

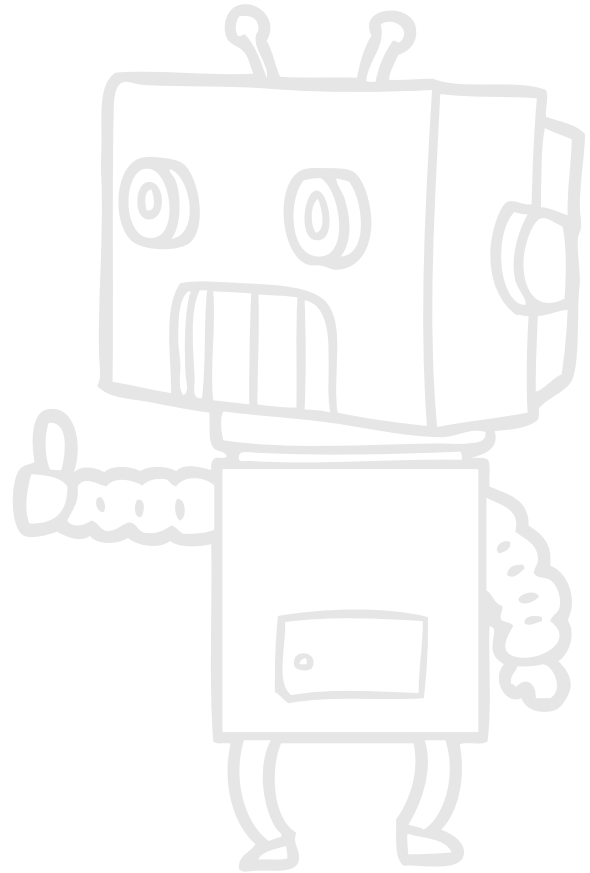
#11 TECHARTIKEL



# **SOCIAL ROBOTICS: AUTONOME POSITIONSBESTIMMUNG UND KARTIERUNG**

08.07.2024

Gabriel Schwald & Dr. Matthias Sommer



# *AraCom*

# Inhalt

1. **Einordnung in das Feld der Sozialen Robotik**
2. **Herausforderungen bei der simultanen Lokalisierung und Kartierung**
3. **Exemplarische SLAM-Pipeline für autonome mobile Roboter**
4. **Komponenten des mobilen Roboters**
5. **Fazit & Ausblick**
6. **Quellen**



# Social Robotics:

## Autonome Positionsbestimmung und Kartierung In dynamischen Umgebungen

In den letzten Jahren hat sich die Robotik rasant entwickelt und autonome Roboter sind zunehmend in der Lage, soziale Interaktionen mit Menschen auf eine neue Ebene zu bringen. Durch den Einsatz von fortschrittlichen Technologien können Roboter nicht nur menschenähnliche Fähigkeiten erlangen, sondern auch eine emotionale Verbindung zu Personen herstellen.

Soziale und emotional befähigte Roboter haben das Potenzial, das Leben der Menschen in verschiedenen Bereichen zu verbessern. Ihr Zweck besteht darin, menschenähnliche Interaktionen zu ermöglichen und ein breites Spektrum an Aufgaben zu erfüllen. Durch die Integration von fortschrittlichen Komponenten können Roboter soziale Verbindungen herstellen, emotionale Unterstützung bieten und bei der Bewältigung von Alltagsaufgaben helfen.

### 1. Einordnung in das Feld der Sozialen Robotik

Ist ein Roboter in der Lage, mit Menschen zu interagieren, spricht man von einem sozialen Roboter [1]. Soziale Roboter können den Pflegenotstand lindern, indem sie beispielsweise Überwachungsaufgaben übernehmen, bei denen verschiedene Vitalparameter gemessen werden. Darüber hinaus können sie als Assistenzroboter im Haushalt helfen, oder zur Unterhaltung von pflegebedürftigen Personen eingesetzt werden. Damit Roboter mit Menschen interagieren können, müssen sie über bestimmte Fähigkeiten verfügen. Dazu gehört beispielsweise die Fähigkeit, natürliche Signale wie Gesten zu interpretieren und Emotionen zu erkennen und auszudrücken.

Um die Mobilität von sozialen Robotern zu gewährleisten, ist es wichtig, dass der Roboter in der Lage ist, sich in einer unbekanntem Umgebung zu orientieren und zu manövrieren. Dieses Problem der gleichzeitigen Standortbestimmung und Kartierung wird unter dem Begriff SLAM (engl.: *Simultaneous Localization and Mapping*) zusammengefasst.

## 2. Herausforderungen bei der simultanen Lokalisierung und Kartierung

Die simultane Lokalisierung und Kartierung in dynamischen Umgebungen basiert auf zwei Hauptschritten. Zum einen beschäftigt sich der Lokalisierungsschritt mit der Positionsbestimmung des mobilen Roboters auf einer Karte. Zum anderen wird während dieses Prozesses nach und nach eine genaue Repräsentation der Umgebung erstellt. Diese basiert auf der Pose des Roboters und den Messungen der Umgebung.

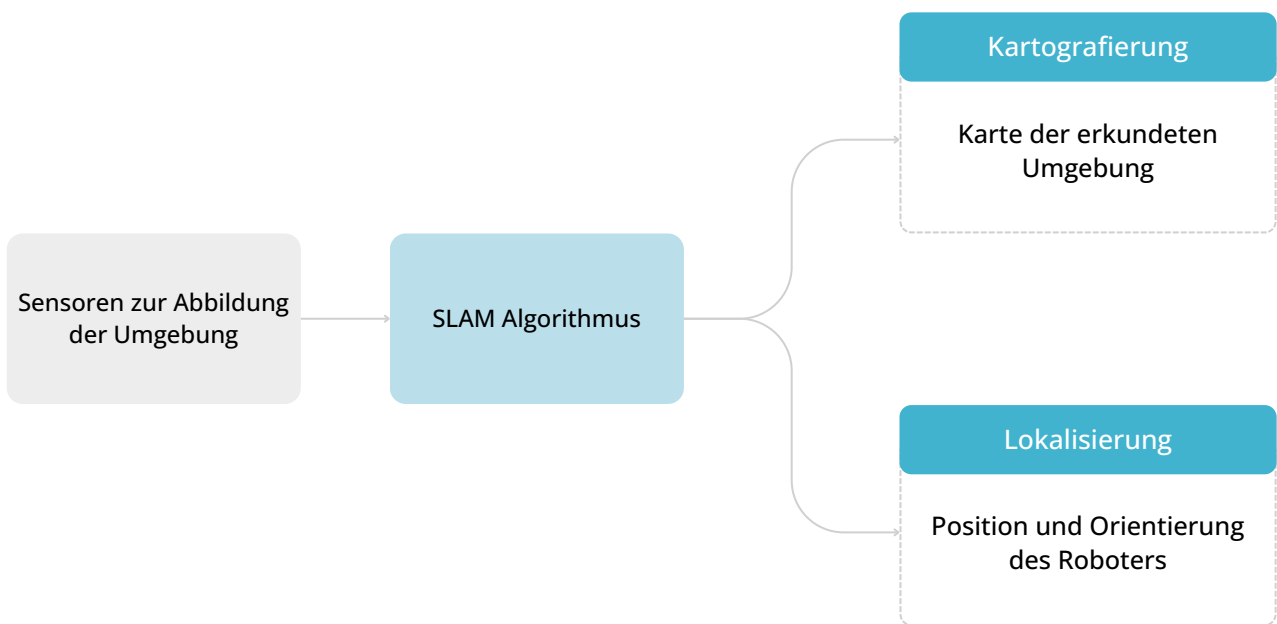


Abbildung 1: Allgemeine Darstellung der SLAM-Problemstellung

Ein großes Problem bei SLAM sind Ungenauigkeiten bei der Messung der Umgebung, welche durch Rauschen in den Sensoren verursacht werden. In der Praxis werden diese Fehler akzeptiert und mit Hilfe von stochastischen Lösungsstrategien modelliert. Die verschiedenen Algorithmen lassen sich am besten anhand der verwendeten Sensoren unterscheiden. Während direkte SLAM-Verfahren mit LiDAR und Beschleunigungssensoren arbeiten, sind visuelle SLAM-Algorithmen (vSLAM) auf Mono- oder Stereokameras beschränkt.



Abbildung 2: Eine generische feature-basierte vSLAM-Pipeline

SLAM wird insbesondere in Umgebungen eingesetzt, in denen andere Ortungstechniken (z.B. GPS) nicht verfügbar oder nicht genau genug sind. Außerdem ermöglicht die Kartierung die Navigation in unbekanntem Gebieten (z.B. eingestürzte Häuser).

### 3. Herausforderungen bei der simultanen Lokalisierung und Kartierung

Der gewählte Modellierungsansatz ist ebenfalls entscheidend für die Pipeline, die von den Sensormessdaten zur Karte führt. Während die genauen Details hier keine Rolle spielen, ist es ein üblicher Ansatz, die verschiedenen Messzeitpunkte als Knoten eines Graphen zu modellieren. Die Verbindungen (Kanten) zwischen den Knoten stellen die geschätzte räumliche Anordnung dar. Betrachtet man die Knoten als Massen und die Kanten als Federn, so berechnet die Optimierung eine maximal entspannte Anordnung der Federn. Diese Verfahren gehören daher zur Gruppe der Posen-Graph-Optimierung (PGO) Algorithmen.

Die in Abbildung 2 dargestellte Pipeline ist eine abstrahierte Variante des Google Cartographer [2] Algorithmus, welcher das am weitesten verbreitete PGO-Verfahren ist. Anstelle konkreter Algorithmen sind hier die einzelnen Schritte dargestellt, die in der einen oder anderen Form von allen aktuellen Lösungsstrategien umgesetzt werden.

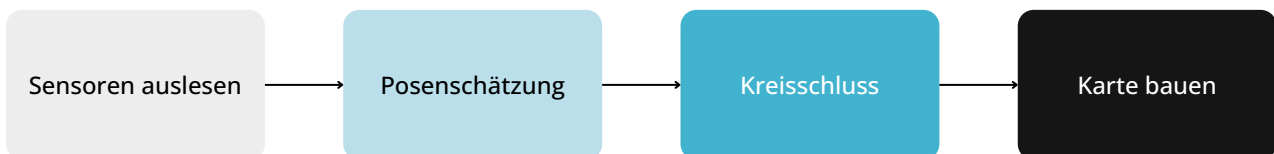


Abbildung 3: Vereinfachte Pipeline von Google Cartographer

#### 3.1 Sensordaten auslesen und vorbereiten

In einem ersten Schritt werden alle verfügbaren Sensoren abgefragt, um deren Informationen über die Umgebung (LiDAR, Kamera, ...) bzw. den inneren Zustand des Roboters (z.B. Beschleunigungssensoren, ...) zu erhalten. Dabei kann es notwendig sein, die Informationsdichte nachträglich künstlich zu reduzieren, um eine Verarbeitung der Daten in angemessener Zeit zu ermöglichen (z.B. Reduktion der Auflösung des Kamerabildes, oder Merkmalerkennung s. Abb. 1).

### 3.2 Pose des mobilen Roboters schätzen

In der nächsten Phase wird die aktuelle Pose, d.h. die Position und Orientierung des Roboters geschätzt. Dazu werden die zuletzt gesammelten Daten, mit denen aus dem vorherigen Durchlauf der SLAM-Pipeline verglichen. Aus den Unterschieden zwischen den beiden Datensätzen kann auf eine relative Bewegung geschlossen werden. Die dazu verwendeten Techniken unterscheiden sich je nach Sensortyp. Bei Bildern wird etwa die Bewegung von Punkten zwischen aufeinanderfolgenden Aufnahmen untersucht.

### 3.3 Kreisschluss und Fehlerkorrektur des Pfades

Im dritten Schritt wird der in Phase 2 akkumulierte Fehler korrigiert. Durch die Schätzung der relativen Bewegung in jeder Iteration sammeln sich die unvermeidlichen, minimalen Fehler mit der Zeit an. Um dem entgegenzuwirken, werden in modernen SLAM-Ansätzen Loop Closure Algorithmen verwendet. Die Grundidee dabei ist, dass Orte, die bereits zu einem früheren Zeitpunkt besucht wurden, wiedererkannt werden. Wird ein solcher Kreisschluss erkannt, kann der Pfad zwischen den beiden Besuchen korrigiert werden, da eine Referenz für die aktuelle Position des Roboters vorliegt. Diese Aufgabe effizient zu lösen ist eine große Herausforderung, da eine große Menge an Daten gespeichert und verglichen werden muss. In diesen Schritt fällt auch die oben genannte Optimierung des Posen-Graphs, wobei die Details über den Rahmen dieses Artikels hinaus gehen würden.

### 3.4 Zusammenbauen der Karte

Der letzte Schritt besteht darin, die in Schritt 1 gesammelten Daten in die Karte an der in Schritt 2 und 3 festgelegten Position einzutragen. Eine beliebte Art der Kartendarstellung ist zum Beispiel ein Belegungsgitter. Jedes Feld des Gitters enthält einen Wahrscheinlichkeitswert, der angibt, mit welcher Sicherheit dieser Ort belegt ist. Felder, die z. B. an einer Wand liegen, erhalten den Wert 1, während freie Felder den Wert 0 erhalten.

## 4. Komponenten des mobilen Roboters

Zum Schluss noch ein Blick auf die Komponenten, die wir verwenden, um einem mobilen Roboter die Fähigkeiten zur Positionsbestimmung und Kartierung zu geben. Die Jetson-Plattform von NVIDIA [6] umfasst eine Reihe von energieeffizienten Mikrocontrollern. Neben einer leistungsstarken CPU verfügen diese über moderne GPU-Cores, die bei parallelisierbaren Aufgaben (z.B. für maschinelles Lernen) erhebliche Beschleunigungen bringen. Die zahlreichen Anschlüsse ermöglichen eine zusätzliche Erweiterung der Fähigkeiten durch Peripheriegeräte wie Kamera, LiDAR oder Beschleunigungssensor. Damit eignet sich die Jetson-Plattform ideal für Anwendungen in der mobilen Robotik, bei denen lokale Rechenleistung benötigt wird.

Das Robot Operating System (ROS) [3] ist das Standardframework, das in der Robotik verwendet wird, um die verschiedenen Aufgaben zu orchestrieren. Intern basiert es auf sogenannten Knoten, Topics und Services. Knoten können autonom Aufgaben lösen (z.B. einen Sensor auslesen), aber auch über Topics mit anderen Knoten kommunizieren. Dazu veröffentlicht ein Knoten Nachrichten in einem vordefinierten Format für ein Topic. Andere Knoten, die dieses Topic abonniert haben, führen beim Empfang der Nachricht automatisch interne Aktionen aus.

Ein idealer Sensor für SLAM im 2D-Raum ist der schon oft erwähnte LiDAR-Sensor (engl. Light Detection And Ranging). Der auf einer rotierenden Plattform montierte Laserstrahl erzeugt ein Bild der Umgebung aus der Vogelperspektive. Die Sensorausgänge können dann fast direkt zur Kartenerstellung verwendet werden (siehe Abbildung 3). Während LiDAR-Sensoren sehr beliebt sind, bringen sie auch einige Schwächen mit sich. So können sie durchsichtige Gegenstände, wie zum Beispiel Glaswände, nicht erkennen.

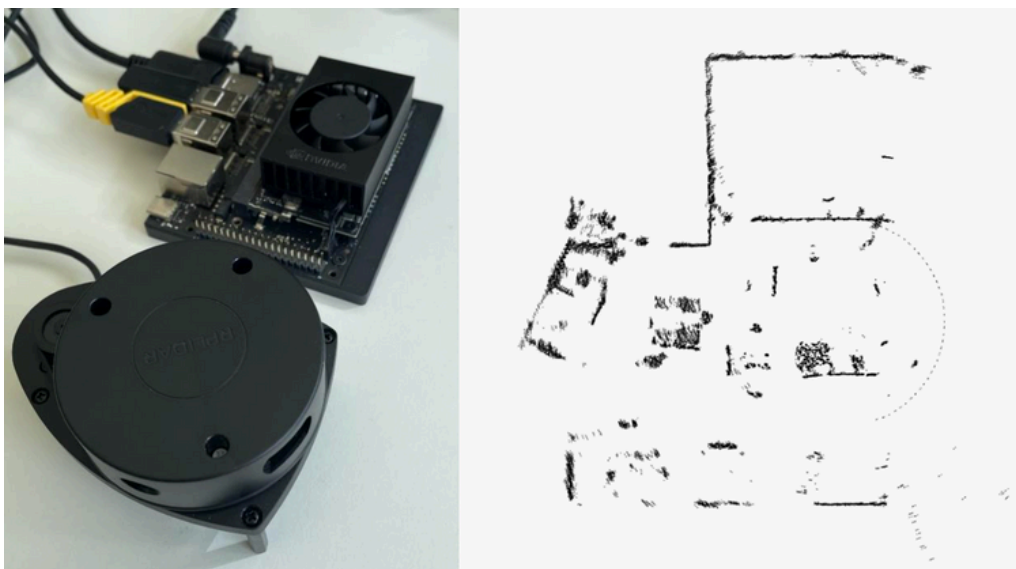


Abbildung 4: (links) NVIDIA Jetson mit LiDAR Sensor, (rechts) Grundriss generiert mit LiDAR

## 5. Fazit & Ausblick

SLAM ist nur einer der notwendigen Bausteine, um mobile soziale Robotik zu ermöglichen. Obwohl bereits seit über 15 Jahren Open-Source-Algorithmen existieren, die eine autonome Navigation ermöglichen, gibt es noch viel Entwicklungspotenzial. Gerade die Robustheit, also die Fehlertoleranz, ist ein Bereich, in dem Verbesserungen notwendig sind, um eine wirklich autonome Navigation zu ermöglichen.

Auch Teilkomponenten von SLAM werden bereits durch neuronale Netze verbessert. So ist z.B. bei vSLAM der weit verbreitete Oriented FAST and Rotated BRIEF (ORB) Algorithmus zur Merkmalerkennung Stand der Technik [4]. Neuronale Netze werden aber nicht nur zur Bildanalyse eingesetzt, sondern können auch effizientere Lösungen für das oben genannte Loop Closure-Problem sein. So verwendet OverlapNet [5] keine manuell erstellten Gleichungen, um Kandidaten für bereits besuchte Gebiete zu finden, sondern ein siamesisches neuronales Netz.

In vielen Bereichen der sozialen Robotik erweitert der Einsatz von Methoden des maschinellen Lernens den Raum des Machbaren. Ein gutes Beispiel ist die in einer vorherigen Abschlussarbeit implementierte Emotionserkennung, die wir unter <https://github.com/AraComITServicesGmbH/emotionale-robotik> bereitstellen. In Kooperation mit verschiedenen Bildungseinrichtungen forschen wir aktiv im Feld der mobilen sozialen Robotik. Dieser Blogartikel gab einen Einblick in die Masterarbeit "SLAM für mobile Roboteranwendungen in häuslichen Umgebungen", die in Zusammenarbeit mit dem Lehrstuhl für Mechatronik der Universität Augsburg, entsteht.

Wir bieten immer studentische Abschlussarbeiten an, die unter [https://aracom.de/deine\\_karriere/abschlussarbeit/](https://aracom.de/deine_karriere/abschlussarbeit/) ausgeschrieben sind.



## Quellen

1. O. Bendel, Soziale Roboter. Wiesbaden: Springer Fachmedien Wiesbaden, 2021. ISBN: 978-3-658-31113-1
2. W. Hess, D. Kohler, H. Rapp and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 1271-1278, doi: 10.1109/ICRA.2016.7487258.
3. Steven Macenski et al. "Robot Operating System 2: Design, architecture, and uses in the wild". In: Science Robotics 7.66 (2022)
4. Carlos Campos et al. "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM", IEEE Transactions on Robotics 37(6):1874-1890, Dec. 2021
5. Chen, X., Läbe, T., Milioto, A. et al. OverlapNet: a siamese network for computing LiDAR scan similarity with applications to loop closing and localization. Auton Robot 46, 61-81 (2022). <https://doi.org/10.1007/s10514-021-09999-0>
6. <https://www.nvidia.com/de-de/autonomous-machines/embedded-systems/>, abgerufen am 24.06.2024

