste Forschungen mit TSDF basiertem SLAM stammen aus dem letzten Jahrzehnt. Izadi et al. [**?**] nutzen ein TSDF-Voxelgrid und eine Kinect-Tiefenkamera, um Umbegungen zu kartieren, während ein Nutzer die Kinect Kamera durch die Umbebung schwenkt. Whelan et al. [**?**] optimieren diesen Ansatz durch Nutzung eines Ringbuffers zur Kartierung großer Umgebungen. Im Gegensatz zu genannten TSDF Verfahren, nutzen Eisoldt et al. [**?**] einen Hardware beschleunigten TSDF basierten SLAM Ansatz, sowie einen 3D-Laserscanner anstelle einer Tiefenkamera.

Borrmann et al. [**?**] schufen in ihrer Arbeit eine gute Grundlage für die Integration von Schleifenschlüssen in SLAM-Verfahren auf Basis von Pose-Graphen, die bis heute vielfach verwendet wird.

1.2 Motivation

Ziel und Zweck dieser Arbeit ist die Lösung des Schleifenschluss Problems für TSDF basierte SLAM Verfahren. Als Basis dient der Ansatz von Eislodt et al. [**?**], der - obgleich performant und funktional - wie die meisten SLAM Verfahren bei der Kartierung großer Umgebungen stark anfällig für Drift ist. Um diesem Problem entgegen zu wirken, soll durch die Integration von Schleifenschlüssen in TSDF-Karten die Kartierung größerer Umgebungen ermöglicht werden. Um dieses Ziel zu erreichen, gilt es Datenassoziationen zwischen Posen und Zellen in der diskretisierten TSDF Karte herzustellen. In einer ersten Implementation soll dazu mittels Ray-Tracing eines simulierten Laserscans und der Detektion von Schnittpunkten mit der TSDF-Karte eine passende Indizierung der Zellen ermöglicht werden.

Diese Arbeit soll als Proof-of-concept für weitere Arbeiten auf diesem Gebiet dienen. Dazu werden die entwickelten Ansätze in einem Post-Processing Schritt auf die vom Eisoldt et al. [**?**] erstellte Karte, so wie den Pfad, den es noch zu extrahieren gilt, angewandt.

In einem Ausblick gilt es zu analysieren, ob die Implementationen sinnvoll beschleunigt werden können, um sie in ein Live Karierungssystem, wie HATSDF-Slam einbauen zu können, um zur Laufzeit Schleifen zu erkennen und die Karte zu optimieren.

Die Implementation dieses Prototyps wird ausschließlich in Software realisiert, allerdings erfolgt eine Untersuchung des Software-Prototyps auf Potenziale zur Hardware-Beschleunigung um Raum für Optimierungen im Rahmen zukünftiger Arbeiten zu eröffnen.

TODO: Update der Motivation? Untersuchung, ob das Loop Closing als Post-Processing angewandt werden kann, gegebenenfalls Integration in inkrementellen SLAM Möglichkeit schaffen mit beliebigen SLAM Verfahren koexistieren zu können (LC) Schaffen einer API? Untersuchen welche Art des Kartenupdates sinnvoll ist und welches nicht Untersuchen, ob der Aufwand die Karte im Teil zu updaten gerechtfertigt ist, Untersuchung von Fallstricken

1.3 Ansatz

Beschreiben, wie der initiale Aufbau dieser Arbeit ist:

1. Herangehensweise, wie soll das Problem gelöst werden? 2. Offenheit des Themas klar herausstellen 3. Mögliche Fallback's aufzeigen, falls sich herausstellt, dass das Thema so in der Bearbeitungszeit nicht möglich ist 4. Deutlich machen, dass die erste Herangehensweise dokumentiert wird 5. Erklären, dass Fallstricke und Probleme erläutert werden 6. Lösungsansätze für zukünftige Arbeiten definieren 7. Kenntlich machen, warum das Problem ïnkrementelles Map Update"nicht in der Zeit lösbar gewesen wäre

Kapitel 2

Grundlagen

2.1 Mathematische Grundlagen

Diese Sektion beschreibt die wesentlichen mathematischen Konzepte, die in dieser Arbeit zur Verwendung kommen.

2.1.1 Koordinatensysteme

Ein wesentliches Konzept bei der Verarbeitung räumlicher Daten ist die Verwendung von Koordinatensystemen. Sie werden genutzt um die Positionen von Daten und Objekten im Raum zu beschreiben. Koordinatensysteme können für sich alleine stehen oder relativ zu anderen Koordinatensystemen. Im Dreidimensionalen besitzt ein Koordinatensystem drei verschiedene Achsen (x, y und z-Achse), die jeweils im 90 Grad Winkel zueinander ausgerichtet sind. Rotationen im Raum werden beschrieben als Rotationen um die jeweiligen Achsen. Welche Achse in welche Richtung zeigt ist nicht eindeutig definiert. Es gibt jedoch verschiedene Standards beziehungsweise Konventionen wie das links- oder rechtshändische Koordinatensystem. In ROS wird konventionell ein rechtshändisches Koordinatensystem, abgebildet in Abbildung 2.1 dargestellt. Dies wird im Folgenden ebenfalls als Standard verwendet.

In der Robotik kommt es häufig vor, dass verschiedene (bewegliche) Komponenten relativ zu einem globalen Bezugssystem oder relativ zueinander beschrieben werden müssen. Die Bewegung eines übergeordneten Bezugssystems kann implizit für eine Veränderung der relativ zu diesem Bezugssystem platzierten Systeme führen. Dies lässt sich anhand eines Arm-Roboters zeigen, der mehrere miteinander verbundene Gelenke hat. Bewegt sich ein Gelenk werden automatisch auch die am Arm weiter außen befindlichen Gelenke mitbewegt. Aus Sicht des bewegten, übergordneten Bezugssystems hat sich die Position der untergeordneten Gelenke nicht verändert, aus Sicht des globalen Bezugssystems, wie zum Beispiel dem Montierungspunkt des Roboters, allerdings schon.

In ROS werden Anhängigkeiten zwischen Bezugssystemen in einer Baumstruktur, genannt *transformation tree (tf-tree)* dargestellt. Die Wurzel dieser Baumstruktur ist das globale Bezugssystem, wie zum Beispiel der Ursprung einer globalen Karte oder der Startpunkt der Trajektorie eines

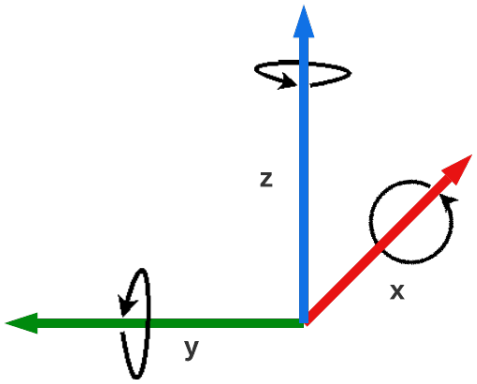

Abbildung 2.1: Schematische Darstellung der Konvention für Koordinatensysteme im ROS Framework. Die z-Achse zeigt nach oben, die x-Achse nach vorne und die y-Achse nach links. Die Rotation um die Achsen ist entsprechend der Konvention im Uhrzeigersinn.

Roboters. Das globale Bezugssystem kann beliebig gewählt werden. Auf diese Weise kann ein Koordinatensystem, welches relativ zu einem anderen gelegen ist im Baum als Kindknoten seinem Bezugssystem untergeordnet werden. Es wird nur die relative Transformation (s. Kapitel 2.1.1) zwischen den Systemen im Baum gespeichert. Dies hat den Vorteil, dass bei der Bewegung eines Systems die untergordneten Systeme nicht ebenfalls verändert werden müssen, da deren Transformationen relativ zum bewegten Bezugssystem angegeben sind und nicht global zur Wurzel des Baumes. Abbildung 2.2 zeigt ein Beispiel für einen Roboter mit mehreren voneinander abhängigen System.

Auch räumliche Daten wie zum Beispiel Punktenwolken aus Laserscannern können relativ zu verschiedenen Koordinatensystemen gesehen werden. So kann es nützlich sein die Punktwolke relativ zum Koordinatensystem des Scanners oder innerhalb des globalen Koordinatensystems zu betrachten. Um zwischen den Koordinatensystemen zu wechseln wird eine Koordinatensystem-transormation. Diese werden im folgenden Kapitel behandelt.

Transformationen

Eine Koordinatensystemtransformation ist ein Sonderfall einer mathematischen Transformation, die eine Menge X auf sich selbst abbildet:

$$f : X \to X \tag{2.1}$$

Eine Koordinatensystemtransformation beschreibt die Differenz zwischen zwei unterschiedlichen Koordinatensystemen und enthält sowohl die Translationsdifferenz als auch die Rotationsdifferenz. Zur Berechnung dieser Transformation zwischen zwei beliebigen Koordinatensystem C_1 und C_2 wird die absolute Translation und Rotation beider Koordinatensysteme zum Ursrungskoordinatensystem, wie zum Beispiel den Urpsrung einer Umgebungskarte, benötigt. Durch diese Rotations und Translationskomponenten beschreibt sich die absolute Position und Rotation

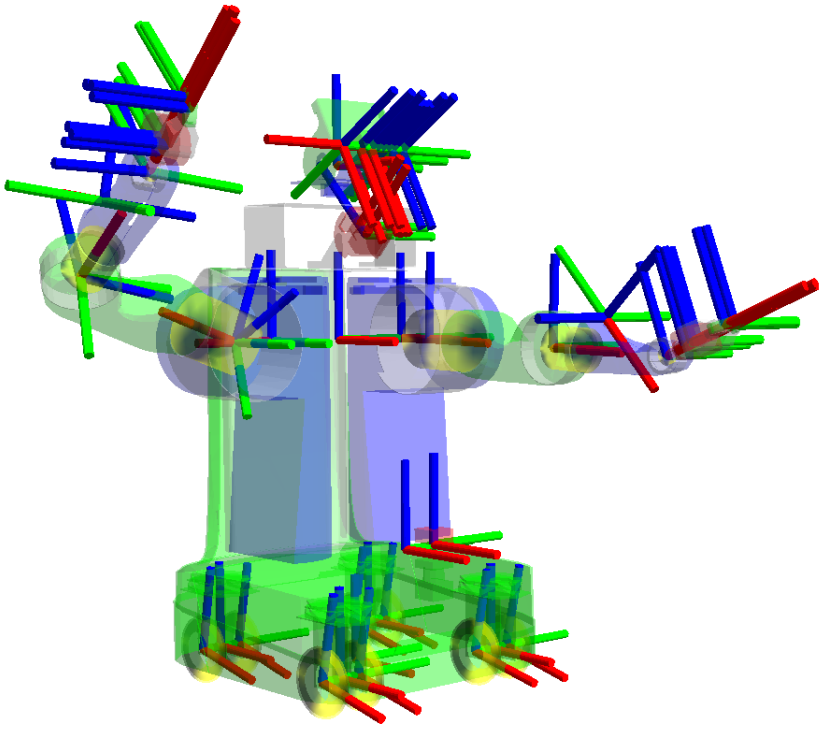

Abbildung 2.2: Schematische Darstellung eines Roboters und dessen beweglicher Teile. Die Ausrichtung der jeweiligen Gelenke wird mit einem lokalen Koordinatensystem beschrieben. Die globale Position einzelner Teile kann durch eine Verkettung der relativen Transformation in Richtung der Wurzel des Baumes bestimmt werden. Bild aus: TODO

der Koordinatensysteme im Raum aus Sicht des Ursprungskoordinatensystems C_{MAP}. Diese absolute Position und Rotation ist die Transformation von den jeweiligen Koordinatensystemen ins Urpsrungskoordinatensystem.

Es existieren diverse Darstellungsweisen für Koodinatensystemtransformationen. Die intuitivste Weise der Darstellung ist die Darstellung als Vektor. In folgendem ist die Transformation vom Koordinatensystem C_X ins Koordinatensystem C_{MAP} dargestellt. Die Variablen t_i bezeichnen dabei die Translationskomponenten und die Variablen r_i die Rotationskomponenten um die jeweiligen Achsen.

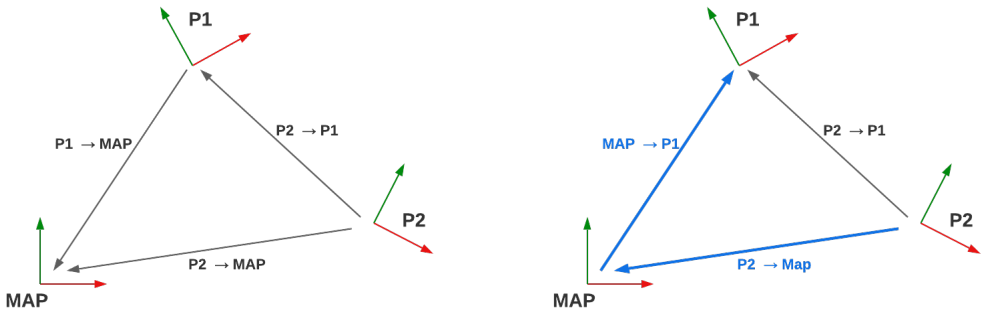

Abbildung 2.3: Schematische Darstellung der bestimmung einer Transformation zwischen zwei Roboter-Posen P_1 und P_2, hier zur Vereinfachung dargestellt in 2D. Gesucht ist die Transformation ($T_{P_2 \to P_1}$). Diese kann implizit bestimmt werden durch eine Verkettung der Transformationen $T_{P_2 \to MAP}$ und $T_{MAP \to P1}$, hier dargestellt im rechten Teil der Abbildung in blau. Die Transformation $T_{MAP \to P1}$ ist dabei nicht explizit gegeben. Sie kann berechnet werden durch eine Inversion der Transformation $T_{P1 \to MAP}$. Die finale Gleichung zur Berechnung der relativen Transformation ist dargestellt in Gleichung 2.3.

$$T_{C_X \to C_{MAP}} = \begin{pmatrix} t_x \\ t_y \\ t_z \\ r_x \\ r_y \\ r_z \end{pmatrix} \tag{2.2}$$

Neben der Vektordarstellung kann eine Transformation zusätzlich als eine $4x4$ Matrix dargstellt werden. Diese Darstellungsweise hat den Vorteil, dass die Transformation direkt per Matrixmultiplikation auf Daten wie um homogene Koordinaten erweiterte Punktdaten angewandt werden kann. Auch die direkt Kombination verschiedener Transformationen ist durch eine Matrixmultiplikation möglich. Hier gilt es zu beachten, dass Matrixmultiplikation nicht kommutativ ist und ein Vertauschen der Reihenfolge bei der Multiplikation zu unterschiedlichen Ergebnissen führen kann. Bei einer Verkettung von Transformationen durch Multiplikation wird die Matrix zuerst angewandt, welche am Ende der Multiplikation steht.

In dieser Arbeit werden Transformationen zum Beispiel verwendet um zu bestimmen, wie sich ein Roboter zwischen zwei Messungen bewegt hat. Diese Transformationen beschreiben relative Differenzen zwischen Roboterpositionen im Raum. Diese Roboterpositionen werden auch **Posen** genannt. Eine Pose ist dabei eine meist absolute Beschreibung der Translation und Rotation eines Roboters zu einem gewissen Zeitpunkt t aus Sicht des Ursprungskoordinatensystems wie zum Beispiel dem Map-Ursprung C_{MAP}.

Abbildung 2.3 zeigt die Mathematik hinter der Berechnung der Posedifferenz exemplarisch. Es

wird deutlich, dass eine gesuchte Transformation aus dem Koordinatensystem von Pose P_2 in das Koordinatensystem von Pose P_1 ($T_{P_2 \to P_1}$) gegeben ist durch:

$$(T_{P_2} \to T_{P_1}) = (T_{P_1 \to MAP})^{-1} * T_{P_2 \to MAP} \tag{2.3}$$

2.2 SLAM

1. Grundlagen von SLAM beschreiben -¿ Varianten des SLAM -¿ Bezug zu HATSDF-SLAM 2. Vorraussetzungen (Repräsentationen für Posen (Pfad) und Umgebung) 3. Überleitungen in weitere Sektionen machen -¿ Loop Closure als mögliche Verbesserung des SLAM -¿ TSDF als Kartenrepräsentation -¿ auf Vorteile von TSDF eingehen (z.B. einfache Integration in Marching Cubes Algorithmus)

2.3 Loop Closure

Warum wird Loop Closure benötigt? Welchen Mehrwert gibt es? Wie wäre ein grundlegendes vorgehen? verweisen auf Loop-Closure Kapitel

2.4 TSDF

Kapitel 3

Bibliotheken und Frameworks

3.1 ROS Framework

3.1.1 RViz

3.1.2 Bondcpp

3.2 Eigen

-¿ grundlegende Datenstrukturen die verwendet werden -¿

3.3 HDF5

-¿ Repräsentation der TSDF Datenstruktur

3.4 Pointcloud Library

3.5 LVR2 Framework zur Oberflächenrekonstruktion

-¿ Evaluation des SLAM und des Updates der Kartenrepräsentation

Kapitel 4

Loop Closure

Diese Sektion baut auf den Grundlagen aus Sektion 2.3 auf.

4.1 Detektion

4.2 Graphenoptimierung

4.3 Optimierungen

1. Vorregistrierung

2. Filtern der Punktwolke

3. Unterschiedliche Scan-Matching Varianten 1. ICP 2. GICP 3. Kurz auf Teaser++ eingehen 4. VGICP

Erklärung zur selbstständigen Abfassung der Masterarbeit

Name: Malte Hillmann

Geburtsdatum: 21.06.1997

Matrikelnummer: 969964

Titel der Masterarbeit: Erzeugung von Cesium 3D Tiles aus LVR2 Large-Scale-Meshes für LOD-Rendering

Ich versichere, dass ich die eingereichte Masterarbeit[1] selbstständig und ohne unerlaubte Hilfe verfasst habe. Anderer als der von mir angegebenen Hilfsmittel und Schriften habe ich mich nicht bedient. Alle wörtlich oder sinngemäß den Schriften anderer Autoren entnommenen Stellen habe ich kenntlich gemacht.

Osnabrück, 29.08.2022 _____

Ort, Datum Unterschrift

[1]Bei einer Gruppenarbeit gilt o. für den entsprechend gekennzeichneten Anteil der Arbeit.