



# Cutting Languages Down to Size

## Student Project

at the Cooperative State University Baden-Württemberg Stuttgart

by

**Nahku Saidy and Hanna Siegfried**

08.06.2020

**Time of Project**

nothing

**Student ID, Course**

8540946, XXXXX; TINF17ITA

**Advisors**

Prof. Dr. Stephan Schulz and Geoff Sutcliffe

# Contents

<b>Acronyms</b>	<b>I</b>
<b>List of Figures</b>	<b>II</b>
<b>List of Tables</b>	<b>III</b>
<b>Listings</b>	<b>IV</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Statement and Goals . . . . .	1
1.2 Structure of the Report . . . . .	1
<b>2 Background and Theory</b>	<b>2</b>
2.1 TPTP Language . . . . .	2
2.2 XXX . . . . .	2
<b>3 Concept</b>	<b>3</b>
3.1 Übersicht . . . . .	3
<b>4 Implementation</b>	<b>4</b>
4.1 Spiellogik . . . . .	4
4.2 Erkennung des Spielendes . . . . .	4
<b>5 Validation</b>	<b>5</b>
<b>6 Conclusion</b>	<b>6</b>
<b>Bibliography</b>	<b>i</b>

# Acronyms

**AABB**     Axis-Aligned Bounding Box

# List of Figures

# List of Tables

# Listings

# 1 Introduction

## 1.1 Problem Statement and Goals

Die Aufgabe bestand in der Entwicklung einer Beispielanwendung unter Verwendung zweier Aspekte aus der Vorlesung. Wir haben uns für die Entwicklung eines 4-gewinnt-Spiels auf Basis der Chat-Anwendung entschieden. -docker -mongo.db -docker-compose -node.js

## 1.2 Structure of the Report

## **2 Background and Theory**

### **2.1 TPTP Language**

### **2.2 XXX**



# **3 Concept**

## **3.1 Übersicht**

# 4 Implementation

## 4.1 Spiellogik

## 4.2 Erkennung des Spielendes

Es gibt zwei Szenarien, unter denen das Spiel zu Ende ist. Entweder gewinnt ein Spieler oder das Spiel endet unentschieden. Unentschieden ist das Spiel, wenn das Spielfeld komplett gefüllt ist, aber kein Spieler eine Reihe aus vier Steinen aufbauen konnte. Bei einem Gewinn sind drei verschiedene Szenarien zu unterscheiden. Eine Reihe aus vier Spielsteinen des gleichen Spielers kann horizontal, vertikal oder diagonal auftreten. Wenn die Reihe diagonal gebildet wird, kann noch zwischen von links unten nach rechts oben und von rechts unten nach links oben unterschieden werden. Alle diese Fälle müssen geprüft werden.

## 5 Validation

# 6 Conclusion

Das Ziel dieser Projektarbeit war die Entwicklung eines Konzepts für eine **LiDAR!** (**LiDAR!**)-basierte Objekterkennung und eine prototypische Implementierung dieser. Dazu wurden bereits existierende Clustering-Verfahren diskutiert und deren Vor- und Nachteile für die Verwendung im Kontext dieser Projektarbeit dargestellt. Außerdem wurden konkrete Probleme des **DBSCAN!** (**DBSCAN!**)-Algorithmus mit dem Clustering von **LiDAR!**-Daten herausgearbeitet. Diesen wurde durch eine Erweiterung des Algorithmus mit einer distanzabhängigen Berechnung des Parameters  $\epsilon$  entgegengewirkt (Kapitel ??).

Die erkannten Objekte werden mit eindeutigen IDs und Axis-Aligned Bounding Box (**AABB**)s im Sensordatenbild visualisiert.

Zusätzlich wurde eine Tracking-Funktion entwickelt, die das Tracken eines einzelnen Objekts mittels des Kalman-Filter ermöglicht und den Bewegungsverlauf zusätzlich zu den Clustering-Ergebnissen im Sensordatenbild darstellt. Dieses Konzept wurde mittels Matlab implementiert. Dabei wurde eine Anpassung des **DBSCAN!**-Algorithmus vorgenommen, welche die Laufzeit reduziert, da die Analyse eines Messzyklus sonst ca. 70 - 100 Sekunden gedauert hätte (siehe Kapitel ??). Nach der Verbesserung ergaben sich je nach Messumgebung Laufzeiten von ca. 10 - 50 Sekunden.

Somit konnten die zu Beginn in Kapitel 1.1 definierten Ziele erreicht werden und zusätzlich die Tracking-Funktion entwickelt werden, die es ermöglicht, nicht nur einen Messzyklus zu analysieren, sondern die zukünftige Position eines Objekts anhand der Analyse vorheriger Messzyklen zu schätzen.

## Outlook

Es gibt einige Bereiche, in denen die Funktion der Objekterkennung erweitert und verbessert werden kann.

Es besteht die Möglichkeit, eine Sensorbewegung in die Berechnungen miteinzubeziehen. Dies bietet sich an, wenn der Sensor auf einem beweglichen Objekt montiert ist. Bisher wurde die Annahme eines stationären Sensors getroffen. Mit einem bewegten

Sensor müssten die Koordinaten, die immer relativ zum Sensor aufgenommen werden, in ein Welt-Koordinatensystem umgerechnet werden. Außerdem könnte die Objekterkennung an Echtzeitanforderungen angepasst werden. Durch die hohe Anzahl an Punkten, die sich in einer Punktwolke befinden, ist die Analyse dieser rechenintensiv. Deshalb müsste voraussichtlich eine Vereinfachung der Sensordaten vor der Analyse vorgenommen werden, um die Ausführungszeit zu reduzieren. Aktuell wird der Matlab-Programmcode für jede Ausführung der Objekterkennung neu interpretiert. Würde der Code der Objekterkennung in ein ausführbares Programm übersetzt werden, so ließe sich die Ausführungszeit weiter reduzieren.

In Bezug auf die Tracking-Funktionalität ist es denkbar, diese um ein Tracking von mehreren Objekten zu erweitern. Dazu müsste eine neue Strategie zur Data-Association entwickelt werden. Es ist dabei zu beachten, dass die Tracking-Funktion auf das Tracken von Personen ausgerichtet ist. Sollen auch Objekte anderer Art getrackt werden, so muss auch ggf. eine Anpassung des Bewegungsmodells für den Kalman-Filter in Betracht gezogen werden.

Neben der Funktion des Trackens mehrerer Objekte könnte das Tracking um die Funktion eines automatischen Starts des Trackings erweitert werden. Bisher wählt der Benutzer aus, welches Objekt getrackt werden soll. Durch eine Analyse, welche Objekte beweglich sind, könnte dieser Prozess automatisiert werden, sodass keine Benutzereingabe mehr erforderlich ist.

# Bibliography

## Publikationen

- [1] Peter Knoll. *Fahrstabilisierungssysteme und Fahrerassistenzsysteme*. Wiesbaden: Vieweg+Teubner Verlag, 2010. ISBN: 9783834813145.