Projekt: Mobile Robotik DLBROESR01_D

Fallstudie

Studiengang: Angewandte Künstliche Intelligenz

Sven Behrens

Matrikelnummer: 42303511

Prof. Dr. Florian Simroth

5. Februar 2026

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

AMR Automated Mobile Robot

C-Space Configuration Space

PRM Probabilistic Roadmap

RRT Rapidly-exploring Random Tree

1 Einleitung

Die Intralogistik befindet sich in einem tiefgreifenden Wandel. Angetrieben durch die steigende Kundenerwartungen an Liefergeschwindigkeit setzen immer mehr Unternehmen auf automatisierte mobile Roboter (Automated Mobile Robot (AMR)) für den innerbetrieblichen Warentransport (Fragapane et al., 2021). Eine zentrale Herausforderung beim Einsatz solcher Systeme stellt die Pfadplanung dar: Der Roboter muss in der Lage sein, kollisionsfrei von einem Startpunkt zu einem Zielpunkt zu navigieren und dabei sowohl statische Hindernisse als auch die eigene Geometrie zu berücksichtigen. Vor diesem Hintergrund wurde im Rahmen des Moduls „Mobile Robotik“ an der IU Internationalen Hochschule ein Softwareprototyp entwickelt, der verschiedene etablierte Pfadplanungsalgorithmen implementiert und vergleichend evaluiert.

Das primäre Projektziel bestand in der Implementierung eines Softwaresystems, das für einen omnidirektionalen Roboter in einer polygonbasierten Lagerumgebung kollisionsfreie Pfade berechnet. Die zentrale Forschungsfrage konzentrierte sich darauf, wie verschiedene Pfadplanungsalgorithmen hinsichtlich Pfadqualität, Rechenzeit und Robustheit in unterschiedlich komplexen Umgebungen performieren. Besondere Aufmerksamkeit galt dabei der korrekten Konstruktion des Konfigurationsraums (Configuration Space (C-Space)), der die Robotergeometrie in die Hindernisdarstellung integriert.

Die methodische Vorgehensweise gliederte sich in mehrere aufeinander aufbauende Phasen. Zunächst wurde eine modulare Softwarearchitektur in Python entwickelt, die eine klare Trennung zwischen Umgebungsmodellierung, Robotergeometrie, Konfigurationsraumberechnung und Pfadplanung vorsieht. Anschließend wurden fünf verschiedene Algorithmen implementiert und verglichen: Die graphbasierten Verfahren A*, Dijkstra und Best-First-Search sowie die samplingbasierten Methoden Rapidly-exploring Random Tree (RRT) und Probabilistic Roadmap (PRM). Die Evaluation erfolgte auf drei Testumgebungen mit steigender Komplexität unter Verwendung von drei unterschiedlichen Robotergeometrien.

Der gewählte Ansatz zeichnet sich durch seine Erweiterbarkeit aus. Neben dem zweidimensionalen C-Space für omnidirektionale Roboter wurde zusätzlich ein dreidimensionaler Konfigurationsraum implementiert, der die Orientierung des Roboters als dritte Dimension berücksichtigt. Dies ermöglicht die Pfadplanung für nicht-holonome Roboter und demonstriert die Skalierbarkeit des entwickelten Systems. Durch die systematische Evaluation verschiedener Algorithmus-Roboter-Umgebungs-Kombinationen wurden quantitative Erkenntnisse gewonnen, die als Entscheidungsgrundlage für den praktischen Einsatz dienen können.

Die vorliegende Fallstudie gliedert sich wie folgt: Nach der Beschreibung der Projektumgebung wird der iterative Entwicklungsansatz erläutert. Die Hauptabschnitte behandeln die 2D- und 3D-Pfadplanung mit ihren jeweiligen Ergebnissen. Abschließend werden die Projektergebnisse kritisch reflektiert und Verbesserungspotenziale aufgezeigt.

2 Hauptteil

2.1 Projektumgebung

Zu Beginn des Projekts wurde ein GitHub-Repository[1] angelegt, um eine nachvollziehbare Versionsverwaltung zu gewährleisten. Anschließend wurde eine virtuelle Python-Umgebung mit `venv` eingerichtet, in der alle projektspezifischen Abhängigkeiten isoliert installiert wurden. Als Implementierungssprache wurde Python gewählt, da der Autor hier über die meiste Erfahrung verfügt.

2.2 Zwei-Phasen-Analyseansatz

Nach dem Einrichten der Projektstruktur wurde die Entwicklung in zwei aufeinander aufbauende Phasen gegliedert. Diese Vorgehensweise ermöglichte es, zunächst die grundlegenden Konzepte der Pfadplanung im einfacheren Fall zu validieren, bevor die Komplexität erhöht wurde.

In der ersten Phase wurde die Pfadplanung für omnidirektionale Roboter implementiert. Bei diesem Robotertyp kann sich der Roboter in jede Richtung bewegen, ohne seine Orientierung ändern zu müssen. Der resultierende Konfigurationsraum ist zweidimensional und umfasst lediglich die Position (x, y) des Roboters.

Die zweite Phase erweiterte das System um die Berücksichtigung der Roboterorientierung. Für nichtholonome Roboter, die sich nicht seitwärts bewegen können, ist die Orientierung θ eine zusätzliche Freiheitsdimension. Der Konfigurationsraum wird dadurch dreidimensional (x, y, θ), was die Komplexität der Kollisionsprüfung und Pfadsuche erheblich steigert. Durch den modularen Aufbau der ersten Phase konnten wesentliche Komponenten wiederverwendet und gezielt erweitert werden.

2.3 2D-Pfadplanung

2.3.1 Umgebungsmodellierung

Die Lagerumgebung wird durch eine rechteckige Grundfläche mit polygonalen Hindernissen modelliert. Jedes Hindernis ist als Liste von Eckpunkten definiert, die ein geschlossenes Polygon bilden. Diese Repräsentation ermöglicht eine flexible Darstellung beliebiger konvexer und konkaver Hindernisformen.

Für die Evaluation wurden drei Testumgebungen mit steigender Komplexität erstellt. Die *easy*-Umgebung umfasst eine Fläche von 20×15 Metern mit zwei rechteckigen Hindernissen, die einen einfachen Umweg erfordern. Die *medium*-Umgebung erweitert die Fläche auf 25×20 Meter und enthält sechs Hindernisse, die korridorartige Strukturen bilden. Die *hard*-Umgebung stellt mit 30×25 Metern und 14 Hindernissen

[1] https://github.com/svenb23/mobile-robotik-pfadplanung

eine labyrinthartige Struktur dar, die deutlich komplexere Pfade erfordert (siehe Fig. 1 im Anhang).

2.3.2 Robotergeometrie

Der Roboter wird als Polygon modelliert, dessen Eckpunkte relativ zum Ursprung definiert sind. Diese Darstellung ermöglicht eine einheitliche Behandlung beliebiger Roboterformen bei der Kollisionsprüfung. Die Methode `at(x, y, theta)` transformiert das Roboterpolygon an eine beliebige Position mit optionaler Rotation.

Für die Evaluation wurden drei unterschiedliche Geometrien implementiert. Der kreisförmige Roboter mit einem Radius von 0,5 Metern wird als 16-Eck approximiert und repräsentiert einen omnidirektionalen Roboter ohne bevorzugte Ausrichtung. Der rechteckige Roboter mit den Abmessungen $0{,}8 \times 0{,}5$ Meter entspricht einem typischen Transportroboter. Der dreieckige Roboter mit einer Basis von 0,8 Metern und einer Höhe von 0,6 Metern zeigt mit seiner Spitze nach vorne, kann sich jedoch als omnidirektionaler Roboter in alle Richtungen bewegen. Diese asymmetrische Form verdeutlicht den Einfluss der Geometrie auf den Konfigurationsraum (siehe Fig. 2 im Anhang).

2.3.3 Konfigurationsraumberechnung

Der Konfigurationsraum (C-Space) transformiert das Pfadplanungsproblem von einem ausgedehnten Roboter zu einem Punktroboter. Dabei werden die Hindernisse um die Robotergeometrie erweitert, sodass eine Kollisionsprüfung nur noch für den Referenzpunkt des Roboters erforderlich ist.

Die Berechnung erfolgt durch Diskretisierung des Arbeitsraums in ein gleichmäßiges Gitter mit konfigurierbarer Auflösung (Standard: 0,5 Meter). Für jede Gitterzelle wird das Roboterpolygon an der entsprechenden Position platziert und mittels der Shapely-Bibliothek (Gillies et al., 2024) auf Überschneidungen mit den Hindernissen geprüft. Zusätzlich wird sichergestellt, dass der Roboter vollständig innerhalb der Umgebungsgrenzen liegt. Das Ergebnis ist ein boolesches 2D-Array, in dem besetzte Zellen als `True` und freie Zellen als `False` markiert sind.

Durch diese Vorverarbeitung reduziert sich die Pfadplanung auf eine Graphsuche im diskretisierten Gitter. Die gewählte Auflösung stellt einen Kompromiss zwischen Genauigkeit und Rechenaufwand dar (siehe Fig. 3 im Anhang).

2.3.4 Pfadplanungsalgorithmen

2.3.5 Evaluation

2.4 3D

2.4.1 Umgebungsmodellierung

2.4.2 Robotergeometrie

2.4.3 Konfigurationsraumberechnung

2.4.4 Pfadplanungsalgorithmen

2.4.5 Evaluation

3 Fazit

3.1 Zielerreichung und Projektergebnisse

3.2 Kritische Reflexion

3.3 Verbesserungspotenziale und Optimierungsansätze

3.4 Ausblick

Projektrepository

Der vollständige Quellcode ist im GitHub-Repository verfügbar: https://github.com/svenb23/mobile-robotik-pfadplanung

Literatur

Fragapane, G., De Koster, R., Sgarbossa, F., & Strandhagen, J. O. (2021). Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *European Journal of Operational Research*, *294*(2), 405. https://doi.org/10.1016/j.ejor.2021.01.019

Gillies, S., et al. (2024). *Shapely* (Version 2.0) [[Python-Bibliothek]]. https://shapely.readthedocs.io/

Verzeichnis der Anhänge

Anhang

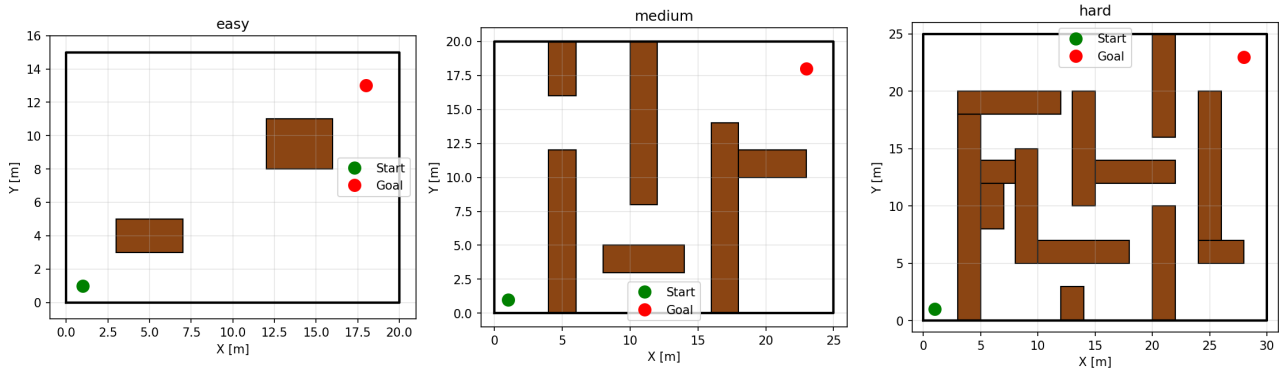

Abbildung 1: Übersicht der drei Testumgebungen: easy (links), medium (Mitte), hard (rechts)

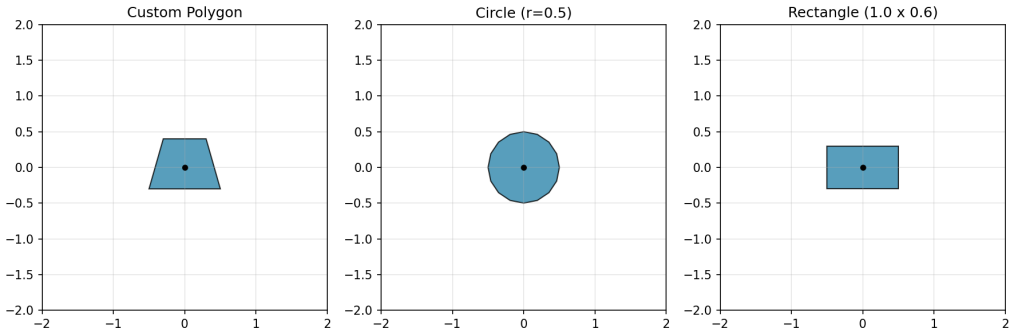

Abbildung 2: Die drei Robotergeometrien: Kreis (links), Rechteck (Mitte), Dreieck (rechts)

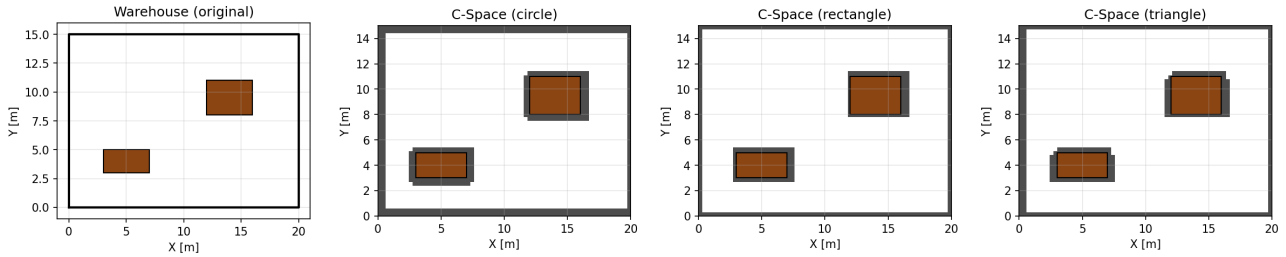

Abbildung 3: Konfigurationsräume für verschiedene Robotergeometrien: Die grauen Bereiche zeigen die erweiterten Hindernisse