
Abschluss Bericht für das Praktikum Mobile Roboter im WS 17/18

Florian Dreschner, Yassine El Himer, Daniel Klitzke, Robin Weitemeyer

3. Februar 2018

Inhaltsverzeichnis

1	Einleitung	3
1.1	Aufgabenstellung	3
1.2	Ausgangssituation	3
1.3	Systemarchitektur	3
1.3.1	Hardware	3
1.3.2	Software	3
1.4	Arbeitspakete	3
2	State of the Art	4
2.1	Bilderkennung	4
2.2	Bahnplanung & Greifen	4
3	Arbeitsbericht	4
3.1	Bilderkennung	4
3.1.1	Tassenerkennung	4
3.1.2	Turtleboterkennung	6
3.1.3	Automatische Kamerakalibrierung	6
3.2	Bahnplanung & Greifen	6
3.2.1	Bahnplanung	6
3.2.2	Greifen	6
3.3	High-level Steuerung & Kommunikation	6
4	Beschreibung des Gesamtsystems	6
4.1	Bilderkennung	6
4.1.1	Tassenerkennung	6
4.1.2	Automatische Kamerakalibrierung	6
4.2	Bahnplanung & Greifen	6
4.2.1	Bahnplanung	6
4.2.2	Greifen	6
4.3	High-level Steuerung & Kommunikation	6
5	Evaluation & Ausblick	6
5.1	Bilderkennung	6
5.1.1	Tassenerkennung	6
5.1.2	Turtleboterkennung	6
5.1.3	Automatische Kamerakalibrierung	6
5.2	Bahnplanung & Greifen	6
5.2.1	Bahnplanung	6
5.2.2	Greifen	6
5.3	High-level Steuerung & Kommunikation	6
5.4	Gesamtsystem	6

1 Einleitung

1.1 Aufgabenstellung

Ziel des diesjährigen Praktikums Mobile Roboter mit dem Thema „Coffee to go“ war es, eine Tasse Kaffee, welche auf einem Tisch platziert wurde mittels eines Roboterarms zu greifen, auf einem mobilen Roboter zu platzieren und diese, nach Transport durch selbigen, dann mittels eines zweiten Roboterarms wieder auf einen zweiten Tisch zu stellen. Die Aufgabe unserer Gruppe war hierbei, die Tasse auf dem ersten Tisch zu lokalisieren, sie zu greifen und auf dem mobilen Roboter zu platzieren.

1.2 Ausgangssituation

Als Ausgangspunkt für die Realisierung der Aufgabe wurde uns sowohl diverse Hardware als auch Software zu Verfügung gestellt. So stand uns für das Greifen der Tasse ein UR5 Roboter von Universal Robots, welcher bereits auf einem Tisch befestigt wurde. Zur Lokalisierung der Tasse waren eine Kinect bzw. alternativ eine Intel Realsense Kamera verfügbar. Außerdem standen uns für die Entwicklung der Software sowohl vorkonfigurierte Rechner in Poolräumen, als auch ein an den UR5 Roboter angeschlossener Shuttle PC zur Verfügung. Als Grundlage für die zu entwickelnde Software dienten ROS Indigo bzw. Kinetic.

1.3 Systemarchitektur

1.3.1 Hardware

1.3.2 Software

1.4 Arbeitspakete

Die Lösung der gestellten Aufgabe haben wir in folgende Arbeitspakete unterteilt:

Hauptaufgabe	Arbeitspaket	Zuständigkeit
Bildererkennung		
	Tassendetektion mit FCN	Daniel Klitzke, Yassine El Himer
	Tassendetektion Segmentierung + Neuronale Netze	Daniel Klitzke
	Tassendetektion SVM + HOG Features	Daniel Klitzke
	Segmentierung der Tasse in Punktwolke	Daniel Klitzke
	Erkennung der Tassenorientierung	Daniel Klitzke
	Turtlebotdetektion SVM + HOG Features	Yassine El Himer
	Automatische Kamerakalibrierung	Daniel Klitzke
Bahnplanung & Greifen		
	Aufsetzen der Simulation	Robin Weitemeyer
	Modellierung der Roboterumgebung	Robin Weitemeyer
	Bahnplanung mit MoveIt	Robin Weitemeyer
	Modellierung eines Greifers	Florian Dreschner
High-level Steuerung & Kommunikationsschnittstellen		Yassine el Himer, Florian Dreschner

2 State of the Art

2.1 Bildererkennung

2.2 Bahnplanung & Greifen

3 Arbeitsbericht

3.1 Bildererkennung

3.1.1 Tassenerkennung

Allgemein Für die Tassenerkennung wurden keinerlei ROS Pakete verwendet. Stattdessen wurden die einzelnen Komponenten in Python selbst implementiert. Die am häufigsten zur Hilfe genommenen Bibliotheken sind unter anderem:

- NumPy (Effizienter Umgang mit Matrizen und Vektoren)

-
- scikit-image (Bildverarbeitung)
 - scikit-learn (Algorithmen für Maschinelles Lernen)
 - Keras (High-level API für die Erstellung von Neuronalen Netzen)
 - Matplotlib (Visualisierung)

Detektion der Tasse in den Bilddaten Für die Detektion der Tasse in den Bilddaten wurden eine Reihe von Ansätzen getestet, bevor die finale Lösung feststand.

Zunächst war der Plan die Tasse mittels eines Fully Convolutional Neural Network aus den Bilddaten zu segmentieren. Als Basisstruktur wurde hierfür die Architektur des Resnet50 Netzes verwendet. Die Eingabe für das Netz stellt ein RGB Bild dar und es liefert als Ausgabe zwei Feature Maps, welche die Wahrscheinlichkeiten für die Zugehörigkeiten der Pixel des Eingabebildes zu den Klassen „Tasse“ bzw. „Keine Tasse“ enthalten. Trainiert wurde auf den Microsoft COCO Datensatz unter Nutzung von GPU Kapazitäten bei Amazon Webservices. Der Ansatz wurde letztendlich verworfen, da es sich herausstellte dass das Netz zu überdimensioniert für die Segmentierung einer Klasse aus einem RGB Datensatz ist und nicht genügend Trainingsdaten zur Verfügung standen.

Der zweite getestete Ansatz basierte auf der Segmentierung des Bildes in eine Menge von Segmenten sowie einer Klassifizierung durch ein Neuronales Netz. Das Bild wird hierfür zunächst mittels des SLIC Algorithmus in eine Menge von Segmenten zerlegt. Diese Segmente werden dann unter Einbeziehung eines gewissen Kontextfensters dem Neuronalen Netz zur Klassifikation übergeben. Jedem Segment wird entweder als „Tasse“ oder „Keine Tasse“ klassifiziert.

Der dritte gewählte Ansatz wurde letztendlich auch in das fertige System übernommen und basiert auf einer SVM welche Bildsegmente, die mithilfe eines Sliding Window Ansatzes gewonnen werden klassifiziert. Details zu diesem Ansatz werden im Abschnitt 4.1.1 näher erläutert.

Segmentierung der Tasse in der Punktwolke Nachdem die Tasse in den Bilddaten detektiert wurde wird sie mittels eines Clustering Algorithmus aus den Bilddaten segmentiert. Hierbei traten keinerlei Komplikationen auf. Der Ansatz wird ebenfalls in 4.1.1 beschrieben.

Erkennung der Tassenorientierung Hier wurden zwei verschiedene Ansätze implementiert wovon letztendlich einer für das finale System ausgewählt wurde. Der erste Ansatz basiert auf dem RANSAC Algorithmus. Zunächst wird die Punktwolke der Tasse auf die x-y-Ebene projiziert und deren Zentrum bestimmt. Anschließend wird in jeder Iteration des Ransac Algorithmus ein zufälliger Punkt ausgewählt und eine Linie durch selbigen und das Zentrum der Tasse gelegt. Es wird die Konfiguration mit den meisten Inliern als Richtung des Tassengriffs ausgewählt. Es stellte sich hierbei heraus, dass die zum Tassengriff gehörigen Punkte im Vergleich zu den restlichen Punkten so wenig ins Gewicht fallen dass so keine zuverlässige Schätzung möglich war. Der zweite Ansatz basiert auf

dem Local Outlier Factor Schätzer und erwies sich als deutlich robuster als der zuvor beschriebene Ansatz. Er wird in 4.1.1 näher beschrieben.

3.1.2 Turtleboterkennung

3.1.3 Automatische Kamerakalibrierung

3.2 Bahnplanung & Greifen

3.2.1 Bahnplanung

3.2.2 Greifen

3.3 High-level Steuerung & Kommunikation

4 Beschreibung des Gesamtsystems

4.1 Bilderkennung

4.1.1 Tassenerkennung

4.1.2 Automatische Kamerakalibrierung

4.2 Bahnplanung & Greifen

4.2.1 Bahnplanung

4.2.2 Greifen

4.3 High-level Steuerung & Kommunikation

5 Evaluation & Ausblick

5.1 Bilderkennung

5.1.1 Tassenerkennung

5.1.2 Turtleboterkennung

5.1.3 Automatische Kamerakalibrierung

5.2 Bahnplanung & Greifen

5.2.1 Bahnplanung

5.2.2 Greifen

5.3 High-level Steuerung & Kommunikation

5.4 Gesamtsystem