

Deutsches Institut für Luft- und Raumfahrt e.V.

Institut für Flugsystemtechnik

Abteilung für Unbemannte Luftfahrzeuge

Universität der Bundeswehr München

Fakultät für Elektrotechnik und Technische Informatik

Wissenschaftliche Einrichtung 6 - Informationstechnik

# Masterarbeit

Relative extrinsische Kalibrierung von bildgebenden  
Umweltsensoren



Prüfer:	Prof. Dr. rer. nat. Harald GÖRL
Betreuer:	Dipl.-Wirt.-Inf. Stefan KRAUSE

Autor:	B.Eng. Michael RIEDEL
Matrikelnummer:	1100582
Datum der Abgabe:	30.08.2015

## **Selbstständigkeitserklärung**

Der Verfasser erklärt, dass er die vorliegende Arbeit selbstständig, ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt hat. Die aus fremden Quellen (einschließlich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind ausnahmslos als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

---

Neubiberg, 30.08.2015

## **Danksagung**

Danke an alle

## **Abstract**

Blubbeldiblubd

# Inhaltsverzeichnis

	Seite
<b>Abkürzungsverzeichnis</b>	<b>i</b>
<b>Abbildungsverzeichnis</b>	<b>ii</b>
<b>Tabellenverzeichnis</b>	<b>iii</b>
<b>1. Einleitung</b>	<b>1</b>
<b>2. Grundlagen</b>	<b>3</b>
2.1. Die verwendeten Koordinatensysteme . . . . .	3
2.1.1. Das Sensorkoordinatensystem . . . . .	3
2.1.2. Das Trägerkoordinatensystem . . . . .	4
2.1.3. Das Weltkoordinatensystem . . . . .	4
2.2. Koordinatentransformation nach Denavit-Hartenberg . . . . .	4
2.3. Sensortechnik . . . . .	5
2.3.1. Lasersanner . . . . .	5
2.3.2. Inertiale Messeinheiten . . . . .	5
2.3.3. Systeme zur globalen Positionsbestimmung . . . . .	6
2.3.4. Systemfehler . . . . .	6
2.4. Algorithmen zur Transformation . . . . .	6
2.4.1. ICP der PCL . . . . .	6
<b>3. Kalibrierung von LiDAR-Sensoren zu Inertialen Messeinheiten</b>	<b>7</b>
3.1. Problemanalyse . . . . .	8

3.2. Möglichkeiten zur Kalibrierung . . . . .	8
3.2.1. Im Sensorkoordinatensystem . . . . .	9
3.2.2. Im Weltkoordinatensystem . . . . .	11
<b>4. Validierung</b>	<b>14</b>
4.1. Einsatzgebiet . . . . .	14
4.2. Aufbau . . . . .	14
4.3. Ablauf . . . . .	14
4.4. Ergebnisse . . . . .	14
4.5. Gesamtergebnis . . . . .	14
<b>5. Fazit und Ausblick</b>	<b>15</b>
<b>Anhang</b>	<b>I</b>
A. Reproduktion der Forschung . . . . .	I
A.1. Entwicklungsumgebung . . . . .	I
A.2. Interessante Links . . . . .	II

## Abkürzungsverzeichnis

<b>6DoF</b>	six degrees of freedom
<b>ARTIS</b>	Autonomous Rotorcraft Testbed for Intelligent Systems
<b>DLR</b>	Deutsches Institut für Luft- und Raumfahrt
<b>GPS</b>	Global Positioning System
<b>ICP</b>	Incremental Closest Point
<b>IMU</b>	Inertial Measurement Unit
<b>LiDAR</b>	Light Detection and Ranging
<b>UA</b>	Unmanned Aircraft
<b>UniBwM</b>	Universität der Bundeswehr München

## Abbildungsverzeichnis

3.1. Kalibrierung im Sensorkoordinatensystem . . . . .	9
3.2. Kalibrierung im Sensorkoordinatensystem mit Bewegungskorrektur . .	10
3.3. Kalibrierung im Weltkoordinatensystem . . . . .	12
3.4. Kalibrierung im Weltkoordinatensystem mit Bewegungskorrektur . . .	13

## Tabellenverzeichnis

3.1. Bezeichnung der Winkel gemäß DIN 9300 / ISO 1151-2:1985 . . . . .	8
1. Notwendige Programme zur Erstellung der Arbeit und Ihrer Abhängigkeiten	I



# 1. Einleitung

In den letzten Jahren hat die allgemeine Akzeptanz im Bereich des autonomen Fahrens stark zugenommen. Namhafte Hersteller wie Audi und BMW bieten bereits hochautomatisierte Fahrzeuge in ihrem Portfolio an, die den Fahrer in alltäglichen Situationen entlasten. Das Fahrzeug kann unter Kontrolle des Fahrers selbstständig einparken oder in Stausituationen die Spur sowie einen einprogrammierten Sicherheitsabstand zu den vorausfahrenden Fahrzeugen einhalten. Im Notfall können Notbremsungen vom Fahrzeug selbst eingeleitet oder dem Fahrer mögliche Ausweichmanöver mitgeteilt und initiiert werden.

Im Bereich der Luftfahrt werden Flugvorgänge automatisiert, um die Piloten von monotonen Aufgaben zu entlasten und die Fehleranfälligkeit zu minimieren. Automatisiert bedeutet dabei, dass bestimmte Funktionen vom Piloten aktiv an das System übergeben werden. Das System arbeitet anschließend anhand von festen Programmen die Aufgaben ab. Der Begriff der Autonomie wird in der Wissenschaft unterschiedlich aufgefasst, für diese Arbeit wird ein System als autonom betrachtet, wenn es seine Aktionen selbstständig plant, ausführt und auf Veränderungen der Umwelt entsprechend reagiert. Wird ein 3-dimensionaler Pfad (im Folgenden als Trajektorie bezeichnet) durch einen Menschen vorgegeben und vom System abgeflogen, so wird dies hochautomatisiert, aber nicht als autonom, betrachtet.

Um einem System die Autonomie zu ermöglichen, müssen einige grundlegende Fähigkeiten gegeben sein. Die erste Bedingung an das System ist das selbstständige Planen von Trajektorien. Bekannte Pfadplanungsansätze arbeiten mit Hinderniskarten, in denen die Hindernisse mit verschiedenen Eigenschaften (wie z.B. der Farbe, der Oberflächenstruktur, etc.) vermerkt werden. Durch bildgebende Sensoren, wie Laserscanner und Kameras, können

detektierte Hindernisse zu diesen Karten hinzugefügt oder die vermerkten Eigenschaften der Hindernisse verbessert werden.

Damit die von den Sensoren detektierten Hindernisse in die Karten eingetragen werden können, müssen die Sensordaten zwischen verschiedenen Koordinatensystemen (Sensor-, Träger- und Weltkoordinatensystemen) transformiert werden. Diese Transformationen sind aufgrund der unterschiedlichen Position und Lage der Sensoren auf dem Träger und des Trägers im 3-dimensionalen Raum notwendig. Die Zusammensetzung einer Position und Lage wird als Pose bezeichnet.

Die Transformation der Sensordaten in das globale Koordinatensystem oder in andere Sensorkoordinatensysteme enthält oft unbekannte und variable Parameter. Einerseits ist die genaue Pose des Sensors im Gehäuse unbekannt und andererseits ist die Pose des Sensors auf dem Träger je nach Experiment unterschiedlich.

In dieser Arbeit wird eine Lösung präsentiert, die die Bestimmung der Montagepose von Light Detection and Ranging (LiDAR)-Sensoren ermöglicht. Dabei werden Bewegungssensoren anhand einer Inertial Measurement Unit (IMU) sowie Sensoren zur Positionsbestimmung wie Global Positioning System (GPS) verwendet.

Ausgangspunkt für den zu entwerfenden Ansatz sind relative Translationen und Rotationen (six degrees of freedom (6DoF)) des Laserscanners zur betrachteten Umwelt. Diese werden mit Hilfe des Incremental Closest Point (ICP) Algorithmus aus den Scandaten bestimmt. Über den Abgleich der berechneten Transformation, der gemessenen Bewegung der IMU und dem Einsatz einer nicht linearen Optimierung, wie Levenberg-Marquardt, wird anschließend die 6DoF Transformation zwischen den beiden Sensoren bestimmt.

## 2. Grundlagen

In diesem Kapitel werden die geometrischen, mathematischen und algorithmischen Grundlagen erläutert.

### 2.1. Die verwendeten Koordinatensysteme

Autonomes Fliegen erfordert verschiedene Sensoren, die die Umwelt des autonomen Systems vermessen. Bevor diese Daten in einen Zusammenhang gebracht werden können, müssen sie auf eine gemeinsame Datenbasis fusioniert werden. Bei Bildgebenden Sensoren entsteht eine Art Bild, entweder aus Pixeln in einem Array oder durch radiale Entfernungen. Diese Daten befinden sich in einem sogenannten Sensorkoordinatensystem und sind abhängig von der Realisierung des Sensors.

Viele Sensoren sind direkt auf dem autonomen System (z.B. bei dem Projekt Autonomous Rotorcraft Testbed for Intelligent Systems (ARTIS)) montiert und unterscheiden sich, in Bezug auf das Trägerkoordinatensystem, zusätzlich in Positionen und Lage zueinander.

Des Weiteren befindet sich das autonome System zu einem bestimmten Zeitpunkt in einer Lage und einer Position in seiner Umgebung. Dessen Bezugsort ist der Ursprung für das Weltkoordinatensystem.

#### 2.1.1. Das Sensorkoordinatensystem

Das Sensorkoordinatensystem ist je nach Sensor und Hersteller unterschiedlich definiert. Im Folgenden werden die Koordinatensysteme erläutert, die in dieser Arbeit hauptsächlich Anwendung finden.

### 2.1.2. Das Trägerkoordinatensystem

Das Trägerkoordinatensystem beschreibt die Position und Lage der verschiedenen Sensoren zum Bezugspunkt des Trägers. Als Träger wird meist die Konstruktion bezeichnet, auf dem die Sensoren befestigt sind. Da in vielen Anwendungen auch Parameter gewünscht sind, die den Zustand des autonomen Systems beschreiben, wird in dieser Arbeit das autonome System als Träger betrachtet.

Der Ursprung und die Ausrichtung der Koordinatenachsen hängt von der Anwendung des Trägers ab. Ein autonomes System hat die Aufgaben, autonom in einer unbekannten Welt zu navigieren und sich entsprechend zu bewegen. Aus diesem Grund wird der Ursprung des Koordinatensystems entweder in den Schwerpunkt des Trägers oder in das Sensorkoordinatensystem der Beschleunigungssensoren gelegt. Die Ausrichtung entspricht dabei der Hauptbewegungsrichtung des Trägers.

### 2.1.3. Das Weltkoordinatensystem

Das Weltkoordinatensystem bezieht sich auf die Umgebung, in der sich der Träger bewegt. Es dient der Kartografie durch die Sensorwerte und zur Navigation des Trägers. Der Koordinatenursprung kann durch die GPS-Position eindeutig festgelegt werden.

## 2.2. Koordinatentransformation nach Denavit-Hartenberg

- Bezug auf Eigen

Zur Verortung von Sensordaten in der globalen Weltkarte, müssen sie Schritt-für-Schritt vom Sensorkoordinatensystem in das Weltkoordinatensystem transformiert (überführt) werden. Diese Problematik ist vergleichbar mit der Bewegungsbeschreibung eines Roboterarmes. Dabei wird in jedes Gelenk ein Koordinatensystem gelegt, dessen Z-Achse in Richtung des nächsten Gelenks im Roboterarm zeigt. Folglich kann die Bewegung des Aktors durch die schrittweise Transformation zwischen den Gelenken oder durch eine einzige Transformationsmatrix beschrieben werden.

## 2.3. Sensortechnik

Reflektivität, Laufzeitmessung, Fehlerbetrachtung

### 2.3.1. Lasersanner

Laserscanner liefern die Messwerte im Allgemeinen in Polarkoordinaten. Dabei handelt es sich um Messwerte und Winkelangaben und sind abhängig vom Aufbau des Laserscanners.

- Verzerrung durch Bewegung
- Distanzfehler (+/- 10 cm)
- Winkelfehler

### 2.3.2. Inertiale Messeinheiten

Aufbau:

- Gyro
- Beschleunigung
- Masseschwerpunkt wird als Hauptbezugssystem verwendet

#### 2.3.2.1. Gyros

Fehlerquellen:

- Drift

#### 2.3.2.2. Beschleunigungssensoren

Bewegungssensoren messen die Beschleunigungen in einer bestimmten Richtung und liefern meist skalare Messwerte. Werden mehrere Beschleunigungssensoren zusammengefasst, entsteht ein Sensorsystem, eine sogenannte IMU. Dieses Sensorsystem liefert je nach

Ausführung mehrdimensionale Messwerte, die die Position und Lage zur befindlichen Welt beschreiben.

Fehlerquellen:

- Ungenauigkeit des Beschleunigungssensors

### 2.3.3. Systeme zur globalen Positionsbestimmung

- Notwendigkeit bei der Kalibrierung, die IMU zu stabilisieren?

Fehlerquellen:

- fehlerhaftes GPS (bis zu 15 m)

### 2.3.4. Systemfehler

- Zeitlicher Versatz der Messungen zwischen Sensoren (lösbar durch Synchronisation)
- Blindzeit (Sensoren messen in unterschiedlichen Intervallen)
- fehlerhafte Angaben zur Pose zueinander

## 2.4. Algorithmen zur Transformation

### 2.4.1. ICP der PCL

Fehlerquellen:

- Rechenungenauigkeiten durch Transformationen
- Elimination wichtiger Features bei Datenreduktion (z.B. Statistical Outlier Removal)

### 3. Kalibrierung von LiDAR-Sensoren zu Inertialen Messeinheiten

Sensoren stellen eine Grundlage der Wissensgewinnung für ein Unmanned Aircraft (UA) dar. Dieses Wissen kann anschließend in verschiedenen Algorithmen verwendet werden, um Trajektorien zu planen, zu verfolgen oder weiterführende Entscheidungen zu treffen. Die Voraussetzung für dieses Wissen ist die Glaubwürdigkeit. Glaubwürdigkeit bedeutet dabei, dass zur genauen Verortung von Hindernissen deren genaue Position bekannt sein muss. Die Sensoren sind an jeweils unterschiedlichen Posen am Träger, dem UA, angebracht. Einige Gründe dafür sind:

- variable Grundkonfigurationen, die je nach Flugauftrag unterschiedliche Sensoren transportieren,
- unterschiedliche Trägersysteme, da je nach Umgebung bestimmte UA eingesetzt werden müssen (Größen- oder Gewichtsbestimmungen)
- verschiedene Beladungszustände, die eine Austarierung der Sensoren erfordern.

Daraus resultiert, dass für jede unterschiedliche Verwendung des UA die Sensorparameter erneut kalibriert werden müssen. Bei bildgebenden Sensoren wie Kameras oder Laserscanner ist die genaue Bestimmung der Pose elementar, da es sonst nicht möglich ist, sichere und eindeutige Rückschlüsse auf die Umgebung ziehen zu können. Auf Grund einer nicht-planaren Trägeroberfläche und einer dynamischen Flotte an UA's gibt es bisher kein automatisiertes Verfahren zur Kalibrierung. Bisher werden die Konfigurationen stets von Hand vermessen und kalibriert. Dadurch kommt es zu großen Ungenauigkeiten und Inkonsistenzen in den Vermessenen Positionen. Des Weiteren ist die Vermessung der Lage des Sensors sehr schwierig.

Eine automatisierte Kalibrierung zwischen LiDAR und IMU Sensoren ermöglicht eine präzisere und deterministische Verwendung. Des Weiteren sollen die Träger problemlos angepasst oder ausgetauscht werden können. Die Kalibrierung soll eine genauere Bestimmung der Pose ermöglichen, als bisher von Hand möglich. Die folgenden Kapitel erläutern die verschiedenen in Betracht gezogenen Ansätze, beleuchten die jeweiligen Vor- und Nachteile und beschreiben die schlussendlich gewählte Implementierung.

### 3.1. Problemanalyse

Die Bestimmung von Lage und Position im 3-dimensionalen Raum bezeichnet die Bestimmung von 6 Freiheitsgraden (6DoF). Die 6 Freiheitsgrade werden in 3 translatorische und 3 rotatorische Freiheitsgrade unterteilt. Die translatorischen Freiheitsgrade bestimmen die Position in  $X$ -,  $Y$ -, und  $Z$ -Achse in einem Weltkoordinatensystem. Die rotatorischen Freiheitsgrade bestimmen die Lage an dieser Position in Bezug zur Erdoberfläche. Gemäß Tabelle 3.1 werden die 3 Eulerschen Winkeln  $\Phi$  (Phi),  $\Theta$  (Theta) und  $\Psi$  (Psi) verwendet.

Tabelle 3.1: Bezeichnung der Winkel gemäß DIN 9300 / ISO 1151-2:1985

Winkel	Bezeichnung	Rotationsachse
$\Phi$	Gierwinkel	$Z$
$\Theta$	Nickwinkel	$Y$
$\Psi$	Rollwinkel	$X$

### 3.2. Möglichkeiten zur Kalibrierung

Für die Kalibrierung der Sensoren werden folgende Anforderungen definiert:

- die Umgebung wird als unveränderlich und starr angenommen,
- die zu vermessende Bewegung muss größer als die größte Messungenauigkeit des Systems sein.



### 3.2.1. Im Sensorkoordinatensystem

#### 3.2.1.1. Ohne Bewegungskorrektur

Abbildung 3.1 zeigt die Kalibrierung im Sensorkoordinaten

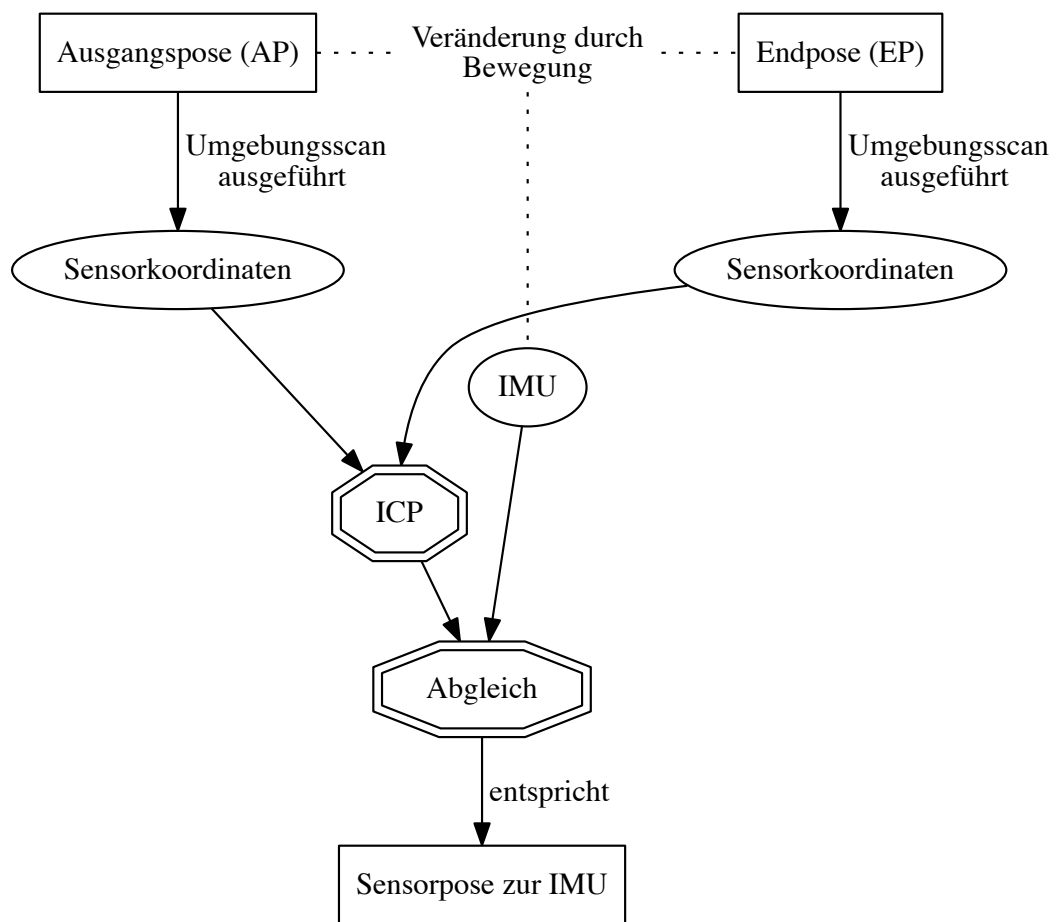


Abbildung 3.1.: Kalibrierung im Sensorkoordinatensystem

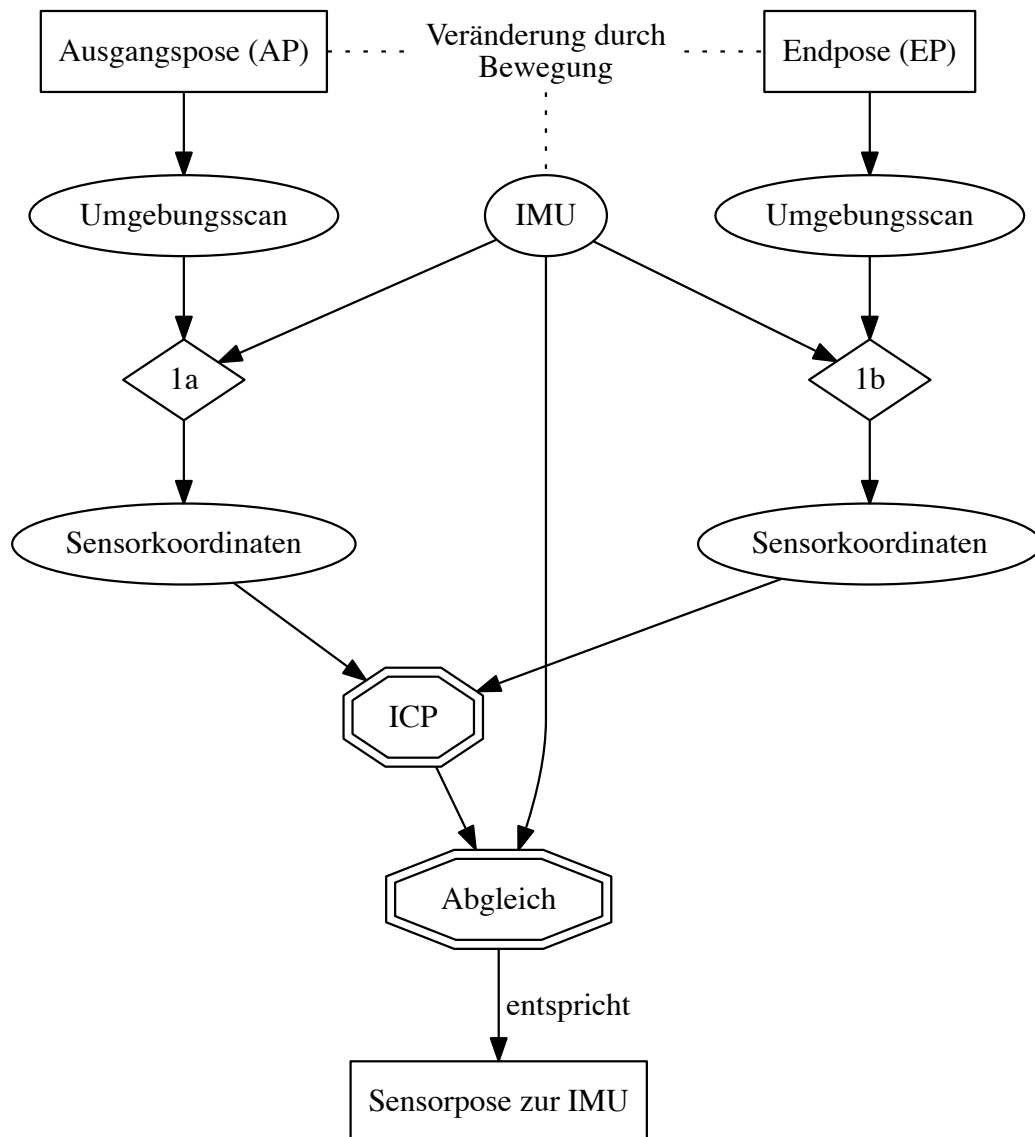


Abbildung 3.2.: Kalibrierung im Sensorkoordinatensystem mit Bewegungskorrektur

**3.2.1.2. Mit Bewegungskorrektur**

**3.2.2. Im Weltkoordinatensystem**

**3.2.2.1. Ohne Bewegungskorrektur**

**3.2.2.2. Mit Bewegungskorrektur**

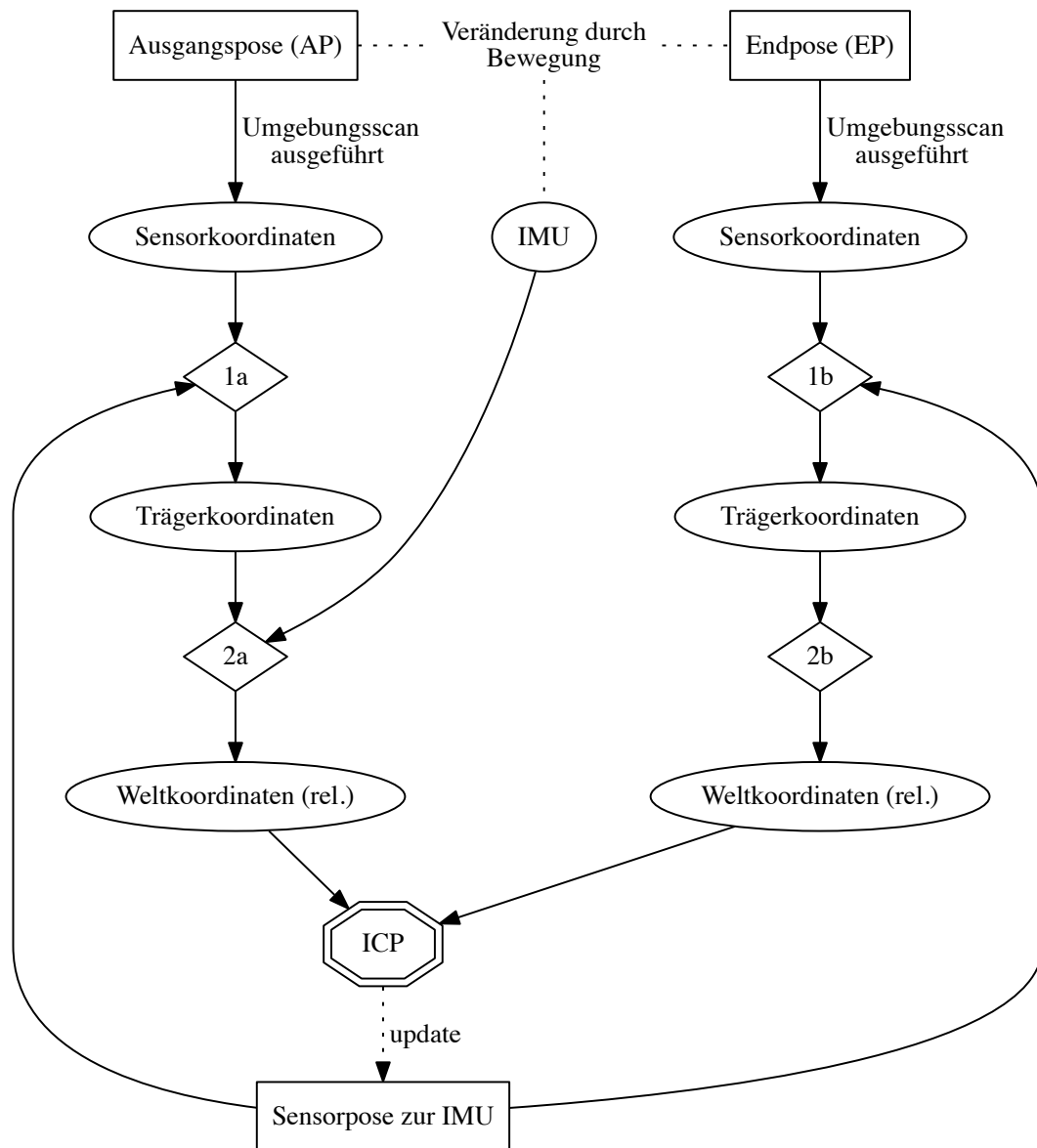


Abbildung 3.3.: Kalibrierung im Weltkoordinatensystem

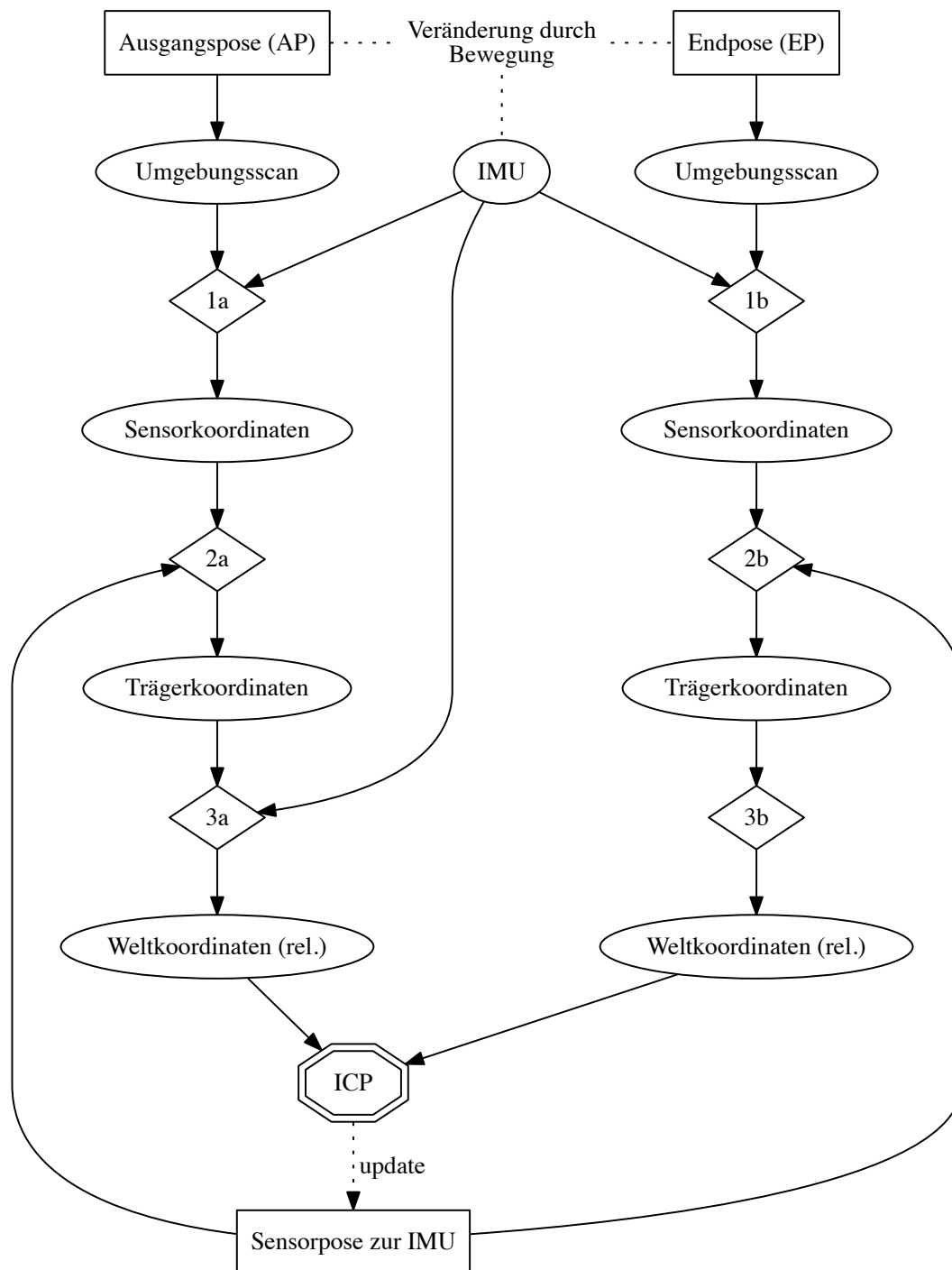


Abbildung 3.4.: Kalibrierung im Weltkoordinatensystem mit Bewegungskorrektur

## **4. Validierung**

### **4.1. Einsatzgebiet**

autonomer Hubschrauberflug

### **4.2. Aufbau**

Welche Sensorprodukte werden wie verwendet (Eigenschaften, Auflösungen etc.)?

### **4.3. Ablauf**

Während eines Experimentalfluges werden in regelmäßigen Abständen Bewegungs- und Laserdaten aufgenommen. Die Abstände richten sich nach den jeweiligen Fähigkeiten der Sensoren.

### **4.4. Ergebnisse**

### **4.5. Gesamtergebnis**

## **5. Fazit und Ausblick**

1. Was wurde gemacht?
2. Was war das Resultat?
3. Was ergeben sich für Folgeaufgaben?

# Anhang

## A. Reproduktion der Forschung

Zur Reproduktion der Forschung wird im Folgenden die Entwicklungsumgebung vorgestellt. Für konkrete Information zur Installation wird einerseits auf die jeweilige Dokumentation für das Programm sowie auf das Github-Repository [gismo141/laserIMUCalibration](https://github.com/gismo141/laserIMUCalibration) als Begleitmaterial der vorliegenden Arbeit verwiesen.

Die Implementierung der Algorithmen erfolgte einerseits in Python (Fast-Implementation) und C++ zur Integration in das vorhandene Framework. Dabei wurden die Betriebssysteme Mac OS X 10.10.3 und Windows 7 verwendet. Es wurde darauf Wert gelegt, Open-Source Software zu verwenden.

### A.1. Entwicklungsumgebung

Zur bequemen Installation der benötigten Programme wird für Mac OS X ein Paketmanager wie Homebrew oder MacPorts empfohlen. Eventuelle Abhängigkeiten können somit während der Installation festgestellt und aufgelöst werden. Homebrew trennt dabei die vom Benutzer installierten Programme von den system-eigenen wodurch Komplikationen vermindert werden.

Homebrew stellt keine Notwendigkeit dar, die benötigte Software kann auch von Hand heruntergeladen, kompiliert und installiert werden.

Tabelle 1: Notwendige Programme zur Erstellung der Arbeit und Ihrer Abhängigkeiten

Tool	Version	Verwendung
Qt5	5.4.1	UI-Design (Oberflächenentwicklung)
VTK	6.2.0	Visualisierung der Daten und Algorithmen
PCL	1.7.2	Algorithmen zum Umgang mit Punktwolken (ICP, Outlier etc.)
CMake	3.2.2	einheitliche Build-Umgebung
Python	2.7	schnelles Algorithmen-Prototyping



---

Tool	Version	Verwendung
Pandoc	1.13.2.1	Erstellung der Masterarbeit in unterschiedlichen Formaten
GraphViz	2.38.0	Visualisierung der Graphen und Algorithmen
Doxygen	1.8.9.1	Dokumentation des Quellcodes
Gnuplot	5.0.0	Plotten der gewonnenen Daten

---

## **A.2. Interessante Links**

- RawGit (Extrahiert Links aus Github-Repositories)
- PCLVisualizer in Qt verwenden
- Python-PCL
- Erstellung eigener Punkt-Datentypen in der PCL