



# Entwicklung eines Indoor-Assistenzsystems für Multicopter mit Hilfe von Monocularer Tiefenbild Rekonstruktion

## Studienarbeit

Studiengang Angewandte Informatik  
Duale Hochschule Baden-Württemberg Karlsruhe

von  
Christoph Meise, Max Lenk

Datum der Abgabe:	15.05.2017
Bearbeitungszeitraum:	2 Semester
Matrikelnummern und Kurse:	4050853, 3460046, TINF14B2, TINF14B1
Betreuer:	Markus Strand

Copyright Vermerk:  
All rights reserved. **Copyright.**

© 2016

# Ehrenwörtliche Erklärung

“Ich erkläre ehrenwörtlich:

1. dass ich meine Projektarbeit mit dem Thema  
*Entwicklung eines Indoor-Assistenzsystems für Multicopter*  
ohne fremde Hilfe angefertigt und selbstständig verfasst habe;
2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe;
3. dass ich meine Projektarbeit bei keiner anderen Prüfung vorgelegt habe.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.”

Walldorf, der 16.09.2016

---

CHRISTOPH MEISE, MAX LENK

**Restriction notice**

This report contains confidential information of

SAP SE

Dietmar-Hopp-Allee 16

69190 Walldorf, Germany

It may be used for examination purposes as a performance record of the department of Applied Computer Science at the Cooperative State University Karlsruhe. The content has to be treated confidentially.

Duplication and publication of this report - as a whole or in extracts - is not allowed.

**Sperrvermerk**

Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anders lautende Genehmigung der Ausbildungsstätte vorliegt.

# Abstrakt

# Inhaltsverzeichnis

<b>Ehrenwörtliche Erklärung</b>	<b>I</b>
<b>Abbildungsverzeichnis</b>	<b>VI</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Aufbau . . . . .	3
<b>2 Grundlagen</b>	<b>4</b>
2.1 AR.Drone 2.0 . . . . .	4
2.2 ROS . . . . .	5
2.3 Simulation . . . . .	5
2.4 Fuzzylogik . . . . .	5
2.5 Kinect . . . . .	5
<b>3 Software Architektur</b>	<b>6</b>
3.1 Anforderungen . . . . .	6
3.2 Überblick . . . . .	7
3.3 Implementation . . . . .	7
3.3.1 Semi-Direct Monocular Visual Odometry - SVO . . . . .	8
3.3.2 Regularized Monocular Depth Estimation - REMODE . . . . .	9
3.3.3 Kamerakalibrierung . . . . .	9

3.3.4	Featureerkennung . . . . .	10
3.3.5	Kinect . . . . .	10
3.3.6	Architektur/Aufbau . . . . .	10
3.3.7	Die einzelnen ROS Nodes . . . . .	10
<b>4</b>	<b>Evaluation</b>	<b>11</b>
4.1	Ergebnis . . . . .	11
4.2	Ausblick . . . . .	11
	<b>Literaturverzeichnis</b>	<b>12</b>

# Abbildungsverzeichnis

# 1 Einleitung

Autonomes Fahren, Machine Learning und Industrie 4.0. Hinter diesen aktuellen Themen steckt das Ziel, Abläufe und Zusammenhänge kontinuierlich zu automatisieren und für den Menschen zu vereinfachen. Das Thema der Automation kann vor allem in der Interaktion zwischen Mensch und Maschine hilfreich sein. Auf Grund der geringen Kosten und der hohen Anwendungsvielfalt bieten sich vor allem Drohnen für die Forschung zu Automation in der Robotik an.

Mit Modellen ab 30 und bis zu mehreren tausend Euro gibt es Ausführungen für nahezu jeden Anwendungsfall. Meist mit mehreren Sensoren und Kameras ausgestattet, stellen sie nicht nur Spielzeug dar, sondern sind essentiell für reale Anwendungsgebiete.

So werden heute schon Drohnen genutzt, um Katastrophengebiete und Kriegsregionen aus sicheren Standorten aufzuklären, oder um die Feuerwehr bei der Branderkundung und Menschensuche zu unterstützen.

Natürlich können sie auch genutzt werden, um alltägliche Probleme zu lösen, wie die schnelle und direkte Lieferung von Paketen.

Da diese große Zahl an Drohnen nicht mehr manuell gesteuert werden kann, müssen sich diese größtenteils autonom bewegen. Dabei treten eine Vielzahl von komplexen Problemen auf, wie das Zurechtfinden in einem unbekannten Raum und die Objekterkennung.

Außerdem ist bei Fluggeräten das Problem, dass die verwendete Ausrüstung ten-



denziell leicht und klein sein muss, damit die Flugeigenschaften nicht eingeschränkt werden bzw. die Drohne nicht zu groß wird.

Im Umfang dieser Arbeit soll eine Vorstufe zum autonomen Fliegen betrachtet werden: ein Assistenzsystem für den manuellen Flug. Dies ist Vergleichbar mit den Assistenzsystemen in PKWs, bei denen ein Tempomat, Licht- und Regensensoren oder Spurhalteassistenten den Fahrer unterstützen, jedoch das Fahren nicht abnehmen.

### 1.1 Motivation

Das Ziel besteht darin, dass die Drohne aktiv die Umgebung auswertet und dabei Objekte wie Türen, oder Wände erkennt. Der Unterschied zu bereits bestehenden Projekten in diesem Themengebiet besteht darin, dass nur eine einzelne monokulare Kamera verwendet werden soll, anstatt das externe Tiefenbildkameras montiert werden.

Dadurch trifft man auf eine Vielzahl von komplexen Problemen, welche im weiteren Verlauf dieser Arbeit dargestellt werden. Auf der anderen Seite könnte man dadurch teure Hardware sparen und somit auch auf andere Projekte anwenden.

Anhand der Tiefenbilder soll die Drohne anschließend in der Lage sein, entgegen der Entscheidungen des Nutzers zu fliegen, um somit beispielsweise Kollisionen zu vermeiden.

Da das Problem nur durch die integrierte Hardware gelöst werden soll, kann nur auf einfache mittelmäßige Kameras zurückgegriffen werden, welche nach Vorne und zum Boden gerichtet sind. Dies soll die Drohne weiterhin auch in einer vollständig simulierten Umgebung können.

## 1.2 Aufbau

Diese Studienarbeit besteht aus 3 Kapiteln. Im ersten Abschnitt der Arbeit sollen die Grundlagen erklärt werden. Zuerst wird dabei die genutzte Drohne und deren Spezifikationen dargestellt.

Anschließend wird das Software Framework Roboter Operating System (*ROS*) eingeführt, welches ein Hauptbestandteil der Projektarchitektur ausmacht.

Im weiteren Verlauf wird dann die Simulationsumgebung beschrieben, da das Projekt sowohl unter realen Bedingungen, als auch in einer simulierten Welt soll. Im letzten Teil dieses Kapitels wird die später verwendete Fuzzylogik erklärt.

Der zweite Abschnitt dieser Arbeit umfasst die Software Architektur. Hierbei sollen sowohl die Anforderungen an das Projekt, als auch die Implementierungstechnischen Spezifikationen dargelegt werden. Im dritten Teil besteht aus dem erzielten Ergebnis, sowie aus dem Ausblick für weitere Betrachtungen dieser Problematik.

## 2 Grundlagen

### 2.1 AR.Drone 2.0

Bei der AR.Drone 2.0 handelt es sich um einen ferngesteuerten Quadrocopter des französischen Herstellers Parrot SA. [1] Die Drohne ist standardmäßig steuerbar mit einer mobilen Applikation für Android und iOS Geräte. Dafür baut sie ein WLAN Netzwerk auf, mit dem sich die Geräte verbinden können. Zur Steuerung stellt die AR.Drone ein Interface zur Verfügung, mit dem sie ferngesteuert werden kann.

Im Umfang der Studienarbeit wird die aktuellste Version der AR.Drone 2.0 verwendet. Diese zeichnet sich unter Anderem durch eine Frontkamera mit einer Auflösung von  $1280 \times 720$  Pixeln und einer Bildrate von 30 fps aus. Weiterhin ist Sie mit einer zum Boden gerichteten QVGA Kamera ausgerüstet, welche 60 Bilder pro Sekunde aufnimmt.

Die Drohne orientiert sich beim Fliegen mit Hilfe einer Vielzahl von Sensoren. Dazu gehören ein dreiaxsiges Gyroskop und ein Magnetometer. Weiterhin nutzt sie Beschleunigungs-, Ultraschall- und Luftdrucksensoren.

Der Grund für die Wahl der Drohne ist vor allem der vergleichsweise niedrige Preis von ca. 200€ und der starken Verbreitung in der Forschung. Dadurch gibt es bereits eine Vielzahl von Projekten, die dazu führen, dass die Drohne und das dazugehörige Interface zu einem großen Umfang fehlerfrei funktionieren.

Weiterhin gibt es schon ROS Nodes (siehe 2.2) und konfigurierte Modelle in Simulationsumgebungen, welche die Arbeit an dem Projekt beschleunigen.

## 2.2 ROS

## 2.3 Simulation

## 2.4 Fuzzylogik

## 2.5 Kinect

## 3 Software Architektur

### 3.1 Anforderungen

Im Folgenden sollen die Anforderungen an die Studienarbeit festgelegt werden.

Ein Hauptbestandteil der Arbeit besteht darin, ein Vorgängerprojekt mit einer AR.Drone 2.0 in das ROS Framework zu portieren. Dies beinhaltet eine Modularisierung der Projektbestandteile in ROS Nodes. Ziel soll sein, den Quadrocopter mit Hilfe einer Kinect und Gestenerkennung steuern zu können. Anzumerken ist, dass ROS nur unter UNIX basierten Betriebssystemen läuft und das bestehende Projekt Libraries verwendet, welche nur unter Windows verfügbar sind.

Eine weitere Vorgabe besteht darin, dass man neben der realen Drohne jegliche Funktionalität auch in einer Simulation laufen soll.

Somit kann man für Präsentationen, in denen der Flug einer Drohne nicht möglich ist, den Quadrocopter in einer frei gestaltbaren simulierten Umgebung fliegen lassen.

Ein weiterer Bestandteil dieser Arbeit besteht darin, Ansätze und Limitationen der Implementierung eines Assistenzsystems zu testen und zu bewerten. Dafür soll ein externes Projekt, zur Gewinnung von Tiefenbildern aus einer monokularen Kamera, in die Projektumgebung integriert werden. Hierbei soll es wiederum möglich sein, dass die Videostream sowohl von der realen, als auch von der simulierten Drohne gesendet werden kann.

Es soll dabei ermittelt werden, ob die Nutzung der Software für den Anwendungs-

zweck praktikabel und sinnvoll ist.

## **3.2 Überblick**

## **3.3 Implementation**

### 3.3.1 Semi-Direct Monocular Visual Odometry - SVO

Eine Grundanforderung an das Projekt ist die Nutzung einer nicht modifizierten AR.Drone. Dadurch entsteht die Problematik, dass man keine Tiefenbildkamera nutzen kann, um in Echtzeit Tiefenbilder zu generieren. Die Drohne ist lediglich mit einer monokularen <sup>1</sup> Frontkamera ausgestattet.

Um Tiefeinformationen aus den Bildern einer solchen Kamera zu gewinnen, wird eine Szene aus verschiedenen Perspektiven aufgenommen. Anschließend gibt es unterschiedliche Ansätze um aus den aufeinanderfolgenden Bildern Kamerapositionen und Umgebungsstrukturen zu ermitteln.

Der aktuelle Standard sind Feature basierte Methoden. Diese versuchen die wichtigsten Merkmale eines Bildes, die Features, zu extrahieren. Mit Hilfe von Feature Deskriptor Algorithmen werden Vektoren mit Informationen zu invarianten Features berechnet, welche als Fingerabdruck für diese Merkmale fungiert.

Aufeinanderfolgende Bilder werden dann mit Hilfe dieser Feature Deskriptoren abgeglichen und sowohl Kamerabewegungen, als auch Strukturen werden rekonstruiert. Zur Optimierung sind abschließend die ermittelten Kamerapositionen anzugleichen. Dies geschieht mit Hilfe von Algorithmen zur Minimierung des Reprojektionsfehlers. <sup>2</sup>

Ein weiterer Ansatz ist die direkte Methode. Hierbei werden die Features nicht über Deskriptor Algorithmen bestimmt, sondern das Problem wird über die Intensitäten der Pixel gelöst. Bei einem Graustufenbild kann entspricht diese Intensität der Helligkeit von Bildbereichen.

Somit kann bei der Rekonstruktion im Gegensatz zum Feature basierten Ansatz auch die Richtung der Gradienten von Intensitäten genutzt werden. Dadurch funktioniert diese Methode auch bei Bildern mit sehr wenig Textur, Bewegungsunschärfe und

---

<sup>1</sup>Monokular ist die Bezeichnung für Kameras mit einer einzelnen Linse

<sup>2</sup>Reprojektionsfehler sind geometrische Fehler die im Zusammenhang zwischen abgebildeten und berechneten Bildpunkten entstehen.

fehlerhaftem Kamerafokus.

Das für diese Arbeit relevante Vorgehen kombiniert die Vorteile der beschriebenen Methoden. Die Semi direkte Odometrie verwendet einen Algorithmus der ebenfalls auf Feature Zusammenhängen basiert. Diese werden jedoch implizit direkten Bewegungsabschätzung bezogen, anstatt explizit durch Algorithmen mit Feature Deskriptoren berechnet zu werden.

Dadurch müssen Features nur extrahiert werden, wenn diese noch nicht auf einem der vorherigen Bildern vorhanden war. Insgesamt ist dieser Ansatz somit schnell, da wenig Berechnungen pro Bild stattfinden und auf Grund Verwendung der Intensitätsgradienten sehr genau und robust.

Diese Eigenschaften sind für die Anforderungen der Studienarbeit essentiell, da die Drohne sich sehr schnell bewegen kann und somit in möglichst kurzer Zeit neue Bilder auswerten muss. Das beschriebene Verfahren minimiert damit die Auswirkungen der typischen Probleme von Drohnen. Diese sind vorwiegend die niedrige Texturierung, welche in Innenräumen auftreten kann, sowie kameraspezifische Probleme wie Bewegungsunschärfe und der Verlust des Kamerafokus.

### 3.3.2 Regularized Monocular Depth Estimation - REMODE

### 3.3.3 Kamerakalibrierung

Ein Grundproblem der Bildverarbeitung ist die Verzerrung des Bildes. Da aus eingehenden Bildern Features erkannt werden sollen und aus diesen darüber hinaus Abstände berechnet werden, ist es essentiell, dass die Kamera richtig kalibriert ist.

Hierbei unterstützt SVO drei Kamera Modelle: ATAN, Pinhole und Ocam. [2]

Das ATAN Model basiert auf dem *Field of View* (FOV) Verzerrungsmodell.



**3.3.4 Featureerkennung**

**3.3.5 Kinect**

**3.3.6 Architektur/Aufbau**

**3.3.7 Die einzelnen ROS Nodes**

## 4 Evaluation

### 4.1 Ergebnis

### 4.2 Ausblick

## Literaturverzeichnis

- [1] "Parrot AR.Drone." [https://de.wikipedia.org/wiki/Parrot\\_AR.Drone](https://de.wikipedia.org/wiki/Parrot_AR.Drone), 2016.  
visited: 01.22.2017.
- [2] Robotics and Perception Group, "Camera Calibration." [https://github.com/uzh-rpg/rpg\\_svo/wiki/Camera-Calibration](https://github.com/uzh-rpg/rpg_svo/wiki/Camera-Calibration), 2014. visited: 02.19.2017.