

Realisierung des autonomen Indoor Fahrens für das ROS-basierte Robotik-Mappingsystem NAVKA-MSM

Diese Arbeit befasst sich mit der Realisierung des autonomen Indoor Fahrens sowie den Auswirkungen von Parameteränderungen.

Als Hardware wurde hierfür der Volksbot R3-2 und der Velodyne VLP 16 Laserscanner verwendet. Zur Datenübertragung wurden diese jeweils über USB/Ethernet-Kabel mit einem Laptop verbunden.



Gesamtkonfiguration der genutzten Hardware

Grundlegend werden mittels des Laserscanners Umgebungsinformationen aufgenommen, welche an den Laptop übermittelt werden. Dieser übernimmt die Aufgabe befahrbare Trajektorien aus den Laserscan-Daten zu berechnen. Diese werden in Form von Geschwindigkeitskommandos an die Roboterplattform übermittelt.

Um die Berechnung sowie die benötigte Software auf dem Laptop nutzen zu können, ist das sogenannte Robot Operating System, kurz ROS, erforderlich.

Dabei handelt es sich um ein Meta-Betriebssystem, welches das Schreiben von Robotersoftware ermöglicht und verschiedenste Bibliotheken und Werkzeuge bietet.

Hochschule Karlsruhe – Technik und Wirtschaft
Fakultät IMM • Studiengang Geodäsie und Navigation
www.hs-karlsruhe.de

Bearbeiter: Jan Retzlaff

E-Mail-Adresse: reja1021@hs-karlsruhe.de

Referent: Prof. Dr.-Ing. Reiner Jäger

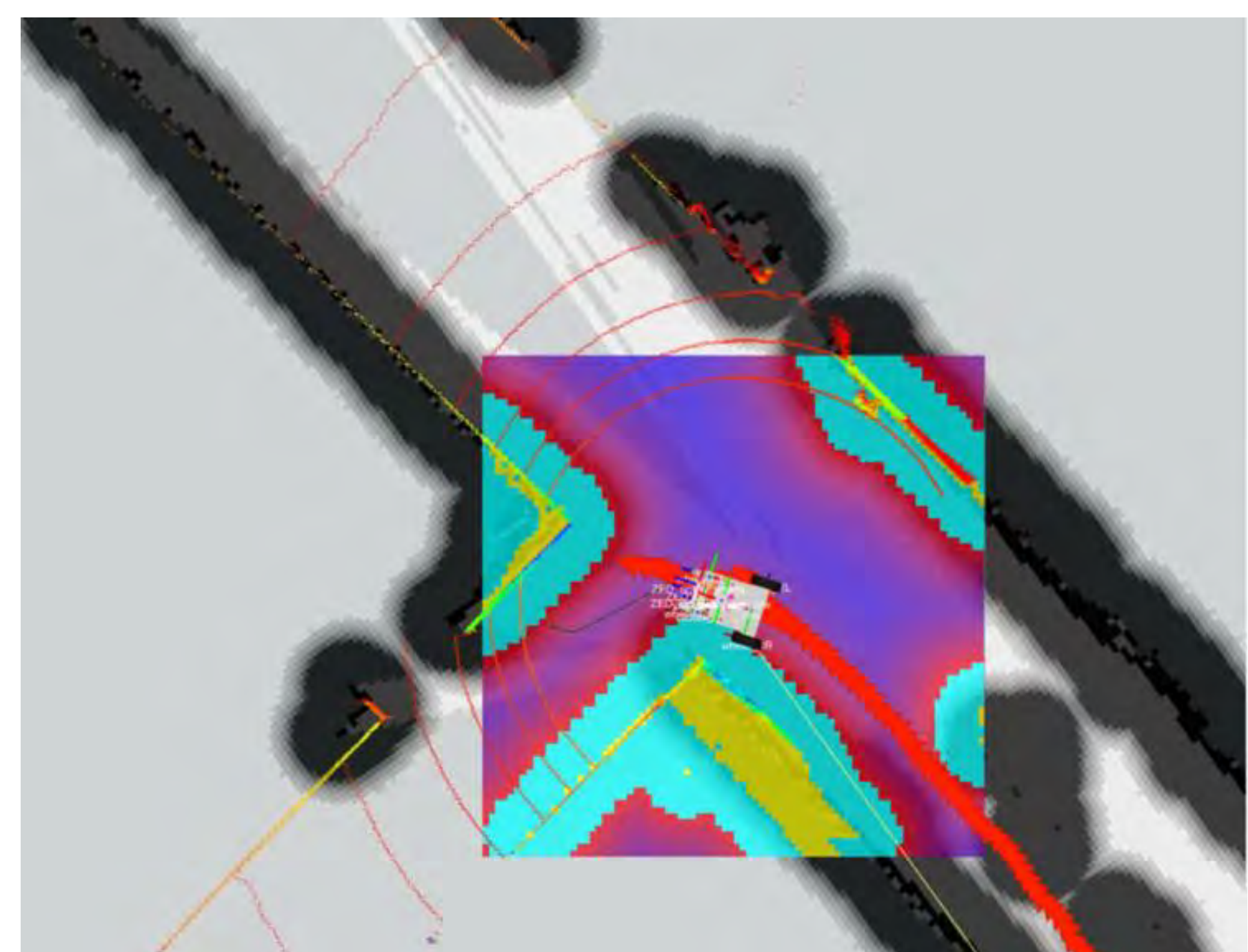
Korreferent: Dipl.-Ing. Julia Diekert

Als Software wird neben den Treibern für Roboterplattform und Laserscanner, ein Programm zur Simultanen Lokalisierung und Kartierung, kurz SLAM und das sogenannte ROS Navigation Stack genutzt.

SLAM übernimmt die Aufgaben die Position des Roboters zu ermitteln und eine Karte der Umgebung zu erstellen.

Das ROS Navigation Stack dient der Navigation des Roboters mit Grundlage einer Karte und Sensorinformationen. Es besteht aus mehreren Komponenten.

Über Kostenkarten werden Hindernisse in Rastern dargestellt und mit, von der Entfernung zum Roboter abhängigen, Kosten versehen.



Visualisierung der globalen/lokalen Kostenkarte

Der globale Bahnplaner berechnet mittels der globalen Kostenkarte, in welcher alle statischen Hindernisse eingezeichnet sind, einen Weg von einem Ausgangs- zu einem Zielpunkt mit möglichst geringen Kosten.

Mittels des lokalen Bahnplaners, welcher auf der lokalen Kostenkarte basiert, können dynamische Hindernisse erfasst und somit vermieden werden. Mittels einer weiteren Komponente, move_base, werden die benötigten Geschwindigkeitskommandos berechnet.