Jan-Gerrit Jaeger
C. Brandau, M. Steinbrink, D. Tutsch

Bergische Universität Wuppertal
Fakultät für Elektrotechnik, Informationstechnik und Medientechnik
Lehrstuhl für Automatisierungstechnik / Informatik

Autonome Erkundung, Kartographierung und Wegfindung mittels Sensorfusion durch einen PhantomX MKIII

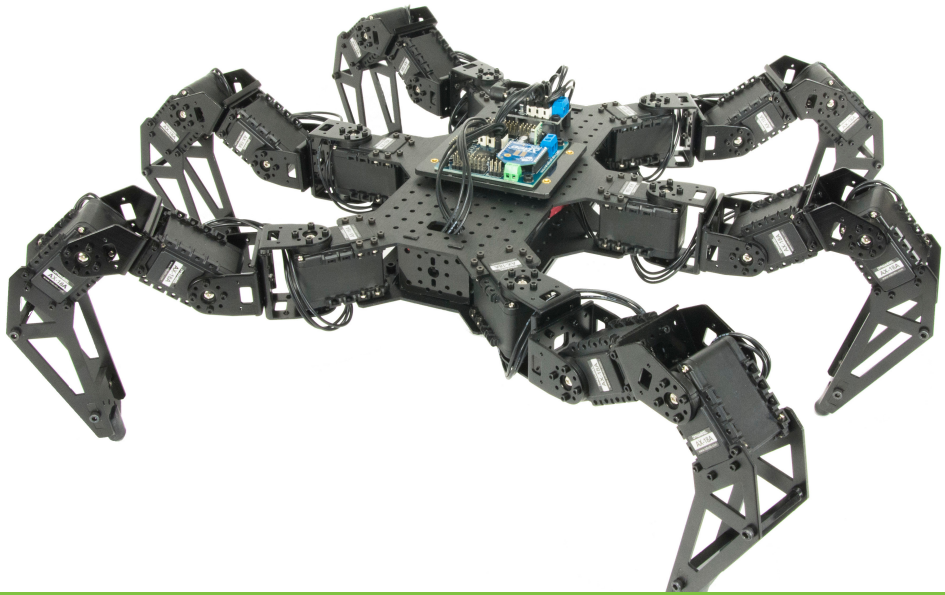
Echtzeit 2019
21. November

Inhaltsverzeichnis

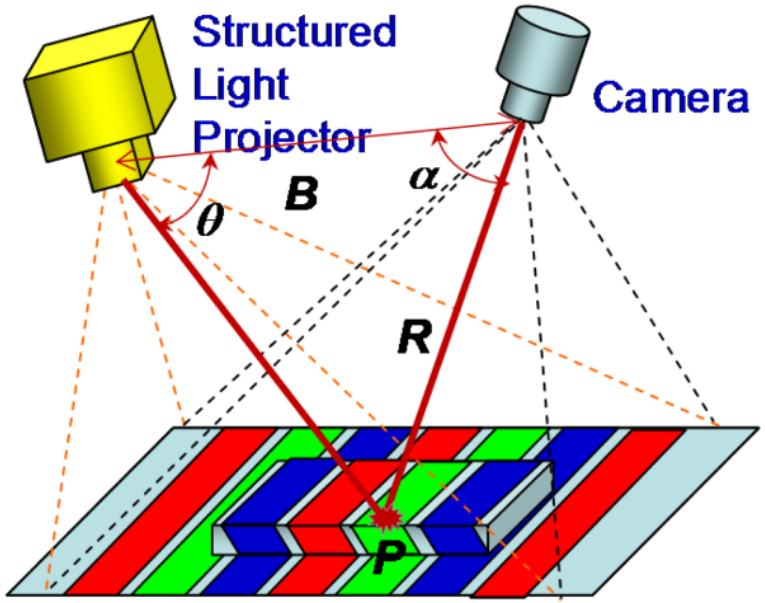
Motivation & Ziele

- Zunahme der Robotik durch fortschreitende Automatisierung
- Sensoren verfügbarer und günstiger
- Einsatz im privaten Bereich
  - Staubsaugroboter
  - Rasenmähroboter
  - Fahrerassistenzsysteme
- Viele offene technische Problemstellungen
- Autonome Navigation durch unbekannte Landschaften
  - Rettungsroboter
- Sensoren zur räumlichen Erfassung
- Algorithmen zur Verarbeitung der Daten

- PhantomX AX Hexapod Mark III
  - Sechsbeiniger Roboter
  - 18 Servomotoren
  - ArbotixM-Robocontroller
  - Lithium-Polymer-Akkumulator
- ODROID-XU4
  - ARM-basierter Einplatinenrechner
  - Zwei Quad-Core-Prozessoren
  - USB-Anschlüsse
  - Serielle Schnittstelle

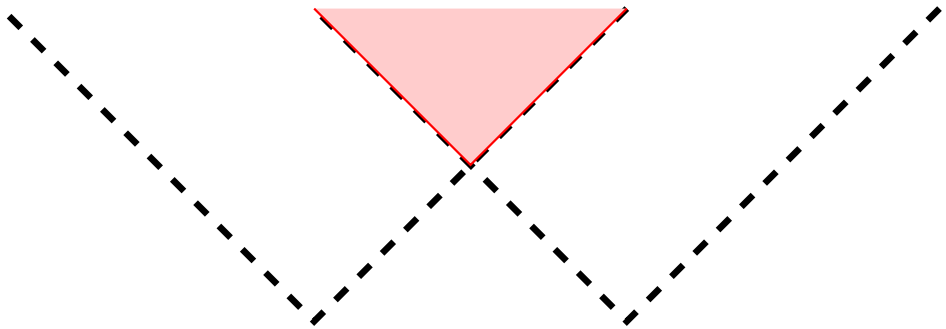


- RGB-D Kamera
  - Orbbec Astra Pro
  - Microsoft Kinect
  - Red, Green, Blue, Depth
  - Zwei unabhängige Kameras
  - Erfasst zusätzlich Tiefenwerte
  - Tiefenmessung mithilfe der Streifenprojektion
  - Einschränkung bei gewissen Oberflächen



- Lidar
  - RPLIDAR A1
  - Light Detection and Ranging
  - 360-Grad-Sensor
  - Distanzmessung mittels Laufzeit
  - Einschränkung bei gewissen Oberflächen
  - Pseudo-Objekte durch Reflexion

- Inertiale Messeinheit
  - Beschleunigungssensor
  - Gyroskop
  - Magnetometer
  - Fehleranfällig bei Integrationsverfahren
- Ultraschallsensor
  - Distanzmessung mittels Schallwellen
  - Weiter Ausbreitungswinkel
  - Einsatz von zwei Sensoren
  - Erkennung von Glasflächen



Grundlagen: ROS

- Robot Operating System
- Entwickelt für Robotik-Anwendungen
- Open-Source
- ROS dient als Middleware
- Pakete
  - Wesentlicher Bestandteil
  - Gekapselte Elemente
  - Kann Prozesse enthalten
  - Relevante Eigenschaften
  - Auslesen der Sensordaten
  - Auswertung der bereitgestellten Daten

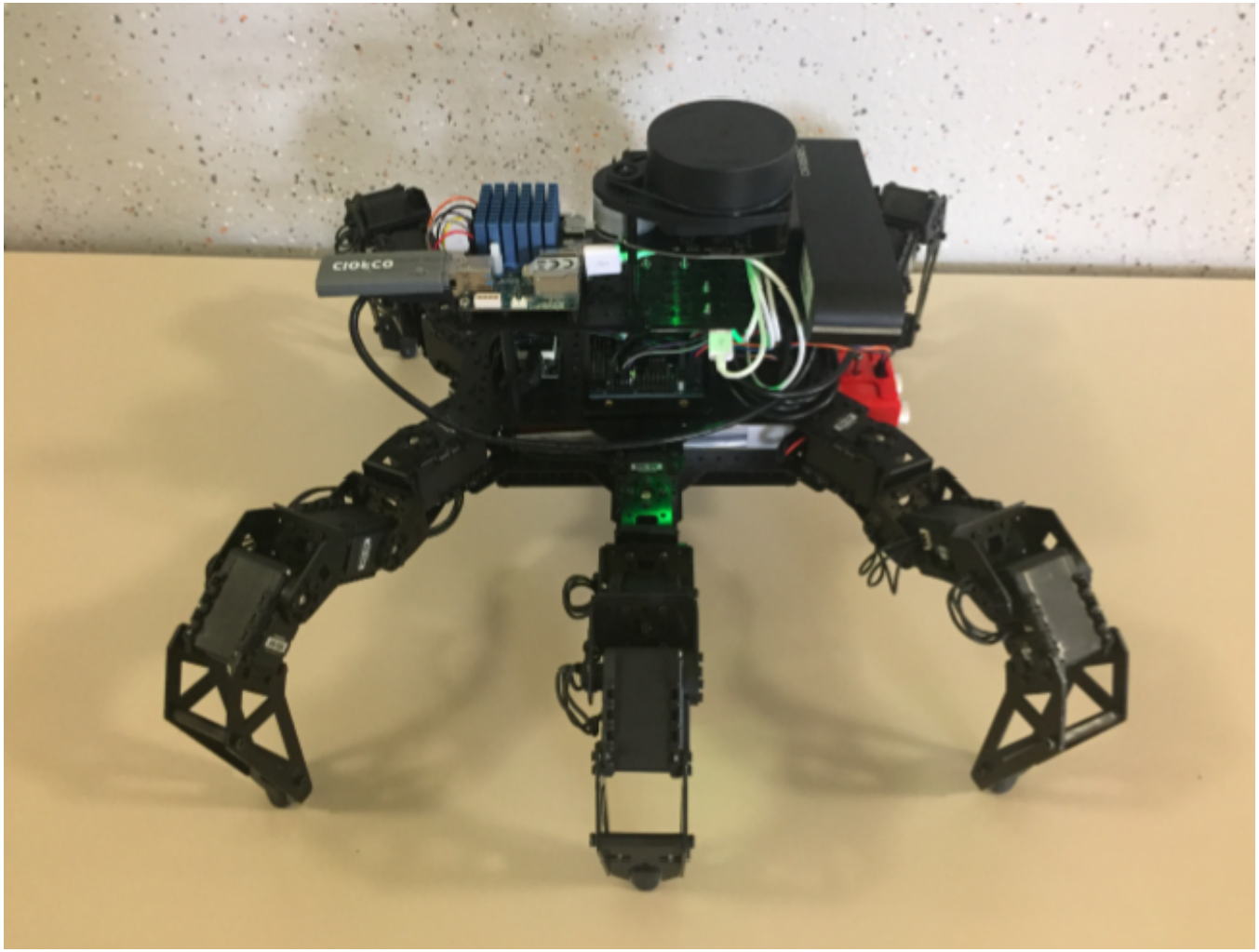
- Kapselung der Prozesse
  - Austausch der Komponenten vereinfacht
  - Keine direkte Kommunikation
  - Überwachbar durch ROS-eigene Werkzeuge
- Kommunikation
  - Master verwaltet die Kanäle
  - Prozessinterkommunikation über Adressen
  - Offene Nachrichtenkanäle (Topics)
  - Kanäle können abonniert werden
  - Synchrone Kommunikation

- Simultaneous Localization and Mapping
- Problem in der Robotik
- Henne-Ei-Problem
- Zwei Kategorien
  - Online SLAM
    - Schätzung: Neue Position und Karte
    - Verfügbare Sensordaten und Odometrie-Messungen
  - Full SLAM
    - Schätzung: Neue Position, Pfad und Karte
    - Verfügbare Sensordaten und Odometrie-Messungen
- Landmarke
  - Einzigartige Bereiche in den Aufnahmen
    - Ecken
    - Kanten

Grundlagen: SLAM

- Verschiedene Ansätze
  - Erweiterte Kalman-Filter
    - Online SLAM-Prinzip
    - Einzigartig identifizierbare Merkmale
    - Messpunkte werden mit alten Aufnahmen verglichen
    - Wahrscheinlichkeiten
  - Rao-Blackwellisierte Partikelfilter
    - Full SLAM-Prinzip
    - Partikelfilter
    - Partikel entspricht Bewegungsbahn und Karte
    - Wahrscheinlichkeitsberechnung für jeden Partikel
- Einsatz im dreidimensionalen Raum
  - Algorithmen für ein planares Gebiet
  - Inertiale Messeinheit
  - Vorfilterung

PhantomX MK3



Odometrie

- Lokalisierung des Roboters
- Positionierung durch Daten des Vortriebsystems
  - Relative Positionsbestimmung
  - Minimale und maximale Geschwindigkeit
  - Integration der Geschwindigkeitsvariabeln
    - Problem: Roboter weist eine Verzögerung auf
    - Lösung: Einbeziehung der IMU
  - Direkte Kinematik
    - Aufwendiger Ansatz
    - Bestimmung der Fußposition durch Ausrichtung der Servomotoren
    - Größe und Position der Beine wichtig
- Virtuelle Odometrie
  - Einbeziehung der verbauten Sensoren
  - Abgleich erstellter Scans
  - Berechnung von Position und Orientierung

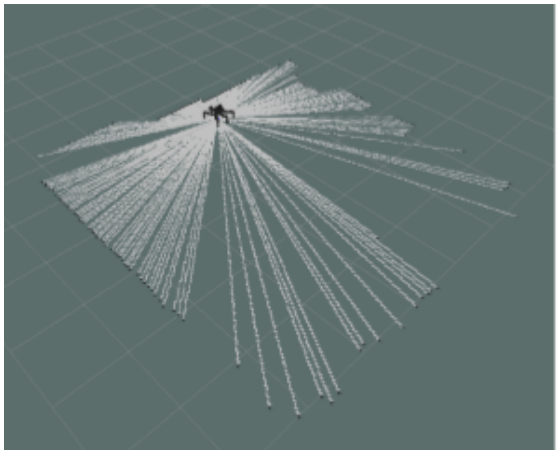
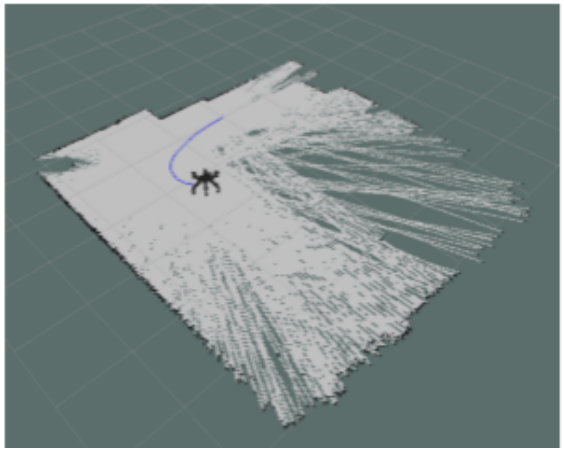
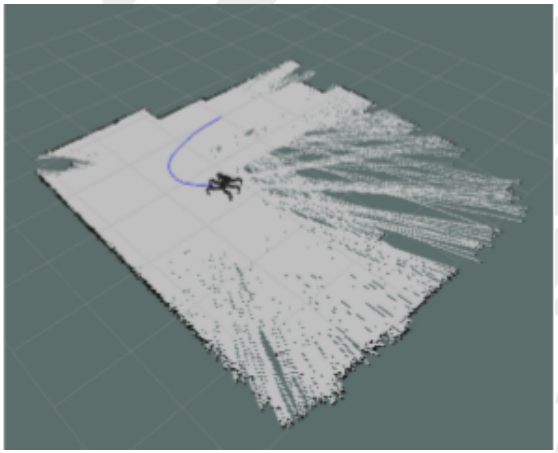
Odometrie

- Virtuelle Odometrie
  - Hector SLAM
    - Lidar-Sensor
    - Positionsbestimmung relativ zur vorherigen Position
  - Laser Scan Matcher
    - Lidar-Sensor
    - Kombination mit IMU und direkter Kinematik
    - Schätzung der neuen Position
    - Erzeugt neue Schlüsselbilder
- Auswertung

SLAM-Algorithmen

- GMapping
  - Benötigt:
    - Horizontal verbauten Lidar-Sensor
    - Inertiale Messeinheit
    - Odometrie-Daten
    - Rao-Blackwellisierten Partikelfilter
  - Erstellt eine 2D-Rasterkarte mit Lokalisierung
  - Auflösung von 0,06 Metern pro Feld
    - Angepasst an die Auflösung des Lidars
  - Aktualisierungsintervall der Karte: Eine Sekunde
  - CPU-Auslastung durchschnittlich bei 15 Prozent

- Hector SLAM
  - Benötigt:
    - Horizontal verbauten Lidar-Sensor
  - Erstellt eine 2D-Rasterkarte
  - Auflösung von 0,06 Metern pro Feld
    - Angepasst an die Auflösung des Lidars
  - Positionierung durch Transformation
  - CPU-Auslastung durchschnittlich bei 17 Prozent

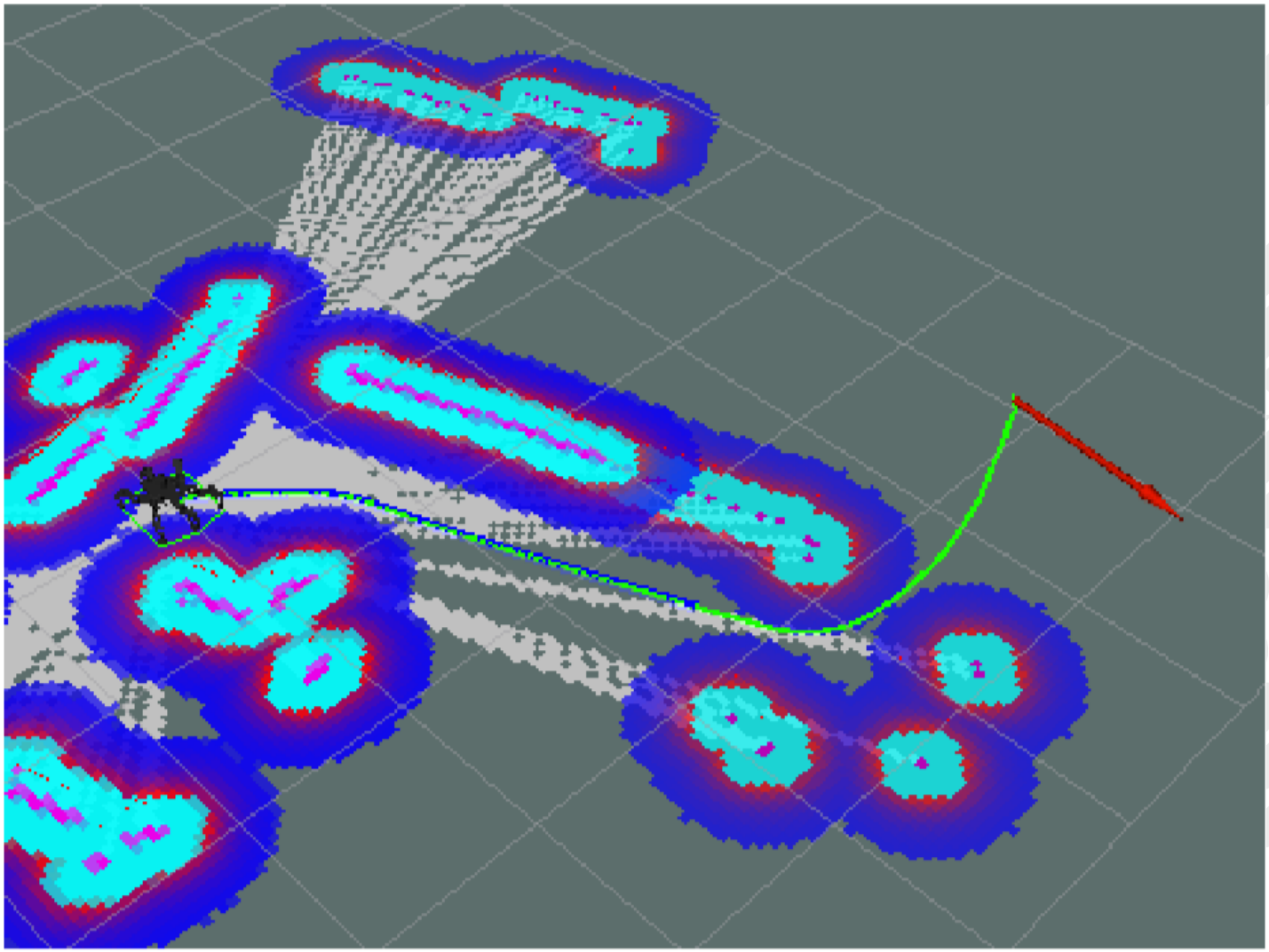
SLAM-Algorithmen

- RTAB-Map
  - Benötigt:
    - RGB-D Kamera
    - Odometrie-Daten
  - Erstellt eine 2D-Rasterkarte
    - Aufgrund der Performance
  - CPU-Auslastung durchschnittlich bei 50 Prozent
  - Keine Erstellung der Karte möglich
    - Instabilität des Roboters
    - Keine Übereinstimmung der Aufnahmen

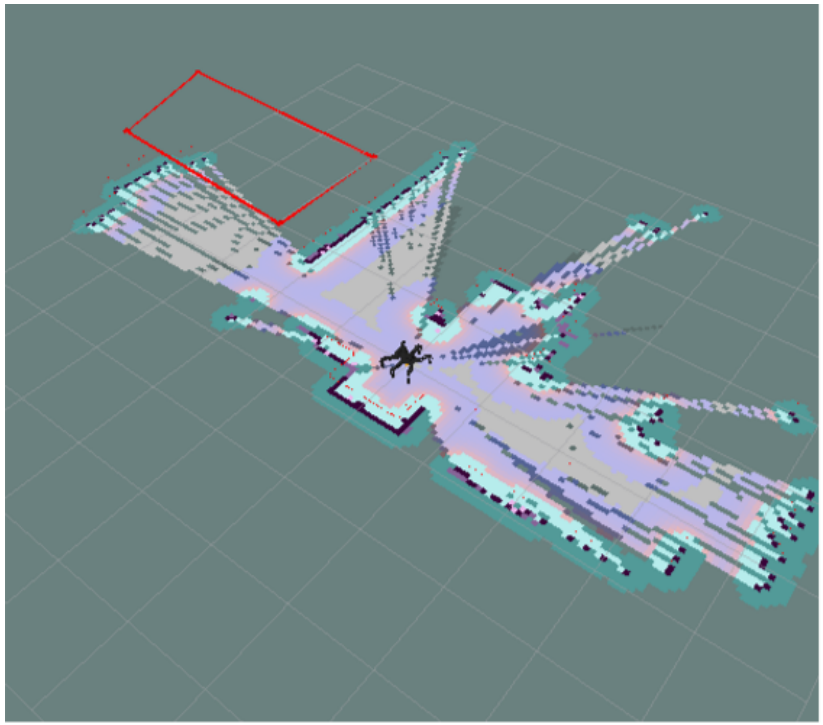
Navigation

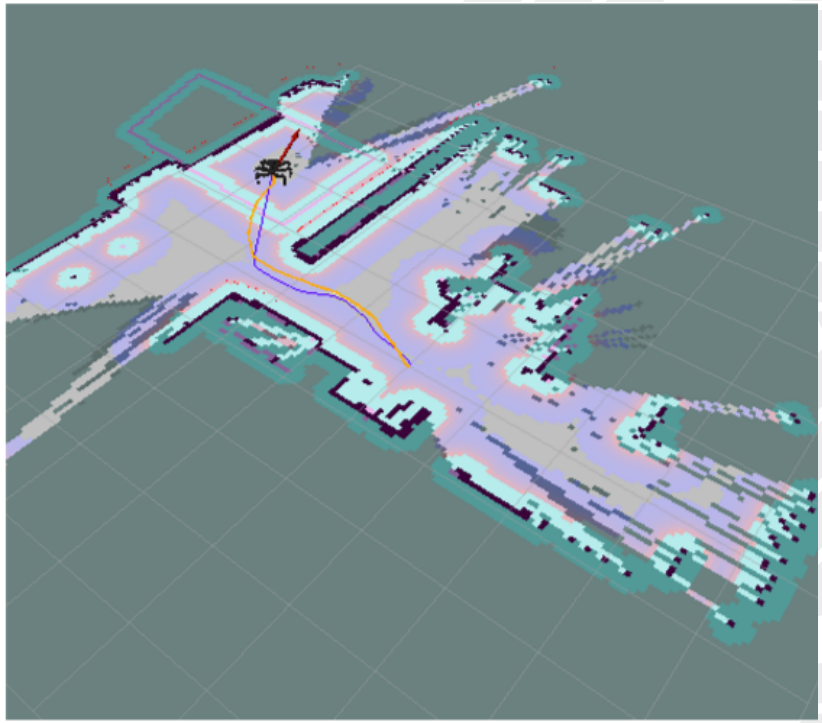
- Erlaubt die selbständige Navigation zu einem Zielpunkt
- Benötigt:
  - SLAM-Algorithmus
  - 2D-Karte
  - Sensordaten
  - Größe des Roboters
- Kostenwertkarte erstellt
- Einbindung des Ultraschallsensors
  - Anpassung der Kostenwertkarte
  - Erkennung von Glasflächen
  - Erkennung von sehr nahen Hindernissen

Navigation



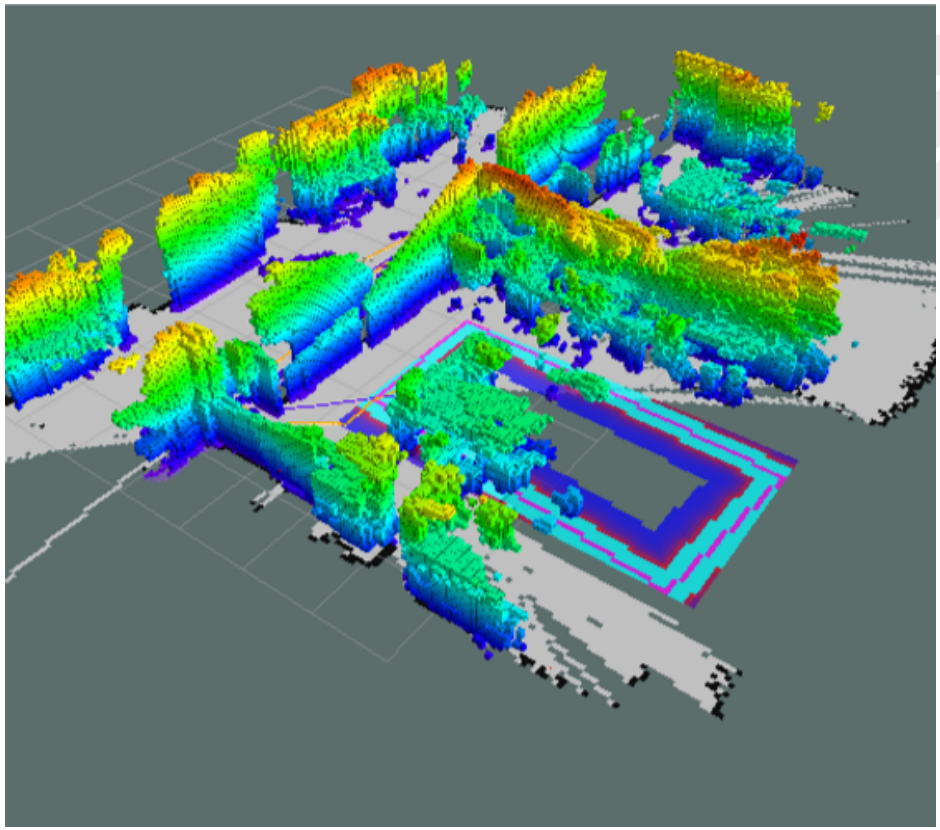
- Navigation ohne Operator
- Zielbereiche können durch ein Rechteck definiert werden
- Kostenwertkarte bestimmt die zu kartografierenden Bereiche





Erstellung einer 3D-Karte

- Darstellung als Punktwolke
- Daten von RGB-D-Kamera
- Große Datenmengen
  - Keine Erkennung bereits vorhandener Punkte
- Keine Farbwerte



Fazit und Ausblick

- Erstellte System ist in der Lage Bereiche autonom zu erkunden
  - Keine Kollision mit Hindernissen
- Sensorfusion erfolgreich
- PhantomX eignet sich als Trägersystem
- GMapping erzielte die besten Ergebnisse
- Bisheriges System nur auf ebenem Gelände

- Verbesserung der Kinematik
- Umsetzung der Laufalgorithmen im FPGA
- Abgrundsensor
- Erweiterung der Beine
- Modularer Systemaufbau

Vielen Dank für Ihre Aufmerksamkeit