

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: 31.08.2015

## **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, 31.08.2015

## **Danksagung**

Ich danke meinem Dozenten Prof. Dr. Harald Görl, der mir die Chance und sein Vertrauen gab, diese Masterarbeit zu ermöglichen und mir durch die anregenden Gespräche ein unvergleichlich interessantes Studium gestaltete. Des Weiteren danke ich Stefan Krause, der meinen Fragen und Problemen stets mit Rat und Tat zur Seite stand und mich erstklassig betreute sowie allen Mitarbeitern und Co-Studenten am DLR in Braunschweig für die freundliche Aufnahme, die gewonnenen Erkenntnisse und das angenehme Arbeitsumfeld. Ich danke meinen Eltern für die tatkräftige Unterstützung während meines Studiums und ganz besonders meiner Frau, Ramona Riedel, und ihrer Familie, die mir jederzeit motivierend zur Seite stand.

## **Abstract**

Unmanned aircraft are used, to fulfil tasks that are dirty, dangerous and demeaning for human beings. Some applications require a map of the environment, that can be generated with LiDAR-sensors. The mapping is done by consecutively registering the laser-generated point-clouds with a manually measured mounting-pose between the LiDAR and NAV sensors (IMU and GNSS). The registration fails, when the mounting-pose is inaccurate. In this thesis a method for an automated relative extrinsic calibration between the LiDAR and NAV is proposed, that uses a non-linear optimisation (Levenberg-Marquardt) to optimize the given mounting-pose. The results show, that the calibration may fail when the point-cloud data suffers from geometric instability, therefore filter-techniques such as covariance-sampling are presented to improve the point-cloud-data. This approach is promising and needs further investigation concerning the parametrization of the proposed techniques.

## **Zusammenfassung**

Unbemannte Luftfahrzeuge werden verwendet, um für den Menschen niedere, gefährliche und zermürbende Aufgaben zu erfüllen. Für anspruchsvolle Anwendungen werden Umgebungskarten benötigt, die mit LiDAR-Sensoren erstellt werden. Dabei werden Punktwolken anhand der Montagepose des LiDAR zu NAV-Sensoren (IMU und GNSS) registriert. Die Genauigkeit der Montagepose, die bisher ungenau und aufwendig von Hand vermessen wird, entscheidet über die Qualität der Karte. In dieser Arbeit wird ein Ansatz zur automatisierten relativen Kalibrierung von LiDAR zu NAV vorgestellt, der die Montagepose durch eine nicht-lineare Optimierung (nach Levenberg-Marquardt) bestimmt. Die Ergebnisse zeigen, dass die Kalibrierung fehlschlagen kann, wenn die Punktwolken geometrisch instabil sind, weswegen Filter-Techniken wie das Kovarianz-Downsampling vorgestellt werden, um die Punktwolken-Daten zu verbessern. Der Ansatz ist vielversprechend. Weitere Forschung bezüglich der Parametrierung der vorgestellten Techniken ist angebracht und sinnvoll.

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>iv</b>
<b>Tabellenverzeichnis</b>	<b>vii</b>
<b>Abbildungsverzeichnis</b>	<b>viii</b>
<b>1. Einleitung</b>	<b>1</b>
<b>2. Grundlagen</b>	<b>3</b>
2.1. Mathematik und Geometrie . . . . .	3
2.1.1. Lageänderung durch Rotation . . . . .	3
2.1.2. Positionsänderung durch Translation . . . . .	5
2.1.3. Transformation von Punktmenzen . . . . .	6
2.1.4. Der Gradient als Richtung und Maß des steilsten Anstiegs . . .	7
2.2. Sensortechnik . . . . .	8
2.2.1. Visuelle Odometrie durch LiDAR-Sensoren . . . . .	9
2.2.2. Messung der inertialen Bewegung und globalen Position . . . . .	17
2.3. Datenstrukturen . . . . .	30
2.3.1. Punkte . . . . .	31
2.3.2. Punktwolken . . . . .	31
2.3.3. Suchbäume . . . . .	32
2.4. Bezugssysteme . . . . .	32
2.4.1. Sensor . . . . .	33
2.4.2. Träger . . . . .	33
2.4.3. Welt . . . . .	33

<b>3. Relative extrinsische Kalibrierung von bildgebenden Umweltsensoren</b>	<b>35</b>
3.1. Forschungsstand . . . . .	36
3.2. Ansatz zur Kalibrierung . . . . .	39
3.3. Bewegungsextraktion aus aufeinanderfolgenden Punktwolken . . . . .	43
3.3.1. Registrierung von Punktwolken . . . . .	43
3.3.2. Extraktion von Features . . . . .	44
3.3.3. Filterung der Daten . . . . .	47
3.3.4. Zusammenfassung . . . . .	50
3.4. Nichtlineare Optimierung nach Levenberg-Marquardt . . . . .	51
3.4.1. One-Variable-Optimisation . . . . .	51
3.4.2. N-Variable-Optimisation . . . . .	53
3.5. Umsetzung der Kalibrierung im DIP-Framework . . . . .	57
3.5.1. Entwicklung der Softwarearchitektur . . . . .	57
3.5.2. Implementierung der optimierten Lösung . . . . .	59
3.5.3. Entwicklung der graphischen Oberfläche . . . . .	60
3.5.4. Anwendung des Filters zur relativen extrinsischen Kalibrierung .	62
<b>4. Validierung</b>	<b>63</b>
4.1. Ausgangspunkt der Kalibrierung . . . . .	63
4.2. Verwendete Sensoren . . . . .	65
4.2.1. Der LiDAR-Sensor . . . . .	65
4.2.2. Die inertiale Messeinheit . . . . .	66
4.3. Versuch - <i>In-Flight</i> . . . . .	67
4.3.1. Aufbau . . . . .	67
4.3.2. Ablauf . . . . .	68
4.3.3. Ergebnisse . . . . .	69
4.4. Versuch - <i>Common Ground</i> . . . . .	71
4.4.1. Aufbau . . . . .	71
4.4.2. Ablauf . . . . .	72

4.4.3. Ergebnisse . . . . .	76
4.5. Ergebniisanalyse . . . . .	79
<b>5. Fazit und Ausblick</b>	<b>80</b>
<b>Literaturverzeichnis</b>	<b>82</b>

## **Abkürzungsverzeichnis**

<b>6DoF</b>	six degrees of freedom
<b>CCD</b>	Charge-Coupled Device
<b>DGPS</b>	Differential Global Positioning System
<b>DIN</b>	Deutsche Industrienorm
<b>DIP</b>	Digital Image Processing
<b>DLR</b>	Deutsches Institut für Luft- und Raumfahrt
<b>DOP</b>	Dilution Of Precision
<b>ECEF</b>	Earth-Centered Earth-Fixed
<b>EGM96</b>	Earth Gravitational Model 1996
<b>ESA</b>	European Space Agency
<b>FoV</b>	Field-of-View
<b>FPFH</b>	Fast Point-Feature-Histogram
<b>GLONASS</b>	Globalnaja nawigazionnaja sputnikowaja sistema (russisch für Globales Satellitennavigationssystem)
<b>GNSS</b>	Global Navigation Satellite System

<b>GPS</b>	Global Positioning System
<b>ICP</b>	Iterative Closest Point
<b>IMU</b>	Inertial Measurement Unit
<b>INS</b>	Inertial Navigation System
<b>LASER</b>	Light Amplification by Stimulated Emission of Radiation
<b>LiDAR</b>	Light Detection and Ranging
<b>MEMS</b>	mikro-elektro-mechanische Systeme
<b>NAV</b>	Navigation
<b>NAVSTAR-GPS</b>	Navigational Satellite Timing and Ranging – Global Positioning System
<b>NED</b>	North-East-Down
<b>PCL</b>	Point-Cloud-Library
<b>RaDAR</b>	Radio Detection and Ranging
<b>RGB-D</b>	Red-Green-Blue and Distance
<b>RTK</b>	Real-Time-Kinematic
<b>SI</b>	Système international d'unités (französisch für Internationales Einheitsystem)
<b>SIFT</b>	Scale-invariant feature transform
<b>SLAM</b>	Simultaneous Localization and Mapping
<b>ToF</b>	Time-of-Flight
<b>UA</b>	Unmanned Aircraft

<b>UDP</b>	User Datagram Protocol
<b>ULF</b>	Unbemannte Luftfahrzeuge
<b>UniBwM</b>	Universität der Bundeswehr München
<b>USB</b>	Universal Serial Bus
<b>UTC</b>	Coordinated Universal Time (englisch für koordinierte Weltzeit)
<b>WGS84</b>	World Geodetic System 1984

## Tabellenverzeichnis

## Abbildungsverzeichnis

2.1. Schematischer Aufbau eines LiDAR nach Kern (2003) . . . . .	10
2.2. Varianten der Reflexion von Licht . . . . .	13
2.3. Auswirkung der Satellitenzahl auf die Positionsbestimmung bei GNSS .	22
2.4. Darstellung der verschiedenen Bezugssysteme (Andert (2011)) . . . . .	34
 3.1. Ablauf der Kartografie von Punktwolken . . . . .	35
3.2. Gegenüberstellung von einem einzelnen Laserscan und seiner Überführung in eine Karte . . . . .	36
3.3. Veränderung des Bildbereichs, wenn die Bewegung des Trägers (von $P_1$ nach $P_5$ ) schneller als die Aufnahmefrequenz des Sensors ist (schematische Darstellung) . . . . .	40
3.4. Kalibrierung im Weltkoordinatensystem . . . . .	41
3.5. Kalibrierung im Weltkoordinatensystem mit Bewegungskorrektur . . .	42
3.6. Angewandtes Kovarianz-Downsampling (rot) und die originale Punktwolke (blau) . . . . .	50
3.7. Klassendiagramm zur Umsetzung der Kalibrierung im DIP . . . . .	58
3.8. Übersicht der graphischen Oberfläche des DIP-Frameworks mit hinzuge- schaltetem LaserNAVCalibration-Filter (unterer Bildausschnitt) . . .	61
 4.1. Überlagerung von Kamera- und Laserscandaten zur Hinderniserkennung	64
4.2. Schematische Darstellung des Velodyne HDL-32e . . . . .	65
4.3. Produktbild der iMAR iTraceRT-F400 Q (iMAR (2013)) . . . . .	66
4.4. SwissDrones Dragon 50 im Flugversuch . . . . .	67
4.5. Trajektorie des UA während des Versuchs <i>In-Flight</i> . . . . .	69

4.6. Abhangigkeit zwischen dem ICP-Fehler und der Scan-Umgebung . . . . .	70
4.7. Gegenuberstellung der mittleren ICP-Fehler . . . . .	70
4.8. Tragerplattform im Versuch <i>Common-Ground</i> . . . . .	73
4.9. Draufsicht des Versuchs <i>Common Ground</i> im Hohenprofil . . . . .	75
4.10. Analyse der Punktwolken-Paare auf markante Bewegungen . . . . .	76
4.11. Gegenuberstellung der berechneten Montageposen im Sensor- und Weltbezugssystem unter der Vorgabe einer falschen und der korrekten Montagepose	77
4.12. Gegenuberstellung der Konvergenz uber die Anzahl der benotigten Iterationen und des fortlaufenden ICP-Fehlers . . . . .	78

## 1. Einleitung

Das Deutsche Institut für Luft- und Raumfahrt (DLR) erforscht am Institut für Flugsystemtechnik in der Abteilung für Unbemannte Luftfahrzeuge (ULF) die automatisierte Bewegung von Unmanned Aircraft (UA) in unbekanntem Luftraum. Zur erfolgreichen Navigation wird eine Karte benötigt, die im Flug aufgebaut werden muss.

Dafür machen bildgebende Sensoren in zeitlichen Abständen Aufnahmen von der Umgebung. Diese Momentaufnahmen müssen für die korrekte Kartographie mit Bewegungsdaten des UA in Beziehung gebracht werden.

Aus bautechnischen Gründen befinden sich die Bewegungs- und die Bildgebungssensoren in der Regel an unterschiedlicher Position und Ausrichtung am UA. Damit eine korrekte Kartographie möglich ist, müssen diese Sensoren zueinander kalibriert werden (in Beziehung gebracht werden).

Bisher wird diese Kalibrierung manuell und für jeden Flug neu angefertigt. Dabei werden die Entfernung und die Lage zwischen den Sensoren von Hand vermessen. Dieses Verfahren ist aufwendig und fehleranfällig. Aus diesen Gründen sollte das Verfahren automatisiert werden.

Deshalb wird für diese Arbeit folgende Forschungsfrage formuliert: „*Ist es möglich, die Kalibrierung von bildgebenden Sensoren relativ zu Bewegungssensoren zu automatisieren?*“ Der präsentierte Ansatz behandelt die automatisierte Kalibrierung von Light Detection and Ranging (LiDAR)-Sensoren als bildgebende Sensoren zu Bewegungssensoren (bestehend aus einer Inertial Measurement Unit (IMU) mit Global Navigation Satellite System (GNSS)).

Zuerst werden die Gemeinsamkeiten zweier aufeinanderfolgender Laserscans identifiziert und die eventuelle Bewegung zwischen den Scans durch einen Iterative Closest Point (ICP)-Algorithmus geschätzt. Das Fachgebiet zur Schätzung von Bewegungen aus

Bildinformationen wird als visuelle Odometrie oder auch Ego-Motion (engl.), bezeichnet.

Parallel werden Navigationsdaten mittels einer IMU und eines GNSS aufgezeichnet. Aus diesen Daten wird eine Bewegung bestimmt und im Anschluss mit der Schätzung des ICP verglichen. Da sich die drei Sensoren (IMU, GNSS und LiDAR-Sensor) zueinander statisch auf einer Plattform befinden, muss die Differenz der zwei geschätzten Bewegungen (Bewegung des bildgebenden Sensors/ Bewegung der Navigationssensoren) das Resultat einer unterschiedlichen Position und Lage (Montagepose) zwischen LiDAR-Sensor und der Kombination aus IMU und GNSS sein. Diese Differenz entspricht den gesuchten Parametern für die relative Kalibrierung.

Die vorliegende Arbeit ist in drei Hauptkapitel unterteilt. Das Kapitel Grundlagen gibt einen Einblick in die physikalischen Grundlagen der verwendeten Sensoren, in die mathematischen Grundlagen zur Verarbeitung der Messdaten und in die verschiedenen Datenstrukturen für die algorithmische Verknüpfung. Im Anschluss wird der Ansatz zur relativen extrinsischen Kalibrierung präsentiert, der während der Masterarbeit als Filter im Digital Image Processing (DIP)-Framework<sup>1</sup> implementiert wurde. Den Abschluss bildet die Validierung der Kalibrierung in zwei voneinander unabhängigen Experimenten.

---

<sup>1</sup> Das DIP-Framework dient seit 2004 als Plattform zur Evaluation digitaler Bildverarbeitungsalgorithmen in der Abteilung ULF am DLR. Für weitere Informationen zur grundlegenden Idee und den Hintergründen des Frameworks siehe Guth (2004) und Goormann (2004).

## 2. Grundlagen

Zum besseren Verständnis der Arbeit werden in den folgenden Abschnitten die Grundlagen in den Bereichen der Mathematik, Sensortechnik, Bezugssystemen und komplexen Datenstrukturen gelegt.

### 2.1. Mathematik und Geometrie

Koordinatensysteme bilden die Grundlage, um die Position und Lage in einem mehrdimensionalen Raum zu definieren. Durch Rotationen und Translationen können die Koordinatensysteme zueinander transformiert werden.

#### 2.1.1. Lageänderung durch Rotation

Die Rotation eines Vektors  $(x, y, z)$  erfolgt mit einer  $3 \times 3$ -Matrix. Die Grundlage bildet dabei die Identitätsmatrix  $I$ , die sich dadurch auszeichnet, gemäß (2.1) auf der Diagonalen mit 1 und auf allen anderen Feldern mit 0 besetzt zu sein.

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.1)$$

Die Identitätsmatrix sorgt dafür, dass die Rotation um die Achse erfolgt und der Wert dieser Achse gleich bleibt. Möchte man nun den Vektor  $(x, y, z)$  mit  $R$  um die  $z$ -Achse rotieren, so bleibt der Anteil der  $z$ -Koordinate gleich, wohingegen die Anteile  $x$  und  $y$  zu  $x'$  und  $y'$  verändert werden. Es resultiert die Gleichung (2.2) in Matrixschreibweise und (2.3) in Kurzform.

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (2.2)$$

$$\vec{v}' = R_z(\phi) * \vec{v} \quad (2.3)$$

Die Kurzform  $R_z\phi$  bezeichnet die Achse und den dazugehörigen Winkel, um den gedreht wird. Für eine anschließende Rotation um die  $y$ -Achse resultiert (2.4).

$$\begin{pmatrix} x'' \\ y'' \\ z'' \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} * \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} \quad (2.4)$$

$$\vec{v}'' = R_y(\theta) * \vec{v}' \quad (2.5)$$

Abschließend wird in diesem Beispiel um die  $x$ -Achse rotiert. Es resultieren die Gleichungen (2.6) und (2.7).

$$\begin{pmatrix} x''' \\ y''' \\ z''' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{pmatrix} * \begin{pmatrix} x'' \\ y'' \\ z'' \end{pmatrix} \quad (2.6)$$

$$\vec{v}''' = R_x(\psi) * \vec{v}'' \quad (2.7)$$

Dieses schrittweise Vorgehen erleichtert das Verständnis, erschwert jedoch die Rechnung. Gemäß (2.8) und (2.9) werden die einzelnen Matrizenmultiplikationen zusammengefasst. Es ist jedoch zu beachten, dass die Multiplikation von Matrizen nicht kommutativ, also nicht tauschbar, ist, da sonst eine andere Gesamtrotation resultiert.

$$\begin{pmatrix} x''' \\ y''' \\ z''' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{pmatrix} * \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} * \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (2.8)$$

$$\vec{v}''' = R_x(\psi) * R_y(\theta) * R_z(\phi) * \vec{v} \quad (2.9)$$

Für die nachfolgende Matrix wurde sin durch  $s$  und cos durch  $c$  ersetzt. Durch Ausmultiplizieren erhält man gemäß Gleichung (2.10) und (2.11) eine gesamte Rotationsmatrix  $R_{zyx}$ . Somit minimiert man die Rechnung von drei einzelnen Rotationen zu einer einzigen Rotation. Durch Einsetzen verschiedener Winkel  $\phi$ ,  $\theta$  und  $\psi$  kann nun jeder Vektor im 3-dimensionalen Raum rotiert werden.

$$\begin{pmatrix} x''' \\ y''' \\ z''' \end{pmatrix} = \begin{pmatrix} c\theta & c\phi & -c\theta & s\phi & -s\theta \\ -s\psi & s\theta & c\phi + s\phi & c\psi & s\psi & s\theta & s\phi + c\psi & c\phi & -s\psi & c\theta \\ c\psi & s\theta & c\phi + s\psi & s\phi & -c\psi & s\theta & s\phi + s\psi & c\phi & c\psi & c\theta \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (2.10)$$

$$\vec{v}''' = R_{zyx}(\phi, \theta, \psi) * \vec{v} \quad (2.11)$$

### 2.1.2. Positionsänderung durch Translation

Wie in den vorherigen Abschnitten beschrieben, können Punkte in Form von Vektoren durch die Multiplikation mit Rotationsmatrizen gedreht werden. Als nächstes wird die Verschiebung einer Position  $P_1$  auf eine Position  $P_2$  betrachtet. Verallgemeinert kann das Problem gemäß (2.12) als Funktion formuliert werden, wobei die Position  $P_1$  als Vektor  $\vec{v}$  einer Funktion  $F(x)$  übergeben wird. Das Resultat der Funktion ist die neue Position  $P_2$  als Vektor  $\vec{v}'$ .

$$\vec{v}' = F(\vec{v}) \quad (2.12)$$

Die kürzeste Verbindung zwischen zwei Punkten ist stets die Gerade. Durch die Addition des Vektors  $\vec{v}$  mit einem Vektor  $\vec{v}_{Translation}$  wird gemäß (2.13) die Position  $P_1$  an die Position  $P_2$  verschoben, sodass die Funktion  $F(x)$  eine Vektoraddition ist.

$$\begin{pmatrix} 5 \\ 1 \\ 1 \end{pmatrix}_{v'} = \begin{pmatrix} 2 \\ 3 \\ 0 \end{pmatrix}_v + \begin{pmatrix} 3 \\ -2 \\ 1 \end{pmatrix}_{v_{Translation}} \quad (2.13)$$

Die Translation eines Punktes beschreibt somit die Verschiebung entlang einer Geraden.

### 2.1.3. Transformation von Punktmengen

Durch die Rotation kann die Lage, durch die Translation die Position eines Punktes verändert werden. Werden beide Operationen in einem Schritt durchgeführt, spricht man von einer Transformation. Eine Transformation kann ebenfalls in Form einer Matrix dargestellt werden. Dazu wird die Rotationsmatrix sowie der Translationsvektor in einer  $4 \times 4$ -Transformationsmatrix  $T$  gemäß (2.14) abgelegt.

$$T = \begin{pmatrix} R_0 & R_1 & R_2 & \Delta x \\ R_3 & R_4 & R_5 & \Delta y \\ R_6 & R_7 & R_8 & \Delta z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.14)$$

Die Transformationsmatrix  $T$  ermöglicht es nun, jeden Vektor durch Rotation zu drehen und durch Translation zu bewegen. Dadurch können nun Beziehungen zwischen mehreren Posen beschrieben werden. Anwendung findet dies zum Beispiel in der Sensortechnik, wenn Messwerte verschiedener bildgebender Sensoren auf Grund unterschiedlicher Blickrichtungen in ein gemeinsames Bezugssystem überführt werden sollen. Dabei wird jeder Punkt  $P_n$  einzeln in die neue Pose transformiert.

Die Rücktransformation bezeichnet die umgekehrte Transformation. Dabei wird gemäß der Matrizenrechnung von der Transformationsmatrix  $T$  die Inverse  $T^{-1}$  gebildet, sodass anschließend gemäß (2.15) jeder Punkt rücktransformiert werden kann.

$$\vec{v} = T^{-1} * \vec{v}' \quad (2.15)$$

#### 2.1.4. Der Gradient als Richtung und Maß des steilsten Anstiegs

Der Gradient  $g$  bezeichnet in der Mathematik einen Differentialoperator. Angewandt auf ein Skalarfeld<sup>2</sup> liefert der Gradient ein Vektorfeld<sup>3</sup>.

Bezogen auf eine Landkarte, die in Form der Funktion  $h(x, y)$  allen Punkten in der  $xy$ -Ebene eine Höhe  $h$  zuordnet, bezeichnet der Gradient  $g(h(x, y))$  einen Vektor in der  $xy$ -Ebene in Richtung des steilsten Anstiegs von  $h$ . Die Länge des Vektors ist ein Maß für die Steilheit an diesem Punkt. Die Bildung des Gradienten wird mit dem Nabla-Operator  $\nabla$  gekennzeichnet. Der Gradient wird durch die partiellen Ableitungen  $\frac{\partial f}{\partial x_i}$  von  $f$  in Richtung  $x_i$  gebildet. Für den 3-dimensionalen Raum wird der Gradient gemäß (2.16) berechnet.

$$\text{grad}(f) = \nabla f = \frac{\partial f}{\partial x} \vec{e}_x + \frac{\partial f}{\partial y} \vec{e}_y + \frac{\partial f}{\partial z} \vec{e}_z \quad (2.16)$$

Alternativ kann der Gradient als Spaltenvektor gemäß (2.17) dargestellt werden.

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix} \quad (2.17)$$

Die Vektoren  $\vec{e}_x$ ,  $\vec{e}_y$  und  $\vec{e}_z$  bezeichnen dabei die Einheitsvektoren in Richtung der Koordinatenachsen.

Geben sei das Skalarfeld gemäß (2.18).

<sup>2</sup> Ein Skalarfeld bezeichnet in der Analysis eine Funktion, die jedem Punkt eines Raumes ein Skalar (eine reelle Zahl) zuordnet.

<sup>3</sup> Ein Vektorfeld bezeichnet eine Funktion, die jedem Punkt eines Raumes einen Vektor zuordnet.

---


$$f(x, y) = 5y^2 + 3x^2 \quad (2.18)$$

Dieses Skalarfeld wird gemäß (2.19) nach  $x$  und (2.20) nach  $y$  partiell abgeleitet.

$$\frac{\partial f}{\partial x} = 6x \quad (2.19)$$

$$\frac{\partial f}{\partial y} = 10y \quad (2.20)$$

Es ergibt sich der Gradient gemäß (2.21) und (2.22) in Vektordarstellung.

$$\text{grad}(f) = \nabla f = 6x * \vec{e}_x + 10y * \vec{e}_y \quad (2.21)$$

$$\text{grad}(f) = \nabla f = \begin{pmatrix} 6x \\ 10y \end{pmatrix} \quad (2.22)$$

Anwendung findet der Gradient unter Anderem im Bereich der Bestimmung der Optimierungsrichtung nach Levenberg-Marquardt im Abschnitt 3.4.

## 2.2. Sensortechnik

Umweltwahrnehmung ist für uns Menschen eine überlebenswichtige Fähigkeit. Durch unsere Sinne erkennen wir unser Umfeld, können die Umgebung klassifizieren und bedrohliche Situationen erkennen. Bedrohlich bezeichnet dabei Situationen, in denen entweder wir, unsere Mitmenschen oder unser geplantes Handeln (zum Beispiel die Durchführung eines Auftrags) gefährdet sind. Ein großer Bestandteil unserer Wahrnehmung ist visuell und erfolgt durch unsere Augen.

### **2.2.1. Visuelle Odometrie durch LiDAR-Sensoren**

In automatisierten Systemen werden bildgebende Sensoren zur visuellen Wahrnehmung verwendet. Diese Sensoren erstellen in zeitlichen Abständen ein Bild, das aus Messgrößen eines realen Objektes besteht. Die Messgröße oder eine daraus abgeleitete Information hat einen geographischen Bezug und wird über Helligkeitswerte oder Farben kodiert visualisiert.

Die visuelle Odometrie, auch Ego-Motion (engl.), bezeichnet das Fachgebiet zur Schätzung von Bewegungen aus Bildinformationen. Dabei werden Veränderungen von Merkmalen (Features) aus verschiedenen Aufnahmen der Umwelt bestimmt. Welche Merkmale verwendet werden, ist abhängig von der Struktur der eingegebenen Daten.

Man unterscheidet organisierte und unorganisierte Datensätze. Organisierte Daten bezeichnen eine Matrix-ähnliche Struktur, in der die Daten in Zeilen und Spalten unterteilt sind. Sensoren, die organisierte Daten erzeugen, sind zum Beispiel Digitalkameras oder Time-of-Flight (ToF)-Kameras. Der Vorteil von organisierten Daten ist, dass durch die Kenntnis von Beziehungen zwischen benachbarten Punkten (z.B. Pixeln), Algorithmen, wie die Suche nach dem nächsten Nachbarn, effizienter gestaltet werden können.

Des Weiteren ist die Erkennung von Merkmalen abhängig von der Qualität der vorliegenden Messungen. Qualität bezeichnet dabei das Zueinanderpassen mehrerer Datensätze. Unterschiede, die durch Rotation, Translation, Skalierung oder Messrauschen verursacht werden, erfordern unterschiedliche Ansätze zur Extraktion von Merkmalen. In dieser Arbeit werden statische Posen der Sensoren zueinander vorausgesetzt. Das heißt, dass Unterschiede zwischen verschiedenen Bildern rein aus der Bewegung der Sensorplattform (Träger) und aus systembedingten Messfehlern resultieren.

#### **2.2.1.1. Exemplarischer Aufbau und Funktionsweise eines LiDARs**

LiDAR-Sensoren dienen der optischen Abstandsmessung. Einsatzgebiete sind unter anderem topographische Geländeaufnahmen (Kartographie) (Lindenberger 1993), Detektion und Tracking von Hindernissen (Mendes et al. 2004) und die Identifikation atmosphärischer

Eigenschaften (Wulfmeyer/Bösenberg 1998; Browell et al. 1998).

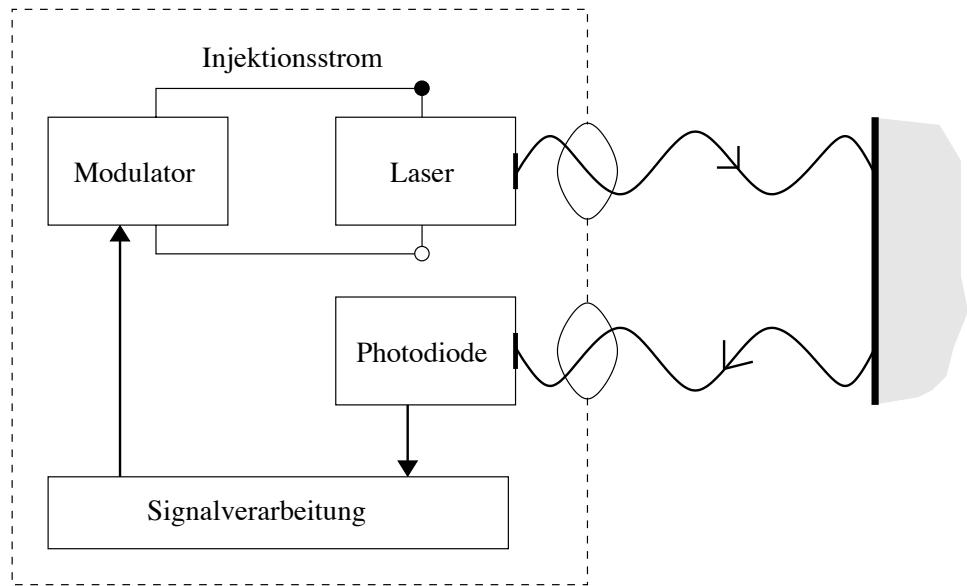


Abbildung 2.1.: Schematischer Aufbau eines LiDAR nach Kern (2003)

Ein LiDAR besteht exemplarisch gemäß Abbildung 2.1 aus einem Light Amplification by Stimulated Emission of Radiation (LASER) als Emitter, einer Photodiode als Detektor, einer Signalverarbeitungseinheit, einem Modulator und einem Gehäuse. Die Laserquelle erzeugt energiereiches Licht, das anschließend das Gehäuse verlässt. Der am Hinder- nis reflektierte Laserstrahl trifft auf den Detektor im Gehäuse, um nach verschiedenen physikalischen Eigenschaften ausgewertet zu werden.

### Funktionsweise eines LASERs

Laserlicht entsteht durch Absorption und spontane sowie stimulierte Emission von Ladungsträgern in einem aktiven Medium (das Material des LASERs).

### Absorption

Absorption bezeichnet den Vorgang, bei dem die Energie eines Photons von einem Elektron aufgenommen wird und es dadurch auf ein höheres Energieniveau gebracht wird.

### Emission

Nach einer gewissen Verweilzeit auf dem höheren Energieniveau fällt das Elektron wieder auf das niedere Energieniveau hinab, wobei ein Photon mit der Energiedifferenz der Niveaus emittiert wird. Dieser Vorgang bezeichnet die spontane Emission. Zusätzlich kann der Übergang gezielt durch den photoelektrischen Effekt ausgelöst werden (stimulierte Emission).

Durch die wechselseitige Absorption und Emission kann es zu einer Verstärkung des Lichts im LASER kommen. Dafür ist es notwendig, dass die resultierende elektromagnetische Welle weniger durch Absorption geschwächt, als durch Emission verstärkt wird. Dies erreicht man durch eine Besetzungsinversion (es befinden sich mehr Teilchen auf dem höheren Energieniveau). Durch konstantes „Pumpen“, zum Beispiel durch Anlegen einer elektrischen Spannung, wird das aktive Medium in der Besetzungsinversion gehalten.

### Aufbau eines LASERs

Um eine Verstärkung des Lichts zu ermöglichen, ist der LASER in Form eines Resonators aufgebaut. Das aktive Medium befindet sich zwischen zwei Spiegeln, an denen die elektromagnetische Welle reflektiert wird.

Das aktive Medium befindet sich zwischen zwei Spiegeln, die im Abstand  $d$  angeordnet sind. Ein Spiegel ist halbdurchlässig, das heißt er reflektiert nur einen Teil der elektromagnetischen Welle und transmittiert den Rest ( $R < 1$ ). Die effektive optische Weglänge des Lichts im Resonator  $L$  ist abhängig vom Brechungskoeffizienten  $n$  und errechnet sich gemäß  $L = n * d$ .

LASER unterscheiden sich im verwendeten aktiven Medium, der erzeugten Energiedichte in Form von Sicherheitsklassen sowie dem Betriebsmodus. Als Betriebsmodi unterscheidet man Dauerlicht- oder Pulsbetrieb. Dauerlichtbetrieb erzeugt einen konstanten Laserstrahl geringer Energiedichte, wohingegen der Pulsbetrieb kurze Lichtimpulse mit hoher Energiedichte ermöglicht. Aktive Medien sind Festkörper, Gase, Farbstoffe und Halbleiter. Sie ermöglichen Laserlicht unterschiedlicher Wellenlängen, Bandbreiten und Energiedichten.

## **Optische Eigenschaften des LASERs**

Durch die exakte Abstimmung des optischen Verstärkers und des Resonators entsteht ein divergentes, monochromatisches Licht hoher Energiedichte.

### **Monochromatisch**

Ein LASER erzeugt Licht einer schmalen spektralen Bandbreite, weswegen es als monochromatisch bezeichnet wird. Die Wellenlänge des Lichts ist dabei abhängig vom verwendeten aktiven Medium. Durch die Kombination verschiedener optischer Effekte können auch mehrere Wellenlängen im LASER erzeugt werden.

### **Zeitliche und räumliche Kohärenz**

Kohärenz bezeichnet die Gleichartigkeit und Gleichförmigkeit mehrerer Lichtwellen. Zeitliche Kohärenz ist bei Laserstrahlen in Form von Interferenzmustern aus Wellenbergen und Wellentälern zu beobachten. Räumliche Kohärenz bezeichnet das Maß, inwieweit sich zwei Wellenzüge aus der gleichen Quelle in unterschiedlichen Richtungen, an unterschiedlichen Orten zur selben Zeit unterscheiden.

### **Divergenz**

Im LASER erzeugtes Licht ist divergent, das heißt, der Durchmesser des Laserstrahls weitet sich relativ zu seiner Ausbreitungsgeschwindigkeit auf. Der Grund besteht in der Beugung des Strahls beim Austritt aus dem Gehäuse. Der Laserstrahl kann durch eine Sammellinse gebündelt werden, wodurch jedoch auch die Intensität des Strahls sinkt.

### **Absorption, Transmission und Reflexion**

Trifft der Laserstrahl auf ein Hindernis, so wird er teilweise vom Hindernis absorbiert (aufgenommen), transmittiert (durchgelassen) und reflektiert (zurückgeworfen). Wie stark sich der jeweilige Effekt auswirkt, ist abhängig von der Beschaffenheit der Hindernisse, dem Winkel zwischen auftreffendem Laserstrahl und Hindernis sowie der Wellenlänge des Laserstrahls.

Gemäß Abbildung 2.2a bezeichnet die Reflexion das Zurückstrahlen auf einer Ebene. Dabei ist der Eintrittswinkel  $\alpha$  gleich dem Austrittswinkel  $\beta$ , gemessen zum Lot der Reflexionsfläche. Daraus resultiert, dass idealisierte parallele Strahlen auch wieder parallel reflektiert werden.

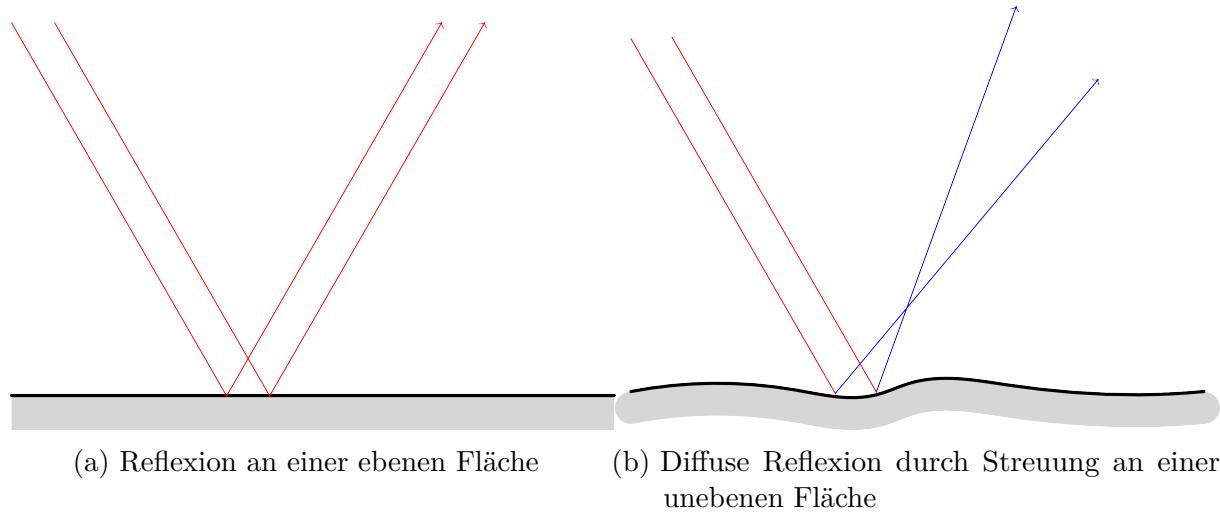


Abbildung 2.2.: Varianten der Reflexion von Licht

Treffen die Lichtstrahlen jedoch auf eine unebene Fläche, so kommt es gemäß Abbildung 2.2b auf Grund von Streuung zur diffusen Reflexion. Die ankommenden, idealisiert-parallelten Strahlen werden normalverteilt von der Oberfläche reflektiert.

Treffen die Lichtstrahlen auf durchlässige Hindernisse, so kommt es an der Oberfläche neben Reflexion auch zur Beugung und Transmission. Die vom LASER ausgesendete elektromagnetische Welle wird somit teilweise von den Hindernissen absorbiert, teilweise zurückgeworfen. Durch die optischen Wechselwirkungen hat die elektromagnetische Welle an Leistung verloren und unter anderem eine Phasenverschiebung erfahren. Die Photodiode am LiDAR wertet anschließend das empfangene Signal gegenüber dem versendeten Signal aus.

### 2.2.1.2. Messen von Entfernungen

LiDAR-Sensoren werden zur Messung von Entfernungen zwischen Sensor und Umgebung verwendet. Hauptsächlich unterscheidet man zwischen der Messung der Signallaufzeit, der Phasenverschiebung und der Triangulation. Durch die Auswertung der Intensität der reflektierten Laserstrahlen können Aussagen über das Material und den Aggregatzustand des gemessenen Hindernisses gemacht werden.

#### Messung der Signallaufzeit

Die Bestimmung von Entfernungen durch Messung der Signallaufzeit entspricht der Funktionsweise des Radio Detection and Ranging (RaDAR). Mit der Auskopplung des Laserstrahls wird am LASER eine Zeitmessung gestartet. Durch Differenzbildung von Austritts- und Eintrittszeitpunkt des Laserstrahls wird gemäß (2.23) die Laufzeit bestimmt.

$$t_{\text{Laufzeit}} = t_{\text{Eintritt}} - t_{\text{Austritt}} \quad (2.23)$$

Da die gemessene Laufzeit den Hin- und Rückweg bezeichnet, hat der Laserstrahl die doppelte Wegstrecke zurückgelegt. Mit der Ausbreitungsgeschwindigkeit von Licht  $c$  wird die halbierte Laufzeit gemäß (2.24) multipliziert.

$$s = \frac{t_{\text{Laufzeit}}}{2} * c \quad (2.24)$$

Die erhaltene Wegstrecke  $s$  ist die Entfernung zwischen dem LiDAR-Sensor und dem Hindernis.

#### Messung der Phasenverschiebung

Die Ausbreitungsgeschwindigkeit von Licht ist abhängig von dem Medium, in dem es sich bewegt. Durch den Zusammenhang zwischen Lichtgeschwindigkeit  $c$  und Wellenlänge  $\lambda$  kann gemäß (2.25) die Frequenz berechnet werden.

$$c = \lambda * f \quad (2.25)$$

Die Frequenz  $f$  ist dabei gemäß (2.26) direkt in die Kreisfrequenz  $\omega$  umformbar. Durch Bildung des Verhältnisses zwischen der gemessenen Phasenverschiebung  $\delta$  und der konstruktionsbedingten Kreisfrequenz  $\omega$  kann gemäß (2.27) die Signallaufzeit des Laserstrahls berechnet werden.

$$\omega = 2 * \pi * f \quad (2.26)$$

$$t_{Laufzeit} = \frac{\delta}{\omega} \quad (2.27)$$

Anhand des vorherigen Abschnitts kann aus der berechneten Laufzeit der Abstand zwischen Hindernis und Sensor bestimmt werden.

### Messung durch Triangulation

Einige LiDAR-Sensoren ermitteln den Abstand zu einem Hindernis mittels Triangulation. Der Detektor eines solchen Systems besteht aus einer Kamera, einer ortsauf lösenden Photodiode oder einer Charge-Coupled Device (CCD)-Zelle. Der emittierte Laserstrahl wird je nach Entfernung in einem unterschiedlichen Winkel auf dem Detektor wahrgenommen, sodass sich auch die Position des Abbildes am Detektor ändert. Dafür wird der Abstand zwischen Emitter und Detektor  $d$  benötigt.

Der Detektor wird so ausgerichtet, dass der Laserstrahl bei einer mittleren Entfernung auch mittig auf dem Sensor detektiert wird. Dieser Winkel zwischen der Ausrichtung des Detektors und der Emitter-Detektor-Achse wird als  $\alpha$  bezeichnet.

Bewegt man das Hindernis, so resultiert die veränderte Distanz in einer Änderung des Auftreffwinkels  $\delta$ . Dieser Winkel ändert sich proportional zur gemessenen Position des reflektierten Laserstrahls im Detektor.

Nun kann der Abstand  $s$  zwischen der Emitter-Detektor-Einheit und dem Hindernis berechnet werden.

$$s = d * \tan(\alpha + \delta) = d * \frac{\tan \alpha + \tan \delta}{1 - \tan \alpha * \tan \delta} \quad (2.28)$$

### 2.2.1.3. Messtechnische Besonderheiten

Für diese Arbeit werden LiDAR-Sensoren verwendet, die Entfernung zu Hindernissen mittels Laufzeitmessung bestimmen. Diese Messvariante unterliegt dabei verschiedenen Einflüssen.

#### Abschattung

Abschattung bezeichnet die Problematik, dass Hindernisse, die sich hinter anderen Hindernissen befinden, nicht durch LiDAR-Sensoren gemessen werden können. Es kann somit nur eine Aussage über die Umgebung bis zum ersten Hindernis gemacht werden.

#### Teildurchlässigkeit

Je nach aktivem Medium werden die Laserstrahlen von Hindernissen, wie Glas oder Wasser, stärker transmittiert, reflektiert oder absorbiert. Bei der Verwendung von einer grünen Laserstrahlung ( $\approx 500 \text{ nm} \rightarrow 600 \text{ nm}$ ) werden die Strahlen an Glas hauptsächlich transmittiert, sodass Hindernisse aus diesem Material nicht detektiert werden können. Für die bathymetrische Vermessung<sup>4</sup> werden aus diesem Grund Feststofflaser wie Nd:YAG (Neodym-dotierter Yttrium-Aluminium-Granat) verwendet, die kollineare Laserpulse zweier Wellenlängen (Rot  $\equiv 1064 \text{ nm}$  für die Wasseroberfläche und Grün  $\equiv 532 \text{ nm}$  für den Untergrund) erzeugen.

#### Messen in Bewegung

LiDAR-Sensoren übertragen ihre Messwerte meist in Datenpaketen. Diese Datenpakete bestehen aus zusammengefassten Messreihen, denen ein Zeitstempel zugeordnet wird. Wird der LiDAR während seiner Messung bewegt, müssen die Zeitstempel zwischen den einzelnen Messungen interpoliert werden, da sonst der geographische Bezug der Datenpunkte verfälscht wird.

---

<sup>4</sup> Bathymetrische Vermessung bezeichnet die Vermessung der topographischen Gestalt von Gewässerbetten, Meeresböden und Seegründen.

### 2.2.2. Messung der inertialen Bewegung und globalen Position

Lage und Position im 3-dimensionalen Raum werden als six degrees of freedom (6DoF) bezeichnet, wobei die 6 Freiheitsgrade in 3 translatorische und 3 rotatorische Freiheitsgrade unterteilt werden. Die translatorischen Freiheitsgrade definieren die Position bezüglich  $x$ -,  $y$ -, und  $z$ -Achse in einem Koordinatensystem. Die rotatorischen Freiheitsgrade bestimmen die Lage an dieser Position durch Rotationen um die Koordinatenachsen. Gemäß Tabelle 2.1 werden die 3 Eulerschen Winkel  $\Phi$  (Phi),  $\Theta$  (Theta) und  $\Psi$  (Psi) verwendet.

Tabelle 2.1: Bezeichnung der Winkel gemäß DIN 9300 / ISO 1151-2:1985.

Bezeichnung	Winkel	Rotationsachse
Rollwinkel (Roll)	$\Phi$	$x''$
Nickwinkel (Pitch)	$\Theta$	$y'$
Gierwinkel (Yaw)	$\Psi$	$z$

Das Bezugssystem ist das World Geodetic System 1984 (WGS84). Dabei handelt es sich um ein geodätisches Referenzsystem, das eine einheitliche Positionsangabe auf der Erde und im erdnahen Weltraum ermöglicht. Die Bestandteile sind ein Referenzellipsoid, ein zusätzliches detailliertes Modell Earth Gravitational Model 1996 (EGM96) gemäß Lemoine et al. (1998) und einem Satz dreidimensionaler Koordinaten von 12 über der Erde verteilten Fundamentalstationen. Das Zentrum des Koordinatensystems ist der Massenschwerpunkt der Erde inklusive der Atmosphäre. Die Orientierung der Achsen ist relativ zum Referenzmeridian und Referenzpol. Die zeitliche Entwicklung (die schrittweise Nummerierung der Längengrade) folgt zudem der Rotation der Erdkruste.

#### 2.2.2.1. Globale Positionsbestimmung durch Satellitensysteme

GNSS werden zur absoluten Positionsbestimmung im 3-dimensionalen Raum nach dem WGS84 eingesetzt. Die bekanntesten eingesetzten Systeme sind das vom russischen Vertei-

digungsministerium eingesetzte Globalnaja nawigazionnaja sputnikowaja sistema (russisch für Globales Satellitennavigationssystem) (GLONASS), das chinesische BeiDou, Galileo der European Space Agency (ESA) und der Europäischen Union und das amerikanische Navigational Satellite Timing and Ranging – Global Positioning System (NAVSTAR-GPS). In dieser Arbeit wird das NAVSTAR-GPS verwendet, dessen Aufbau sowie die Funktionsweise im Folgenden erläutert werden.

## Aufbau

Das NAVSTAR-GPS-System besteht gemäß The United States Government (1995) aus einem Weltraumsegment, einem Kontrollsegment und einem Benutzersegment.

### Das Weltraumsegment

Das Weltraumsegment umfasst 24 bis 32 Satelliten, die in einer mittleren Bahnhöhe von 20.200 km zweimal pro Sterntag<sup>5</sup> um die Erde kreisen. Die Satelliten sind dabei so verteilt, dass überall auf der Erde und zu jeder Zeit ein Global Positioning System (GPS)-Empfänger die Signale von mindestens vier Satelliten empfangen kann. Dabei sind die Satelliten um 55° gegen die Äquatorebene inkliniert (geneigt) und gegeneinander um jeweils 60° verdreht. Jeder Satellit sendet kontinuierlich ein Datensignal von insgesamt 37.500 Bit mit einer Datenrate von 50 Bit/s. Die vollständige Übertragung benötigt somit circa 12,5 Minuten. Als gemeinsame Zeitbasis werden in den einzelnen Satelliten mehrere Atomuhren eingesetzt, die es ermöglichen, dass die Satelliten bis zu  $10^{-14}$  Sekunden synchron laufen.

Das Datensignal setzt sich dabei aus 25 Frames zusammen. Die Frames sind wiederum in fünf Subframes unterteilt, deren erstes Wort das *telemetry word* ist. Das *telemetry word* enthält Informationen zur Aktualität der Ephemeridendaten.<sup>6</sup> Es folgt das *hand over word*, das die Anzahl der sogenannten Z-Epochen seit dem letzten Sonntag 0 Uhr enthält.

---

<sup>5</sup> Bei einem Sterntag handelt es sich um den Zeitraum zwischen zwei oberen Kulminationen des Frühlingspunkts, verursacht durch die Eigendrehung der Erde. Ausgedrückt in Système international d'unités (französisch für Internationales Einheitensystem) (SI)-Einheiten beträgt ein Sterntag 23 Stunden, 56 Minuten und 4,091 Sekunden.

<sup>6</sup> Ephemeriden sind Positionsdaten, Bahndaten sich bewegender astronomischer Objekte.

Nach diesen zwei Wörtern folgen im ersten Subframe die Daten zur Genauigkeit, dem Zustand und den Uhrenkorrekturwerten des sendenden Satelliten. Im zweiten und dritten Subframe werden die Ephemeridendaten übertragen. Das vierte Subframe enthält das sogenannte Refraktionsmodell sowie die Almanachdaten<sup>7</sup> der Satelliten 25 bis 32. Das Refraktionsmodell besteht aus Ionosphärenkorrekturdaten, speziellen Nachrichten sowie Coordinated Universal Time (englisch für koordinierte Weltzeit) (UTC)-Zeitinformationen. Das letzte Subframe enthält die Almanachdaten der Satelliten 1 bis 24 sowie die Uhrzeit des Satelliten und die GPS-Wochennummer. Die GPS-Zeit unterscheidet sich dabei von der UTC-Zeit. Die UTC-Zeit wird mit Hilfe von Schaltsekunden regelmäßig der Erddrehung angepasst. Die GPS-Zeit nicht, sodass sie sich aktuell circa 19 Sekunden unterscheiden.

### **Das Kontrollsegment**

Das Kontrollsegment dient der Überwachung des Weltraumsegments. Jeder Satellit wird bezüglich seiner Bewegung und seines technischen Zustandes in Kontrollzentren auf der Erde beobachtet, die Ephemeriden- sowie die Almanachdaten berechnet und die Zeiten der Satelliten synchronisiert. Diese neuen Informationen werden im Anschluss an alle in Kontakt stehenden Satelliten übertragen (bei NAVSTAR-GPS alle 2 Stunden). Befindet sich ein Satellit nicht auf seiner gewünschten Umlaufbahn, wird der Satellit als *not good* markiert, die Triebwerke werden gezündet und der Satellit wird neu positioniert. Sobald sich die Umlaufbahn des Satelliten wieder stabilisiert hat, wird der Status wieder auf *good* gesetzt.

### **Das Benutzersegment**

Das Benutzersegment bezeichnet den GPS-Empfänger der Satellitendaten. Er besteht aus mindestens einer Antenne, einer Berechnungseinheit sowie, aus Kostengründen und Machbarkeit, aus einer einfachen Quarzuhr. Der Empfänger dient der Berechnung der

---

<sup>7</sup> Im Vergleich zu den Ephemeriden enthält der Almanach weniger genaue, aber dafür länger gültige Bahndaten, die um Informationen über die Integrität der ausgestrahlten Signale (technischer Zustand, momentane Konfiguration und die Identifikationsnummer) ergänzt sind.

---

globalen Position, die durch Auswertung der Satellitensignale bestimmt wird. Satelliten, die vom Kontrollsegment als *not good* markiert wurden, werden von den GPS-Empfängern ignoriert, da sonst die Betrachtung ihrer Daten zu Berechnungsfehlern führt.

Wurde der Empfänger komplett von der Spannungsversorgung getrennt, spricht man von einem *cold start*. Dabei muss die interne ungenaue Uhr mit der Satellitenzeit synchronisiert werden. Im Anschluss werden alle Almanach- und Ephemeridendaten empfangen, was im schlimmsten Fall auf Grund der Datenrate bis zu 12,5 Minuten dauern kann. Durch Auswertung der Almanachdaten und der Vorgabe einer groben Position durch den Anwender kann die Zeit bis zur Positionsbestimmung reduziert werden.

War der Empfänger weniger als 2 Stunden außer Betrieb, mit einer Spannungsversorgung verbunden und weniger als 300 km von der letzten bekannten Position entfernt, so spricht man von einem *hot start*. In diesem Modus werden Daten von bereits bekannten Satelliten empfangen. Treffen die Bedingungen für einen *hot* oder *cold start* nicht zu, spricht man von einem *warm start*.

## Funktionsweise

Die Grundlage der Positionsbestimmung mittels GPS ist die Entfernungsmessung zwischen einem Empfänger zu mehreren Sendern. Die Sender sind dabei die Satelliten, die auf höheren Erdumlaufbahnen kreisen. Das Verfahren zur Positionsbestimmung heißt Trilateration. Die Position der Satelliten ändert sich auf Grund der stetigen Bewegung auf ihren Umlaufbahnen. Somit ist auch die Position der Satelliten im Verhältnis zum Empfänger stets unterschiedlich. Durch Betrachtung der Ephemeriden- und Almanachdaten kann der Empfänger die Satellitenstandorte zu jedem Zeitpunkt berechnen. Bei NAVSTAR-GPS und Galileo handelt es sich bei den Ephemeriden um keplersche Bahnelemente<sup>8</sup>, wohingegen es bei GLONASS Koordinaten-, Geschwindigkeits- und Beschleunigungsvektoren sind.

Die Satelliten vermerken den Sendezeitpunkt einer Übertragung im GPS-Zeitsystem,

---

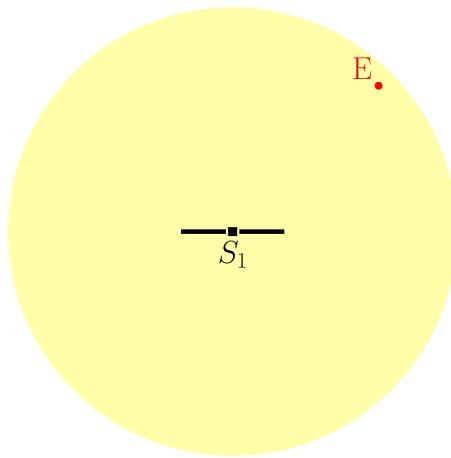
<sup>8</sup> Keplersche Bahnelemente beschreiben die Bewegung eines astronomischen Objekts gemäß den Kepler-schen Gesetzen im Schwerkraftfeld eines Himmelskörpers.

beim Empfang setzt das Bodensegment den Empfangszeitpunkt. Durch Differenzbildung von Empfangs- und Sendezeitpunkt kann theoretisch durch Betrachtung der Lichtgeschwindigkeit die Entfernung zwischen Satellit und Empfänger bestimmt werden. Diese momentane Entfernung ist auf Grund von Fehlern der Uhren (Empfänger und Satellit) und weiteren Einflüssen wie atmosphärischen Eigenschaften ungenau.

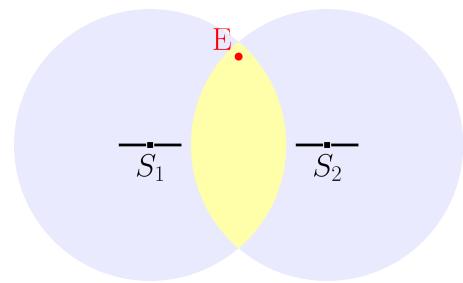
Abbildung 2.3 zeigt die Auswirkungen, die die Anzahl der Satellitenanzahl auf die Positionsbestimmung bei GNSS hat. Ein einzelner Satellit reicht gemäß Abbildung 2.3a nur zur Positionsbestimmung des Empfängers auf einer Kugeloberfläche rund um die aktuelle Position des Satelliten aus. Durch einen zweiten Satelliten wird die potenzielle Position des Empfängers auf den Umkreis des Schnitts der zwei Satellitenkugeln reduziert. Trotz eines dritten Satelliten ist die Positionsbestimmung im Schnittpunkt aller Kugelschnitte nicht eindeutig, da der Empfänger meist mit einer ungenauerer Uhr ausgestattet ist, als die Satelliten.

Die Empfängeruhr wird durch sogenanntes *Pseudoranging* synchronisiert. Dabei werden Pseudostrecken herangezogen, die von den wahren Distanzen um einen konstanten, aber vorerst unbekannten Betrag abweichen. Im 2-dimensionalen Raum werden dazu durch den Schnitt von zwei Bögen gemäß Abbildung 2.3b zwei Schnittpunkte gefunden. Einer der beiden Schnittpunkte kann aus Grund der Plausibilität vernachlässigt werden. Im 3-dimensionalen Fall handelt es sich dabei nicht um zwei Schnittpunkte, sondern den Umkreis, der durch den Schnitt zweier Kugeln entsteht. Weichen nun beide Bögen im Radius um den gleichen konstanten aber unbekannten Wert ab, werden die Schnittpunkte fehlerhaft. Man spricht dann von einem schleifenden Schnitt. Durch die Messung eines dritten, um den gleichen konstanten unbekannten Wert abweichenden Bogens erhält man gemäß Abbildung 2.3c ein krummliniges Fehlerdreieck, dessen Mittelpunkt der gesuchten Position entspricht. Analog zum 2-dimensionalen Fall benötigt man demzufolge im 3-dimensionalen Raum gemäß Abbildung 2.3d 4 Bögen beziehungsweise Satelliten.

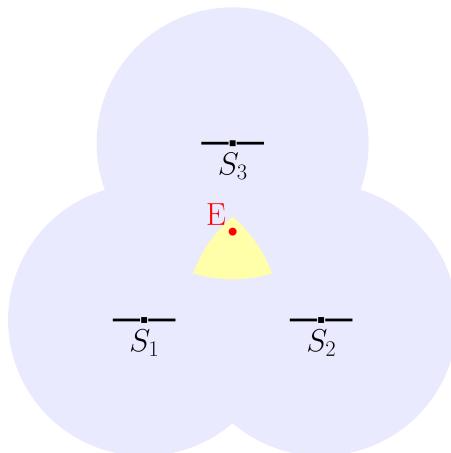
Durch die Verwendung von Differential Global Positioning System (DGPS) kann die Genauigkeit von GPS gesteigert werden. Dabei werden Korrekturdaten für das Bahn- und



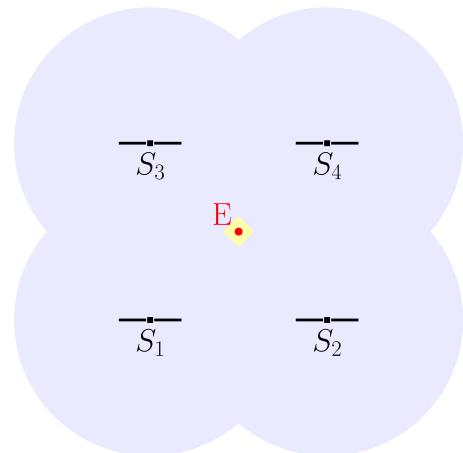
(a) Variation auf der Kugeloberfläche



(b) Variation auf dem Umkreis der Schnittfläche



(c) Ungefähre Position auf der Schnittfläche



(d) Positionsbestimmung durch Zeitkorrektur

Abbildung 2.3.: Auswirkung der Satellitenzahl auf die Positionsbestimmung bei GNSS  
(orange markiert die möglichen Positionen des Empfängers E und S markiert den/die nummerierten Satelliten)

Zeitsystem durch vorher vermessene, ortsfeste Referenzstationen ausgestrahlt und bei kompatiblen Empfängern zur Berechnung einbezogen.

Besitzt der Empfänger eine zusätzliche GPS-Antenne, die in einem bekannten Abstand montiert ist, so können zusätzliche Winkelinformationen (das Heading) berechnet werden. Durch Auswertung der Phasenverschiebung und des Antennenabstandes wird der Winkel zwischen Antennenverbindungsachse zu den Ephemeriden des Satelliten sowie gegen die Nordrichtung bestimmt. Geräte dieser Kategorie bezeichnet man als elektronischer Kompass oder GPS-Kompass.

### **Fehlerbetrachtung**

Die Genauigkeit der Positionsbestimmung mittels GPS unterliegt verschiedenen physikalischen und politischen Einflüssen.

### **Selective Availability**

Bis zum 02. Mai 2000 wurde gemäß The National Coordination Office for Space-Based Positioning, Navigation, and Timing (2013) durch die *selective availability* ein künstlicher Fehler zugeschaltet. Dabei wurde das Taktsignal einem Jitter (eine geringe zeitliche Schwankung) ausgesetzt. Dies resultiert in einem Fehler der Positionsbestimmung von mehreren 100 m.

### **Satellitengeometrie**

Die Positionierung zwischen Empfänger zu den empfangenen Satelliten bezeichnet man als Satellitengeometrie. Befinden sich die Satelliten in hauptsächlich einer Richtung, vergrößert sich der Bereich der potenziellen Positionen des Empfängers. Dieses Problem tritt zum Beispiel bei konstanter Abschirmung der Signale aus einer bestimmten Richtung auf. Die Positionsbestimmung ist dabei entweder gar nicht oder nur auf bis zu 150 m möglich. Als *Güte* der Satellitengeometrie sind im Allgemeinen die Dilution Of Precision (DOP)-Werte (Verschlechterung der Genauigkeit) sehr verbreitet. Abhängig von den zur Berechnung herangezogenen Daten unterscheidet man gemäß Tabelle 2.2 verschiedene

Güte-Kategorien.

Tabelle 2.2: Kategorien verschiedener DOP-Werte nach Köhne/Wößner (2008).

DOP-Wert	Genauigkeit	herangezogene Daten
<i>Geometric Dilution Of Precision (GDOP)</i>	Gesamt	3D-Koordinaten + Zeit
<i>Positional Dilution Of Precision (PDOP)</i>	Position	3D-Koordinaten
<i>Horizontal Dilution Of Precision (HDOP)</i>	Horizontal	2D-Koordinaten
<i>Vertical Dilution Of Precision (VDOP)</i>	Vertikal	Höhe
<i>Time Dilution Of Precision (TDOP)</i>	Zeit	Zeit

### Satellitenumlaufbahnen

Trotz regelmäßiger Korrektur der Ephemeridendaten durch das Kontrollsegment kommt es durch Gravitation von Sonne und Mond zu Bahnfehlern. Diese wirken sich mit einer Abweichung von circa zwei Metern auf die Genauigkeit der Positionsbestimmung aus.

### Mehrwegeausbreitung

Durch lokale Abschirmung unter Bäumen, unter Brücken, in oder zwischen Gebäuden wird zudem das Satellitensignal gestört, geblockt oder gelangt durch Mehrwegeausbreitung zum Empfänger. Die reflektierten Signale legen eine längere Strecke als Signale im Direktweg zurück und kommen zeitlich versetzt am Empfänger an. Durch Reflexionen ändert sich die Polarisation des Satellitensignals, sodass der Effekt durch eine geeignete Konstruktion der Antennen minimiert werden kann.

### Atmosphärische Eigenschaften

Einen weiteren Einfluss auf die Genauigkeit hat der Zustand der Ionosphäre und der Troposphäre. Je nach Elevationswinkel des Satelliten wird eine größere Strecke durch die Schichten der Atmosphäre zurückgelegt, was zum Beispiel Effekte der Refraktion verstärkt. Zusätzlich verursacht die Schwankungsbreite der Anzahl der freien Elektronen in der

Ionosphäre einen Fehler von circa 30 m in der Positionsbestimmung. Durch Auswertung der 6 Ionosphärenparameter der Satelliten kann der Fehler auf unter 10 m reduziert werden. Durch die Verwendung zweier unterschiedlicher Frequenzen zur Übertragung der Satellitensignale kann zudem die Einwirkung der Atmosphäre herausgerechnet werden.

### **Zeitungenaugkeiten und Rundungsfehler**

Trotz Synchronisation der Uhr des Empfängers bleibt die Ungenauigkeit der Satellitenuhr. Durch zusätzliche Rundungsfehler im Empfänger steigt diese Ungenauigkeit um 1 - 3 m.

Zusammenfassend kann man sagen, dass die Bestimmung der Position mittels GPS bis zu circa 12 m genau ist. Durch die Verwendung von DGPS wird die Genauigkeit auf 0,3 - 2,5 m für die Position  $(x, y)$  und 0,2 - 5 m für die Höhe gesteigert. Durch die zusätzliche Auswertung der Phasenverschiebung kann die Genauigkeit auf 1 - 10 mm pro km Abstand zur Referenzanlage verbessert werden.

Durch die Verwendung einer sogenannten Real-Time-Kinematic (RTK)-Lösung werden Positionsdaten zwischen einer Referenzantenne zur bewegten Antenne (Rover) mittels einer Mobilfunkverbindung übertragen. Diese Daten ermöglichen eine Positionsgenauigkeit  $(x, y)$  von 1 - 2 cm und einer Höhengenauigkeit von circa 2 - 3 cm.

#### **2.2.2.2. Bestimmung von Lage und Ausrichtung durch Inertiale Messeinheiten**

Inertialsensoren messen anhand der Trägheit von intern verbauten, quantitativ bekannten Massen die durch eine Bewegung einwirkende Beschleunigung und Drehrate. Durch die Kombination mehrerer Inertialsensoren in einer IMU können so die Bewegungen in allen 6DoF bestimmt werden.

### **Aufbau**

Grundsätzlich besteht eine IMU aus Beschleunigungs- und Drehratensensoren. Die Beschleunigungssensoren messen die Beschleunigung entlang 3 orthogonal zueinander stehenden Hauptachsen  $x$ ,  $y$  und  $z$ . Durch die Drehratensensoren können zudem die Rotationen um diese Hauptachsen gemessen werden.

Ursprünglich verwendete man seismische Massen. Eine seismische Masse bezeichnet eine in der Messachse beweglich aufgehängte träge Masse. Auf Grund der Massenträgheit wurden die Bewegungsänderungen der Massen gemessen. In Folge der Miniaturisierung verwendet man heutzutage sogenannte mikro-elektrisch-mechanische Systeme (MEMS). Dabei handelt es sich um gefedert aufgehängte Siliziumstege im  $\mu\text{m}$  Bereich. Bei einer Beschleunigung werden die Stege ausgelenkt, was in der Änderung der elektrischen Kapazität zu einer festen Bezugselektrode messbar ist.

### Funktionsweise

Inertialsensoren bestimmen den Betrag und die Richtung der Beschleunigung  $a(t)$  eines Massepunktes  $m$  im Raum. Gemäß des zweiten Newtonschen Gesetzes (Gleichung (2.29)) wird bei festgelegten Anfangsbedingungen durch einfache Integration über die Zeit die Geschwindigkeit  $v(t)$  bestimmt. Durch die zweifache Integration erhält man die Positionsänderung  $s(t)$ .

$$\sum_J F^J = m * a(t) = m * \frac{\partial v}{\partial t} = m * \frac{\partial^2 s}{\partial t^2} \quad (2.29)$$

Sind die Anfangsgeschwindigkeit sowie der Ausgangspunkt bekannt, so folgt durch die Integration ein absoluter Ort nach der Verschiebung des Sensors.

Durch Drehratensensoren wird die Winkelgeschwindigkeit entlang einer Drehachse gemessen. Die Winkelgeschwindigkeit kann dabei je nach verwendetem Medium anhand zweier Messprinzipien ermittelt werden:

- die auf ein mechanisch bewegtes System wirkende Corioliskraft und
- den Sagnac-Effekt bei optischen Systemen.

Die Corioliskraft bezeichnet gemäß Kuchling (2011) die Kraft, die auf einen sich bewegenden Körper auf einem rotierenden Bezugssystem wirkt. Der bewegte Körper beschreibt trotz radialer Bewegung (nach innen oder außen) eine Bahnkurve, die auf seiner Trägheit beruht.

Bei dem Sagnac-Effekt handelt es sich um die Beobachtung einer relativistischen Eigenschaft. Gemäß Demtröder (2013) wird bei einem Sagnac-Interferometer eine ebene Lichtwelle in zwei Teilwellen aufgeteilt, durch Spiegel um eine Fläche  $A$  umgelenkt und anschließend wieder überlagert. Im Ruhezustand sind die Strecken für beide Teilwellen gleich lang, wodurch am Detektor die maximale Intensität gemessen wird. Rotiert man nun das gesamte Interferometer, verlängert sich die zurückzulegende Strecke für eine Teilwelle, für die andere verkürzt sie sich. Das Resultat ist eine Phasendifferenz  $\Delta\phi$  zwischen den beiden Teilwellen und eine Änderung in der Intensität  $I$  im Überlagerungsgebiet.

Gemäß Gleichung 2.30 ist die zu messende Intensität  $I(\Delta\phi)$  abhängig von der Phasendifferenz  $\Delta\phi$ .

$$I(\Delta\phi) = \frac{1}{2} * I_0 * (1 + \cos \Delta\phi) \quad (2.30)$$

Die Phasendifferenz wiederum ist gemäß Gleichung 2.31 abhängig vom Winkel zwischen der Flächennormalen und der Drehachse  $\Theta$ , der Kreisfrequenz der Drehung  $\Omega$ , der zu umlaufenden Fläche  $A$  und der Wellenlänge des verwendeten Lichts  $\lambda$ .

$$\Delta\phi = \frac{8 * \pi * A}{c * \lambda} * \Omega * \cos \Theta \quad (2.31)$$

Durch das Experiment von Michelson (1925) konnte mit dem Sagnac-Interferometer die Rotation der Erde nachgewiesen werden. Laser-Kreisel bieten heutzutage bereits auf kleineren Flächen die Möglichkeit, Rotationen mithilfe des Sagnac-Effekts zu messen. Aus messtechnischen Gründen wurde das Messverfahren abgewandelt.

Durch die Interferenz von Teilauskopplungen beider Strahlen kann man eine Differenzfrequenz als Schwebung beobachten, das heißt, die Interferenzstreifen wandern je nach Drehrichtung nach links oder rechts. Durch Messung der Amplitudenmodulation der interferierenden Strahlen mit einer Fotodiode kann die Differenzfrequenz der Strahlen berechnet werden.

## Fehlerbetrachtung

Der größte Fehler bei einer IMU entsteht durch den sogenannten Kreiseldrift. Mechanische Unwuchten am Kreisel führen zu einem unvermeidlichen Drift des Kreisels. Diese Unwuchten im Rotor (Kreiselkörper) haben teilweise ihren Ursprung in Fertigungstoleranzen und sind abhängig von der Umgebung des Kreisels.

### **Präzession**

Wirkt eine mechanische Kraft ein senkrecht Drehmoment auf die Rotationsachse des Rotors aus, so wird die Kreiselachse abgelenkt, es kommt zur Präzession.

### **Nutation**

Wirkt ein Impuls auf einen rotierenden Kreisel, der nicht parallel zum minimalen oder maximalen Trägheitsmoment ausgerichtet ist, vollführt der Kreisel eine Nickbewegung. Diese Nickbewegung bezeichnet man als Nutation.

Durch Schwankungen der Temperatur, Wechselwirkungen mit den Beschleunigungssensoren, Lagerreibung oder mechanische Stöße sowie Vibrationen wird die Messung zudem erschwert.

Bei der Verwendung von Laser-Kreiseln kommt es zu einem weiteren Fehler, der als Lock-In-Effekt bezeichnet wird. An jedem Spiegel treten neben der Reflexion und Transmission auch Streuung auf, wodurch ein gewisser Teil des gestreuten Lichts entgegen der Umlaufrichtung eingekoppelt wird. Bei kleinen Drehraten resultiert daraus, dass beide Laserschwingungen die gleiche Frequenz haben. Dieser lineare Fehler ist abhängig von der Stärke der Streuung, die durch Staubteilchen und Temperaturschwankungen im System gesteigert wird.

Es ist zu erkennen, dass der Großteil der Fehler einer IMU konstruktionsbedingt ist. Nach Edwan (2013) unterteilt man IMU's gemäß den Tabellen 2.3 und 2.4 in die Produktkategorien, in denen sie hauptsächlich verwendet werden. Die Kategorisierung ist dabei abhängig von der Genauigkeit, die für die Anwendung gefordert wird.

Tabelle 2.3: Kategorisierung von Drehratensensoren in Inertialen Messeinheiten.

Kategorie	Bias Stabilität $\left[\frac{\circ}{h}\right]$	Skalierungsfaktor [PPM]	Grundrauschen $\left[\frac{\frac{\circ}{h}}{\sqrt{Hz}}\right]$
<b>Consumer</b>	> 200	–	180
<b>Automotive</b>	10 – 200	> 500	180
<b>Tactical</b>	1 – 10	200 – 500	12 – 30
<b>Navigation I</b>	0.1 – 10	100 – 200	3 – 12
<b>Navigation II</b>	< 0.01	5 – 50	0.12 – 0.3

Tabelle 2.4: Kategorisierung von Beschleunigungssensoren in Inertialen Messeinheiten.

Kategorie	Bias Stabilität [ $\mu g$ ]	Skalierungsfaktor [PPM]	Grundrauschen $\left[\frac{\mu g}{\sqrt{Hz}}\right]$
<b>Consumer</b>	2400	–	1000
<b>Automotive</b>	1200	> 1000	1000
<b>Tactical</b>	200 – 500	400 – 1000	200 – 400
<b>Navigation I</b>	50 – 100	100 – 200	50
<b>Navigation II</b>	5 – 10	10 – 20	5 – 10

### 2.2.2.3. Verbesserung der Genauigkeit durch Koppelnavigation

Koppelnavigation bezeichnet die kontinuierliche Ortsbestimmung eines bewegten Objekts auf Grund der Bewegungsrichtung (Kurs) und Geschwindigkeit (Fahrt). Durch die Kombination von GPS und einer IMU werden die Vor- und Nachteile der einzelnen Systeme ausgeglichen und eine kontinuierliche Bestimmung der Pose ermöglicht.

Zum Einschaltzeitpunkt ermöglicht das GPS auf Grund von veralteten Ephemeriden keine absolute Positionsbestimmung, durch die Verwendung der IMU kann jedoch eine relative Positionsbestimmung durch Bewegung durchgeführt werden. Das GPS bietet

sekündlich neue Positionsinformationen, während dazwischen mit Hilfe der IMU die Position interpoliert wird.

Durch den unvermeidlichen Drift eines Kreisels weicht der gemessene Kurs vom realen Kurs über die Zeit ab. Heutige Kreiselsysteme bieten einen minimalen Drift von circa  $0.01^\circ$  pro Stunde. Auf Grund dieser Abweichung rechnet man mit einer Abweichung von 1 Seemeile pro Stunde. Durch die regelmäßige Positionsinformation durch das GPS kann der Drift der IMU in regelmäßigen zeitlichen Abständen korrigiert werden.

Zur Kopplung der Sensordaten werden gemäß Benzerrouk et al. (2013) Kalman-Filter verwendet. Ein Kalman-Filter ist dabei ein Satz mathematischer Gleichungen, mit denen Störungen, die durch Messgeräte entstehen, entfernt werden. Der Kalman-Filter ist dabei auf die Messfehler und auf die mathematische Struktur des zugrundeliegenden dynamischen Systems angepasst. Diese Kombination aus optimierten GPS- und IMU-Sensordaten wird im Folgenden als Navigations (NAV)-Lösung beziehungsweise NAV-Sensorik bezeichnet.

## 2.3. Datenstrukturen

Datenstrukturen dienen der geordneten Speicherung von einzelnen vorliegenden Daten in einer zusammenhängenden Struktur. Dadurch können die Daten in einem gemeinsamen Zusammenhang verglichen und je nach Implementierung effizient auf Speicherbereiche aufgeteilt werden. Für die vorliegende Arbeit werden Datenstrukturen wie Punkte, Punktwolken und Suchbäume aus der Point-Cloud-Library (PCL) verwendet und erweitert.

Die PCL<sup>9</sup> beinhaltet verschiedene Algorithmen zur Filterung, Feature-Bestimmung, Oberflächen-Rekonstruktion, Registrierung, Modell-Anpassungen und Segmentierung von Punktwolken. Durch die Verwendung der Bibliotheken Boost<sup>10</sup>, FLANN<sup>11</sup>, Eigen<sup>12</sup> und VTK<sup>13</sup> werden zudem verschiedene mathematische Operationen und Möglichkeiten zur

---

<sup>9</sup> Rusu/Cousins (2011).

<sup>10</sup> Dawes et al. (1998).

<sup>11</sup> Muja/Lowe (2008), Muja/Lowe (2012), Muja/Lowe (2014).

<sup>12</sup> Guennebaud et al. (2010).

<sup>13</sup> Schroeder et al. (2004).

Visualisierung der Punktwolken zur Verfügung gestellt.

### 2.3.1. Punkte

Punkte bezeichnen die kleinste Datenstruktur und repräsentieren zum Beispiel eine mehrdimensionale Position oder eine Eigenschaft an einer bestimmten Stelle des Messraumes. Gemäß Delmerico (2013) bietet die PCL verschiedene standardmäßig implementierte Typen von Punkten. Punkte bieten somit die Möglichkeit, Koordinaten, Farbinformationen, Normalenvektoren oder Intensitätswerte zu speichern. Für die Verwendung im DIP wurde eine eigene Punkt-Klasse angelegt, die neben den Koordinaten  $x$ ,  $y$  und  $z$  auch die Speicherung des Normalenvektors sowie einer Ursprungskoordinate ermöglicht. Da die Position des Sensors während der Aufnahme eines Scans nicht konstant ist, wird die Position des Sensors zum Zeitpunkt der Aufnahme des Messpunktes als Ursprungskoordinate gespeichert. Dadurch ist es möglich, die Bewegung des Sensors bei der Verortung der Messpunkte zu berücksichtigen.

### 2.3.2. Punktwolken

Eine Punktwolke dient der Repräsentation einer Zusammenstellung von mehreren Punkten. Punktwolken werden meist durch Sensoren wie Red-Green-Blue and Distance (RGB-D)-Kameras, Stereokameras, LiDAR-Sensor, ToF-Kameras oder durch Software wie Blender generiert. Gemäß den implementierten Punktwolken in der PCL bestehen Punktwolken aus einem Vektor von Punkten, einer Breite (`width`), einer Höhe (`height`) sowie den Parametern `is_dense`, `sensor_orientation` und `sensor_origin`. Handelt es sich bei der vorliegenden Punktwolke um organisierte Daten, so entspricht die Breite der Zahl der Spalten und die Höhe der Zahl der Zeilen. Bei unorganisierten Daten entspricht die Breite der Anzahl an Punkten in der Punktwolke und die Höhe hat den Wert „1“.

Enthält die Punktwolke nur finite Messwerte, also keine `NaN`<sup>14</sup> oder `Inf`<sup>15</sup>, so ist der Wert `is_dense` wahr. Die Eigenschaften `sensor_orientation` und `sensor_origin` können als Parameter für den Sensor verwendet werden, was jedoch nur im stationären Zustand während der Aufnahme der Punktwolke Anwendung findet.

### 2.3.3. Suchbäume

Um mehrere Punktwolken effizient miteinander zu vergleichen, werden die Punkte in sogenannten Suchbäumen abgespeichert. Ein Suchbaum wird dabei entsprechend eines Schemas aufgebaut und ist abhängig von der darauf anzuwendenden Suchstrategie. Für den ICP-Algorithmus werden die Punktwolken iterativ anhand der nächsten Nachbarn der einzelnen Punkte registriert. Um diese Suche zu optimieren, werden die Punkte in einem *kd*-Baum gespeichert. Gemäß Bentley (1975) ist ein *kd*-Baum ein mehrdimensionaler (*k*-dimensional) Binärbaum. Ein *kd*-Baum ermöglicht somit die orthogonale Bereichsanfrage, wie sie zum Beispiel bei der Suche nach den nächsten Nachbarn eines Punktes vorliegt.

## 2.4. Bezugssysteme

Das DLR erforscht die Verwendung von LiDAR-Sensoren zur Erstellung digitaler Karten und als Grundlage zur automatisierten Navigation. Dabei werden Navigationssensoren eingesetzt, die eine Bestimmung der Lage und Position im 3-dimensionalen Raum ermöglichen. Zur Verarbeitung der Sensordaten werden unterschiedliche Bezugssysteme verwendet. Dabei handelt es sich um kartesische Koordinatensysteme, die unterschiedlich ausgerichtet und positioniert sind.

---

<sup>14</sup> In der Informationstechnik bezeichnet `NaN` (Not a Number) einen numerischen Datentyp-Wert, der einem undefinierten oder nicht darstellbaren Wert entspricht.

<sup>15</sup> `Inf` (Infinity) bezeichnet die Darstellung der Unendlichkeit eines Wertes.

### 2.4.1. Sensor

Jeder Sensor misst eine physikalische Eigenschaft in einem speziellen, dafür vorgesehen Koordinatensystem. Bei einem NAV-Sensor ist dies eine zeitliche Abfolge von Positions- und Lagevektoren. Bildgebende Sensoren wie LiDAR-Sensoren speichern ihre Messdaten in polaren Kugelkoordinaten ab. Die Kugelkoordinaten werden in ein kartesisches Koordinatensystem überführt und als PCL-Punktwolken gespeichert.

### 2.4.2. Träger

Die Sensoren sind auf sogenannten Trägern montiert. Je nach Konstruktion kann ein UA mehrere Träger transportieren. In der vorliegenden Arbeit befinden sich die verwendeten bildgebenden und NAV-Sensoren auf einem gemeinsamen Träger, der fest mit dem UA verbunden ist (nicht beweglich). In diesem Fall entspricht die Bewegung des Trägers auch der Bewegung des UA. Die NAV-Sensoren befinden sich im Massenschwerpunkt des UA. Dadurch kann die Bewegung des UA in alle Richtungen gleichmäßig gemessen werden. Das Trägerbezugssystem hat seinen Ursprung im Massenschwerpunkt des Trägers und ist gemäß dem North-East-Down (NED)-Standard ausgerichtet. Gemäß Abbildung 2.4 zeigt die  $x$ -Achse in Hauptflugrichtung entlang der Längsachse des UA, die  $y$ -Achse orthogonal nach rechts und die  $z$ -Achse nach unten. Alle Rotationen werden gemäß der Deutschen Industrienorm (DIN) 9300 (siehe Tabelle 2.1) in der Reihenfolge  $z$ ,  $y'$  und anschließend  $x''$  durchgeführt. Dies entspricht der Konvention in der Luft- und Raumfahrt.<sup>16</sup>

### 2.4.3. Welt

Das Weltbezugssystem wird unterschieden in ein relatives und ein absolutes Bezugssystem. Das relative betrachtet die Position des Trägers von der Startposition aus, die mit  $(0, 0, 0)$  initialisiert wird. Das absolute Bezugssystem entspricht dem WGS84 nach dem Earth-

---

<sup>16</sup> In der Automobilbranche werden die Rotationen und Koordinatensysteme anders ausgerichtet. Dies ist in der DIN 70000 erläutert.

Centered Earth-Fixed (ECEF), wobei die Startposition mit der zum Start befindlichen Position in Längen- und Breitengraden initialisiert wird.

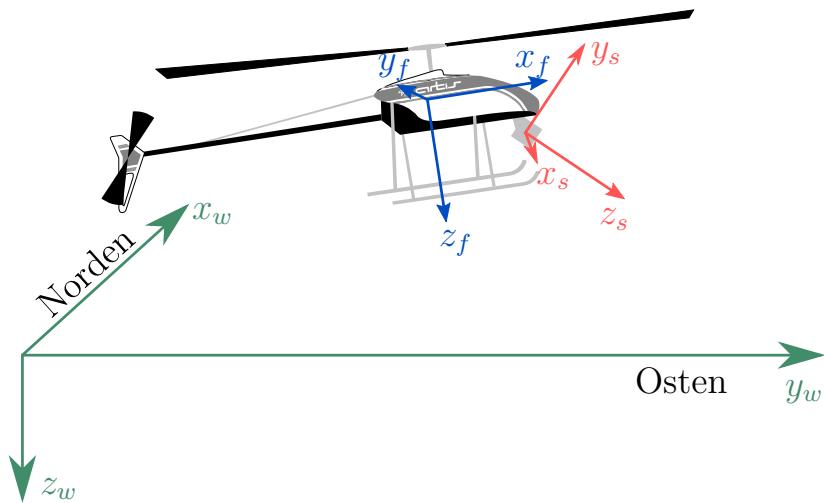


Abbildung 2.4.: Darstellung der verschiedenen Bezugssysteme (Andert (2011))

Da das Sensorbezugssystem keinen Bezug zu einer Posenänderung hat, kann eine digitale Karte nur im Träger- oder Weltbezugssystem erstellt werden. Damit die aufgenommenen Punktwolken korrekt in die Karte eingetragen werden können, müssen sie in das entsprechende Bezugssystem transformiert werden. Diese Transformation ist abhängig von der Montagepose des LiDAR in Relation zur IMU und wird bisher durch eine manuelle extrinsische Kalibrierung bestimmt.

### 3. Relative extrinsische Kalibrierung von bildgebenden Umweltsensoren

Für die Erstellung einer Karte wird ein bildgebender Sensor benötigt, der die Umgebung vermisst. Zur Vermessung eines Gebietes muss dieser Sensor über das Gebiet bewegt werden. Dafür wird der Sensor an einem UA angebracht. Damit die Messungen korrekt in eine Karte überführt werden können, müssen sie mit der Bewegung des UA in Verbindung gebracht werden. Dafür wird eine NAV-Sensorik eingesetzt, die die Bewegung des UA kontinuierlich aufzeichnet. Aus konstruktionstechnischen Gründen befindet sich die NAV-Sensorik an einer anderen Position des UA und in unterschiedlicher Ausrichtung als der bildgebende Sensor. Dieser Unterschied in Position und Lage wird als Montagepose bezeichnet.

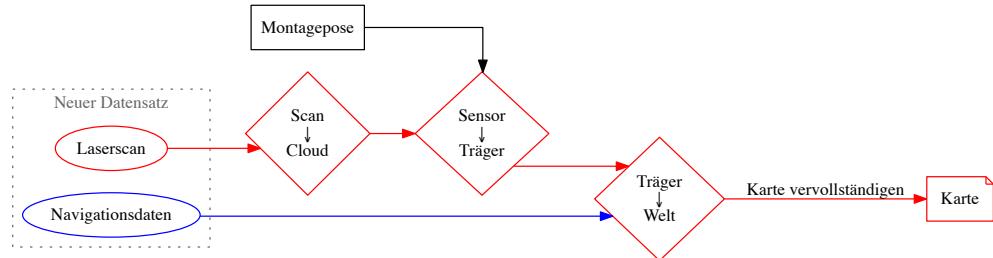
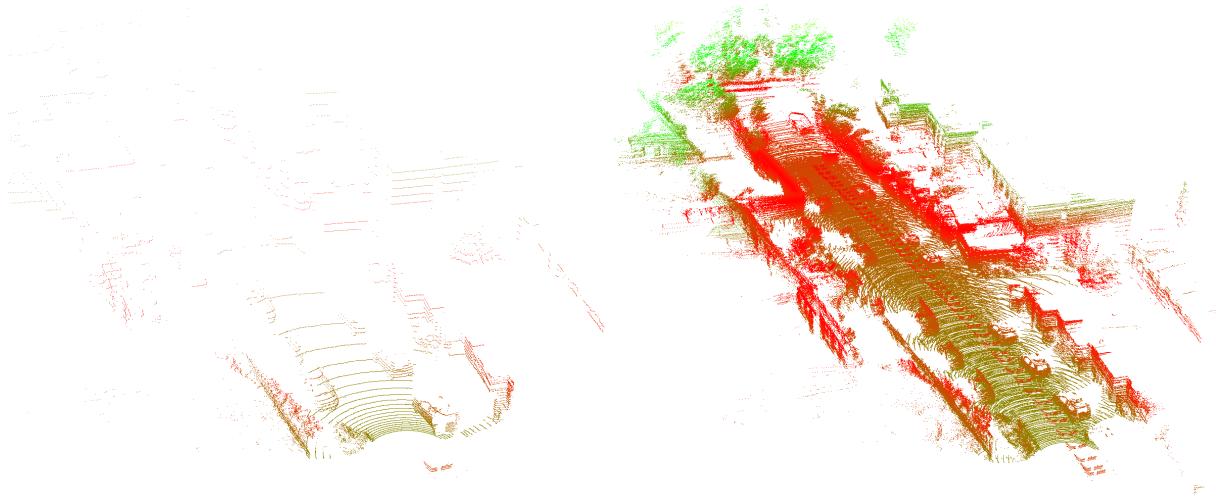


Abbildung 3.1.: Ablauf der Kartografie von Punktwolken

Die aufgenommenen Laserscans der Umgebung werden nun gemäß Abbildung 3.1 mit der Montagepose vom Sensorbezugssystem des bildgebenden Sensors in das Trägerbezugssystem transformiert. Nun können die Laserscans durch Transformation mit den Navigationsdaten der NAV-Sensorik vom Trägerbezugssystem in das Weltbezugssystem überführt werden.

Durch kontinuierliche Speicherung der aufgenommenen Messungen der Umgebung, die sich nun im Weltbezugssystem befinden, kann eine Karte erstellt werden. Abbildung 3.2 zeigt eine Gegenüberstellung von einem einzelnen Laserscan 3.2a und seiner Überführung in eine Karte 3.2b.



(a) Einzerner Laserscan

(b) Laserscan zu einer Karte hinzugefügt

Abbildung 3.2.: Gegenüberstellung von einem einzelnen Laserscan und seiner Überführung in eine Karte

Eine fehlerhafte oder ungenaue Angabe der Montagepose führt dazu, dass die Karte falsch aufgebaut wird. Aus diesem Grund wird in den folgenden Abschnitten eine automatisierte relative extrinsische Kalibrierung vorgestellt, die eine korrekte Bestimmung der Montagepose ermöglicht.

### 3.1. Forschungsstand

Zu diesem Zeitpunkt ist nur die Arbeit von Levinson (2011) über eine automatisierte extrinsische Kalibrierung von LiDAR-Sensoren bekannt. Er beschreibt eine intrinsische sowie eine extrinsische Kalibrierung von LiDAR-Sensoren für ein autonomes Fahrzeug, wobei die präsentierte Lösung auf der Auswertung der einzelnen Laserstrahlen basiert.

Er setzt voraus, dass die Echos benachbarter Laserstrahlen auf das gleiche Hindernis in der realen Welt treffen. Bei höheren Fahrzeuggeschwindigkeiten und dynamischeren Bewegungen scheitert das vorgestellte Verfahren. Des Weiteren ist diese Art der extrinisischen Kalibrierung stark auf den verwendeten LiDAR-Sensor zugeschnitten und nicht allgemein anwendbar.

Bisher wurde die extrinsische Kalibrierung von LiDAR-Sensoren zu IMUs anhand fest definierter *Features*<sup>17</sup> unter anderem durch Talaya et al. (2004) vorgenommen. Dabei wurden Häuserwände vermessen, deren NAVSTAR-GPS-Positionen bestimmt und mit reflektierenden Materialien beklebt. Die reflektierenden Materialien konnten im Anschluss auf Grund einer höheren Intensität beim Laserscan extrahiert werden. Durch die Analyse der bekannten NAVSTAR-GPS-Positionen und den gemessenen Positionen im Laserscan mit höheren Intensitäten konnten Rückschlüsse auf die Montagepose des LiDAR-Sensors gezogen werden. Das Verfahren basiert auf der Erkennung von planaren Features. Eine mathematische Lösung für dieses Verfahren zeigten Bae/Lichti (2007), die die Abstände zwischen Scanpunkten zu einer möglichen zugehörigen Fläche minimierten.

Für den Fall, dass keine planaren Flächen zur Verfügung stehen, präsentierte Chan et al. (2015) einen Ansatz, der zylinderförmige Features aus Laserscans zur Kalibrierung verwendet. Dabei wurde die Annahme getroffen, dass in urbanem Gelände häufig Zylinder in Form von Rohren auftauchen, deren Ausrichtung meist horizontal oder vertikal ist. Zur Validierung ihres Ansatzes untersuchten sie verschiedene Ausrichtungen der Zylinder und unterschiedliche Montagepositionen des LiDAR-Sensors. Ihr Ansatz scheiterte, wenn sich mehrere unterschiedlich ausgerichtete Zylinder in der zu detektierenden Umgebung befanden.

Natürliche Umgebungen weisen kaum planare oder zylinderförmige Formen auf, sodass diese Ansätze nur in urbanem Gelände Anwendung finden können. Sheehan et al. (2011) und Maddern et al. (2012) identifizierten Scannerparameter anhand eines Maßes der

---

<sup>17</sup> Features bezeichnet eindeutig erkennbare und unterscheidbare Merkmale in aufgenommenen Bildern.

Vielfältigkeit von Punktwolken (Rényi Quadratic Entropy). Ihr Hauptaugenmerk lag dabei auf der Synchronisation mehrerer LiDAR-Sensor zueinander um detaillierte Umgebungs-scans zu generieren. Des Weiteren befinden sich die Scanner zwar in einer rotatorischen Bewegung, jedoch translatorisch in einem stationären Zustand.

Des Weiteren ist in der Literatur der Simultaneous Localization and Mapping (SLAM)-Algorithmus bekannt. SLAM bezeichnet die Korrektur von Sensordaten anhand einer kontinuierlichen Kartenerstellung.<sup>18</sup>

Davison et al. (2007) präsentierten einen SLAM Ansatz, um mehrere Kamerabilder bei schnellen Bewegungen zu einer spärlichen Landkarte zusammenzufügen. Sie verwendeten dafür ein probabilistisches Feature-basiertes Kartenmodell und ein allgemeines Bewegungs-modell einer Kamera. Informationen über die Entfernung wurden durch die Features definiert. Einen Ansatz zur Generierung von 3D-Landkarten mit Entfernungsinformationen bietet May et al. (2008). Der Ansatz verwendete ToF-Kameras und einen Roboter, der für jede Aufnahme stoppt, wodurch stabile Punktwolken generiert werden können.

Bisherige Entwicklungen am DLR beschreiben gemäß Stellmacher (2014) die Regis-trierung von Punktwolken auf Basis eines OcTrees (eine effiziente Baumstruktur, um dreidimensionale Datensätze hierarchisch zu gliedern). Das Ziel seiner Arbeit war eine Optimierung der Aneinanderreihung der Punktwolken in Echtzeit<sup>19</sup>. Das Resultat ist eine minimale Verbesserung der Registrierung, jedoch konnte die Echtzeitbedingung nur unter Einschränkungen eingehalten werden.

Des Weiteren werden bereits gemäß Krause/Evert (2012) erfolgreich Feature-basierte Kalibrierungen zwischen LiDAR-Sensoren und Kameras eingesetzt. Grundlage für diese Forschungen ist unter anderem die Arbeit von Andert (2011).

---

<sup>18</sup> Im Automobilbereich werden zur Verfolgung der Bewegung sogenannte Odometriesensoren verwendet, die zum Beispiel die Umdrehungen der Räder zählen. Durch Schlupf auf Grund unterschiedlicher Luftdrücke der Räder, wechselndem Beladezustand oder variierendem Untergrund, sind diese Messdaten sehr fehlerbehaftet. Durch die Erstellung einer Karte und einem *Loop-Closing-Pfad* (es wird eine zyklische Strecke abgefahren) werden die Sensorwerte entsprechend korrigiert.

<sup>19</sup> Echtzeit bezeichnet in diesem Zusammenhang die Bedingung, dass die Punktwolken schneller zueinander registriert werden können, als neue Sensordaten vorliegen.

### 3.2. Ansatz zur Kalibrierung

Für die relative extrinsische Kalibrierung des bildgebenden Sensors zur NAV-Sensorik werden folgende Anforderungen definiert: (1) die Umgebung wird als unveränderlich und starr angenommen, (2) die zu vermessende Bewegung muss größer als die größte Messungenaugkeit des Systems sein, (3) die Position der Sensoren zueinander ist während der Kalibrierung unveränderlich und (4) der Träger ist fest mit dem UA verbunden, sodass die Bewegung des UA direkt der Bewegung des Trägers entspricht.

Die Grundlage der Kalibrierung bilden die Aufnahme von Umgebungsscans durch bildgebende Sensoren und die gleichzeitige Verfolgung der Bewegung durch eine NAV-Sensorik, bestehend aus einer 6DoF-IMU und einem GPS.

Die Erstellung der Umgebungsscans erfolgt durch scannende LiDAR-Sensoren, die für jeden Scan eine 3-dimensionale Punktwolke in Form von Entfernungsmessungen generieren. Die Punktwolken werden dabei in einer festen Taktrate, die abhängig vom verwendeten LiDAR-Sensor ist, generiert. Aus aufeinanderfolgenden Umgebungsscans wird die Differenz anhand eines ICP-Algorithmus berechnet und in Form einer Transformationsmatrix gespeichert. Parallel werden die Bewegungen des UA in Form von Positions- und Winkeländerungen gemessen. Die Taktrate für die Bewegungsmessung ist systembedingt höher als die der Erstellung der Umgebungsscans. Diese Bewegungen werden ebenfalls als Transformationsmatrix gespeichert.

Bei einem LiDAR-Sensor handelt es sich um ein scannendes System. Dabei wird die Umgebung in zeitlich diskreten Schritten vermessen. Befindet sich der LiDAR im Stillstand, kann die Umgebung korrekt vermessen werden. Wird der LiDAR jedoch während der Aufnahme einer Messung bewegt, führt dies zu Verzerrungen.

Abbildung 3.3 zeigt die Auswirkung, wenn sich der Träger, in diesem Fall das UA, in Bewegung befindet. Ist die Aufnahmefrequenz des LiDAR zu gering, hat sich der Träger deutlich bewegt. Die soeben erstellte Punktwolke repräsentiert nun eine Umgebung, die auf Grund der Bewegung verzerrt wird. Um dies zu korrigieren, muss der Laserscan Strahl für Strahl an die zwischenzeitliche Position des Trägers angepasst werden. Diese Anpassung

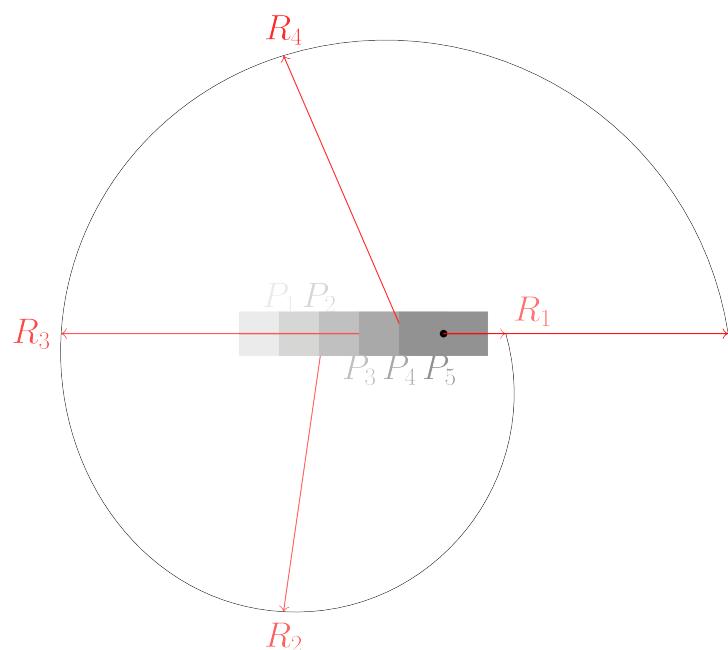


Abbildung 3.3.: Veränderung des Bildbereichs, wenn die Bewegung des Trägers (von  $P_1$  nach  $P_5$ ) schneller als die Aufnahmefrequenz des Sensors ist (schematische Darstellung)

wird im Folgenden als Bewegungskorrektur bezeichnet und erfolgt durch Interpolation der NAV-Daten. Während der Recherche wurden diesbezüglich zwei Strategien zur Kalibrierung erarbeitet.

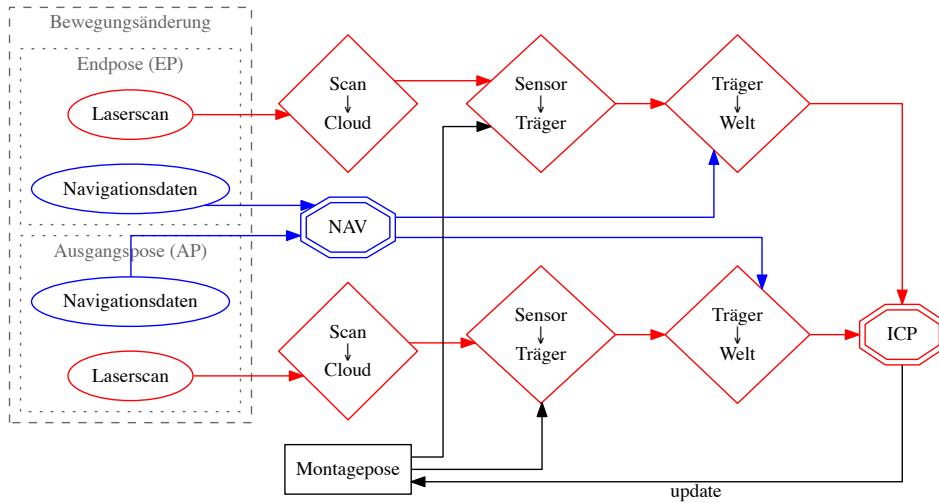


Abbildung 3.4.: Kalibrierung im Weltkoordinatensystem

Abbildung 3.4 zeigt das Ablaufdiagramm der relativen extrinsischen Kalibrierung ohne Bewegungskorrektur. Dabei werden alle Bewegungen des Trägers während der Generierung der Datenpakete des LiDAR ignoriert. Dies führt dazu, dass die gemessenen Hits (ein Hit bezeichnet den Auftreffpunkt des Laserstrahls auf einem Objekt) je nach Geschwindigkeit des Trägers zu weit in der Hauptbewegungsrichtung verortet werden.

Das Diagramm zeigt eine Ausgangs- und eine Endpose. Dazwischen wurde eine Bewegungsänderung durchgeführt, die mittels der NAV-Sensorik gemessen wurde. Die empfangenen Umgebungsscans befinden sich im Bezugssystem des Sensors und werden nach der Umwandlung von Kugelkoordinaten zu kartesischen Koordinaten in einer Punktwolkenstruktur gespeichert.

Durch Transformation mit der Montagepose (zu Beginn kann diese Pose durch den Be-

nutzer gesetzt werden, andernfalls sind alle Komponenten „0“) werden beide Punktwellen in das Trägerbezugssystem transformiert. Die Punktwellen, die zum Zeitpunkt der Endpose aufgenommen wurde, entspricht zeitgleich dem relativen Weltbezugssystem, da danach noch keine Bewegung stattgefunden hat. Die Punktwellen zur Ausgangspose wird hingegen mit der durch die NAV-Sensorik gemessenen Bewegung in das relative Weltbezugssystem transformiert. Anschließend werden die beiden Punktwellen, die sich nun im relativen Weltbezugssystem befinden, dem ICP-Algorithmus übergeben. Der ICP-Algorithmus bestimmt, wie gut die zwei Punktwellen zueinander passen. Wurde die richtige Montagepose zwischen dem LiDAR und der NAV-Sensorik verwendet, sollten die Punktwellen perfekt übereinstimmen. Liefert der ICP-Algorithmus hingegen eine Differenz zwischen den Punktwellen, so ist die ursprünglich übergebene Montagepose um diese Differenz falsch und muss angepasst werden.

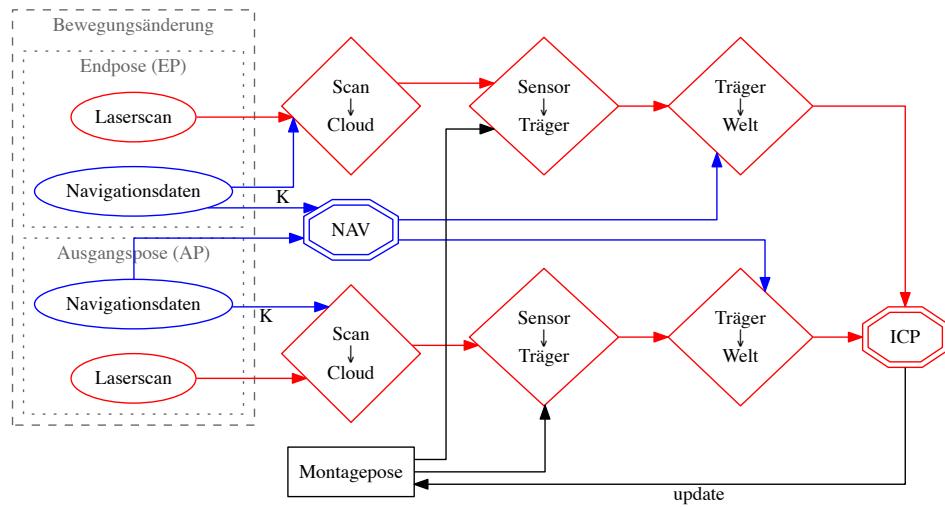


Abbildung 3.5.: Kalibrierung im Weltkoordinatensystem mit Bewegungskorrektur

Abbildung 3.5 zeigt den Ablauf der Kalibrierung mit Bewegungskorrektur. Der Unterschied zum vorherigen Diagramm ist die zusätzliche Transformation der einzelnen

Umgebungsscans unter Verwendung der NAV-Daten. da der LiDAR-Sensor jedoch eine andere Montagepose als die NAV-Sensorik aufweist, ist die gemessene Bewegung der NAV-Sensorik zur tatsächlichen Bewegung des LiDAR-Sensors unterschiedlich.

Eine Möglichkeit zur Korrektur wäre die Anpassung der NAV-Daten auf die initial vorgegebene Montagepose, die jedoch ebenfalls fehlerbehaftet ist. Im Allgemeinen weist diese Methode einen zusätzlichen Berechnungsschritt für jede Punktwolke durch die Transformation auf. In der aktuellen Implementierung wird die Bewegung des Trägers berücksichtigt.

### 3.3. Bewegungsextraktion aus aufeinanderfolgenden Punktwolken

Eine Grundlage für die Kalibrierung ist die Schätzung der Bewegung zwischen zwei Punktwolken.

#### 3.3.1. Registrierung von Punktwolken

Die Registrierung von Punktwolken erfolgt durch die Implementierung des ICP-Algorithmus der PCL gemäß Besl/McKay (1992). Dabei ist das Modell, das die Punktwolke repräsentiert, unabhängig für die Registrierung. Das Ziel des ICP ist die Transformation zwischen zwei vorliegenden Punktwolken anhand der Fehlermetrik gemäß (3.1).

$$d(\vec{p}, X) = \min_{\vec{x} \in X} ||\vec{x} - \vec{p}|| \quad (3.1)$$

Dabei bezeichnet  $d$  die Distanz-Metrik zwischen einem beliebigen Punkt  $\vec{p}$  der Datenmenge  $P$  und der anzulegenden Datenmenge  $X$ . Der Punkt in  $X$ , der die kleinste Distanz zu  $\vec{p}$  hat, wird als  $\vec{y}$  bezeichnet.  $Y$  bezeichnet anschließend das Set aus nächsten Nachbarn durch den Operator  $C$  und wird gemäß (3.2) berechnet.

$$Y = C(P, X) \quad (3.2)$$

Der Algorithmus unterteilt sich dabei in 3 Schritte:

- 
1. Berechne die nächsten Nachbarn für  $Y_k = C(P_k, X)$
  2. Berechne die Transformation zwischen  $P_0$  und  $Y_k$
  3. Wende die Transformation auf  $P_0$  an

Diese Schritte werden solange wiederholt, bis der Algorithmus terminiert. Dies ist der Fall, wenn (a) die Anzahl der Iterationen ein vom Benutzer angegebenes Maximum erreicht, (b) die Differenz zwischen der aktuellen geschätzten und der vorherigen Transformation  $\epsilon$  kleiner als die vom Benutzer angegebenen Grenze ist (die Transformation wurde nur minimal besser) oder (c) die Summe des euklidischen quadratischen Fehlers kleiner als der vom Benutzer angegebene Fehlerbereich ist (die Transformation ist ausreichend genau).

Zur Beschleunigung der Berechnung können die Punktwolken nach unterschiedlichen Aspekten gefiltert und Features extrahiert werden.

### 3.3.2. Extraktion von Features

Für die Registrierung von Punktwolken müssen Punkte miteinander verglichen werden. Punkte werden in einer Punktwolke in Form von kartesischen Koordinaten gespeichert. Vergleicht man nur die kartesischen Koordinaten zweier Punktwolken miteinander, besteht die Gefahr, dass zwei Punkte, die zu unterschiedlichen Zeiten aufgenommen wurden, die gleichen Koordinaten vorweisen, jedoch von unterschiedlichen Oberflächen stammen.

Rusu/Cousins (2011) führt das Konzept der *point feature representations* (auch als *local descriptors*, *shape descriptors* und *geometric features* in der Literatur bekannt) ein.

Durch Zusammenfassung mehrerer Punkte in einer Nachbarschaft kann die Geometrie der Oberfläche erfasst und als *feature* formuliert werden. Im Idealfall sind die so definierten *features* zu einem gewissen Grad ähnlich zu anderen Punkten auf der gleichen oder einer ähnlichen Oberfläche.

Eine gute Repräsentation von *features* zeichnet sich dadurch aus, dass es möglich ist, sie unter Einwirkung einer Transformation in Form von Rotationen und Translationen, trotz unterschiedlich stark aufgelöster Punktwolken und der Existenz von Messfehlern identifizieren zu können.

Durch die Bestimmung von Keypoints kann der Berechnungsaufwand von Features im Vorhinein minimiert werden, da der Suchraum eingeschränkt wird. Ebenfalls wird damit verhindert, dass beim späteren Matching Fehler durch Nicht-Schlüsselpunkte verursacht werden.

### 3.3.2.1. Bestimmung von Keypoints

Punkte, die in irgendeiner Weise als relevant identifiziert werden, bezeichnet man als Keypoints (Schlüsselpunkte). Sie besitzen dabei folgende Eigenschaften:

- *Sparseness* (nur eine kleine Auswahl an Punkten aus der Gesamtmenge kommt in Frage),
- *Repeatability* (die identifizierten Punkte sind auch in ähnlichen Punktwolken an der entsprechenden Position zu finden),
- *Distinctiveness* (die Umgebung rund um den Schlüsselpunkt weist eine eindeutige Form oder Erscheinung auf).

Zur Identifikation von Keypoints werden die Punkte einer Punktfolge gemäß ihrer nächsten Nachbarn sortiert. Dafür werden in der PCL die kd-Suchbäume gemäß Abschnitt 2.3.3 verwendet. Im Anschluss besteht die Möglichkeit, die Punktmenge nach einer festen Anzahl an Nachbarn für einen Punkt zu durchsuchen oder alle benachbarten Punkte in einem Radius  $r$  zu betrachten.

### 3.3.2.2. Skaleninvariante Merkmalstransformation (SIFT)

Scale-invariant feature transform (SIFT) bezeichnet gemäß Lowe (1999) einen Algorithmus zur Detektion und Beschreibung lokaler Merkmale, die in gewissen Grenzen invariant gegenüber Koordinatentransformationen wie Translation, Rotation und Skalierung sind.

Der Algorithmus ist dabei in fünf Aufgaben unterteilt: (1) Feature-Detektion, (2) Feature-Matching und Indizierung, (3) Identifizierung von Clustern anhand eines *Hough transform*-Votings, (4) Überprüfung des Modells durch die Methode der kleinsten Quadrate und (5) Entfernung von Ausreißern.

Diese Variante ist die Grundlage für viele Optimierungen und wird bereits erfolgreich im Bereich der Bildverarbeitung eingesetzt.

### 3.3.2.3. (Fast) Point-Feature-Histogramm

Das Point-Feature-Histogramm ist eine Repräsentation der geometrischen Eigenschaften der benachbarten Punkte an einem Punkt. Dafür wird die mittlere Krümmung in Form eines mehrdimensionalen Histogramms von Werten generalisiert. Die Repräsentation eignet sich gut für unterschiedlich dichte Punktwolken und ist invariant gegenüber Änderungen des Betrachtungswinkels und Messfehlern.

Dafür werden die Oberflächen-Normalen der Umgebung geschätzt und die Variationen aller Interaktionen zwischen den Normalen extrahiert. Dieses Verfahren ist demzufolge stark von der Qualität der Normalen-Schätzung an jedem Punkt abhängig.

Die Filterfunktion ist sehr zeit- und rechenaufwendig, weswegen im Allgemeinen die *Fast*-Variante verwendet wird. Sie unterscheidet sich hauptsächlich darin, dass nicht alle Punkte miteinander verknüpft werden, sondern dass eine Gewichtung der Interaktionen durchgeführt wird.

### 3.3.2.4. Schätzung von Oberflächen-Normalen

Oberflächen-Normale sind eine wichtige Eigenschaft von geometrischen Oberflächen. Sie ermöglichen die Kombination von erkannten Oberflächen zu geometrischen Körpern, die speziell für die Erkennung von Hindernissen von Vorteil sind.

Für eine vorliegende Punktfolge können die Oberflächen-Normalen auf zwei Arten bestimmt werden: (a) durch die Erstellung von Meshs und anschließender Berechnung oder (b) durch Annäherung der Normalen einer Ebene an eine Oberfläche durch Minimierung der kleinsten Quadrate.

### 3.3.3. Filterung der Daten

Filter dienen der Extraktion oder Anpassung von Daten. Für die vorliegende Arbeit werden verschiedene Filterfunktionen zum Anpassen des Messbereichs, der Minimierung der Punktmenge und der Korrektur von Messfehlern in Punktwolken vorgestellt.

#### 3.3.3.1. Begrenzung des Messbereichs

Jeder LiDAR-Sensor ist für einen bestimmten Messbereich ausgelegt. Messungen außerhalb dieses Messbereiches sind meist ungenau oder undefiniert. Messwerte, die sich unterhalb einer Mindest- und oberhalb einer Maximalentfernung befinden, können durch einen *Pass-Through*- oder auch Bandpass-Filter entfernt werden. Diese Filterfunktion entfernt Punkte, bei der es sich mit sehr hoher Wahrscheinlichkeit um fehlerbehaftete Messpunkte handelt.

Des Weiteren kann durch einen *Pass-Through*-Filter das Field-of-View (FoV) verändert werden, wenn zum Beispiel ein bestimmter Bereich durch Elemente des Trägers verdeckt werden. Dies ist zum Beispiel der Fall, wenn der LiDAR-Sensor an der Unterseite zwischen dem Fahrgestell montiert ist. Das Fahrgestell reflektiert die Laserstrahlung, sodass es vom LiDAR-Sensor als Hindernis detektiert wird. Da sich das Fahrgestell jedoch dauerhaft im FoV befindet, würde es bei der Bewegungsextraktion eine statische Umgebung implizieren.

Die Begrenzung des Messbereichs kann nur an einer Punktfolge erfolgen, die noch nicht in das Weltbezugssystem überführt wurde, da ein unmittelbarer Zusammenhang zwischen Messpunkten und LiDAR benötigt wird.

#### 3.3.3.2. Entfernen von Messfehlern

Nach der Begrenzung der Daten auf den gültigen Messbereich können durch stochastische Annahmen Punktfolgen von Messfehlern bereinigt werden. Es ist nahezu unmöglich, Messfehler in einer Punktfolge zu identifizieren. Aus diesem Grund bedient man sich der Annahme, dass Ausreißer nahe einer Ansammlung von mehreren Punkten möglicherweise Messfehler repräsentieren. Dies bezeichnet man als *statistical outlier*. Die Funktion filtert gemäß der Angabe der mittleren Punktdichte die gesamte Punktfolge und entfernt

entsprechende Punkte, die sich außerhalb dieser Ansammlungen befinden. Die Gefahr bei diesem Filter besteht darin, dass auch Punkte entfernt werden können, bei denen es sich nicht um Messfehler handelt.

### 3.3.3.3. DownSampling der Punktwolken

Scannende LiDAR-Sensoren haben die Eigenschaft, dass die Winkelauflösung mit zunehmender Entfernung der zu messenden Hindernisse abnimmt. Das bedeutet, dass Hindernisse, die sich nah am LiDAR-Sensor (innerhalb des Messbereiches) befinden, mit einer höheren Zahl an Hits vermessen werden. Die Struktur der Oberfläche ist somit höher aufgelöst und kleinere Unterschiede sind messbar. Befindet sich ein Hindernis jedoch weiter vom LiDAR-Sensor entfernt, so werden die Hindernisse deutlich größer aufgelöst als nähere.

Das heißt, dass der Nahbereich des LiDAR-Sensors eine höhere Punktdichte aufweist. Kleine Rotationen zwischen den Punktewolken sind durch Veränderungen in der Ferne erkennbar, die Menge an Punkten im Nahbereich überwiegt jedoch, sodass diese kleinen Unterschiede für die Schätzung der Transformation nicht berücksichtigt werden.

Durch sogenanntes DownSampling werden Punkte aus der Punktewolke entfernt, um den Einfluss bestimmter Regionen auf die Registrierung zu minimieren. Im Folgenden werden zwei Varianten zum DownSampling der Punktewolken vorgestellt.

#### Voxel-Grids

Die erste Variante erzeugt ein gleichmäßiges DownSampling, das heißt, die Punktewolke wird in allen Regionen gleichmäßig ausgedünnt. Dabei werden die Punkte in einem definierten Bereich in ihrem Schwerpunkt, dem Voxel, zusammengefasst.

Das resultierende Voxel-Grid ist gleichverteilt und erfüllt die Bedingung, dass alle Regionen die gleiche Relevanz für die Bestimmung der Transformation haben. Wichtig ist, dass die Größe des Voxel-Grids auf die Verteilung der vorliegenden Daten angepasst wird, da sonst entweder zu viele oder zu wenige Punkte zusammengefasst werden.

Dieses Verfahren hat den Nachteil, dass auf Grund der Einteilung in Voxel die wahren

Positionen der Punkte manipuliert werden. Die resultierende, geschätzte Transformation entspricht somit unter Umständen nicht mehr der eigentlich vorherrschenden Transformation.

### Kovarianz-Downsampling

Nach Gelfand et al. (2003) wirkt sich eine fehlerhafte Kalibrierung der Sensorpose negativ auf die Registrierung der Punktwolken mit dem ICP-Algorithmus aus. Dazu analysierten sie die Daten im Vorhinein und berechnen eine geometrische Stabilität. Punktwolken sind geometrisch instabil, wenn sie keine eindeutige Transformation ermöglichen. In diesem Fall divergiert der ICP oder fällt mit hoher Wahrscheinlichkeit in ein lokales Minimum.

Das Resultat ist eine falsche Transformation, was zum Beispiel bei planaren Flächen der Fall ist. Bei der Registrierung existieren mindestens zwei Freiheitsgrade, anhand derer die Flächen zueinander verschoben oder rotiert werden können. Dieses Problem tritt somit bei geometrischen Grundformen wie Quadern, Zylindern und Kugeln auf. Das Maß dieser geometrischen Stabilität ist die Konditionszahl der Kovarianzmatrix<sup>20</sup>.

Die Kovarianz bezeichnet in diesem Zusammenhang die Varianz der Oberflächen-Normalen. Dazu wird die Punktwolke gemäß Abschnitt 3.3.2.4 analysiert. Die Kovarianzen der verschiedenen Dimensionen werden in einer Kovarianzmatrix abgelegt. Aus den Eigenvektoren der Kovarianzmatrix berechnen sie eine Konditionszahl. Im Anschluss sortieren sie die Punkte absteigend nach ihrer Unabhängigkeit von der Umgebung und entfernen die voneinander abhängigsten 70% aus dem Datensatz. In ihren Experimenten konvergiert der ICP-Algorithmus nach 25 Iterationen. Dieses Verfahren wird als Kovarianz-Downsampling bezeichnet. Abbildung 3.6 zeigt einen Laserscan vor und nach dem Kovarianz-Sampling.

---

<sup>20</sup> Eine Kovarianzmatrix enthält Informationen über die Streuung eines Zufallsvektors und über die Korrelationen zwischen dessen Komponenten

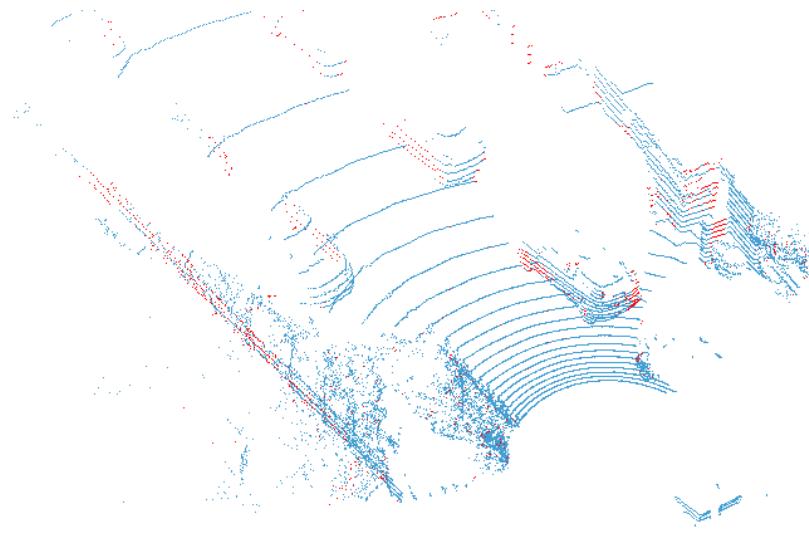


Abbildung 3.6.: Angewandtes Kovarianz-Downsampling (rot) und die originale Punkt Wolke (blau)

### 3.3.3.4. Vorgabe einer initialen Ausrichtung

Die Vorgabe einer initialen Ausrichtung bezeichnet die initiale Transformation, mit der die Eingabe-Punkt Wolke transformiert wird. Sie kann entweder vom Benutzer vorgegeben oder durch einen *Sample-Consensus*-Algorithmus unter Verwendung des Fast Point-Feature-Histogram (FPFH) errechnet werden. Eine initiale Ausrichtung vermindert das Risiko, dass der ICP zu einem lokalen Minimum konvergiert.

Die Implementierung des *Sample-Consensus* nach Rusu et al. (2009) ermöglicht das schnelle Sortieren nach Relevanz von einer Vielzahl an gesampelten Punkten unter der Wahrung der geometrischen Beziehungen zwischen den Punkten.

### 3.3.4. Zusammenfassung

Die Registrierung von Punkt Wolken mit dem ICP-Algorithmus ist hauptsächlich für die feiner Anpassung einer initialen Transformation gedacht und konvergiert schnell in lokale Minima, wenn diese Transformation stark vom Ziel abweicht.

Punkt Wolken können anhand verschiedener Filterfunktionen optimiert und angeglichen

werden. Durch die Berechnung von verschiedenen Features kann zudem die Registrierung zweier Punktwolken beschleunigt und das Risiko von lokalen Minima reduziert werden.

Trotz aller Bemühungen liefert die Standard-Implementierung des ICP-Algorithmus keine Garantie für das Auffinden eines globalen Minimums. In der Literatur existieren bereits verschiedene alternative Ansätze, die durch zusätzliche Bedingungen den ICP eingrenzen und versuchen, entsprechende lokale Minima zu erkennen. Beispielhaft seien dafür Yang et al. (2013), Bellekens et al. (2014), Belshaw/Greenspan (2009), Boughorbel et al. (2010), Droschel/Behnke (2011), Gelfand et al. (2003), Hervier et al. (2012) und Rusinkiewicz/Levoy (2001) erwähnt.

Für diese Arbeit wird die Registrierung hingegen durch eine äußerlich geschachtelte Nichtlineare-Optimierung variiert. Die Punktwolken werden nicht durch den ICP registriert, sondern seine Fehlermetrik als Variationseingabe für eine Levenberg-Marquardt Optimierung verwendet.

## 3.4. Nichtlineare Optimierung nach Levenberg-Marquardt

Die Maximierung und Minimierung eines Problems ist interdisziplinär. Beispielsweise soll der Profit einer Firma maximiert, aber die Kosten und eventuelle Risiken sollen minimiert werden. Nichtlineare Optimierung betrifft Probleme, wenn die zu minimierenden Faktoren in einer nicht-linearen Funktion zueinander stehen.

Prinzipiell unterscheidet man zwischen *one-variable-* und *n-variable-optimisation*. Eine *one-variable-optimisation* kommt zum Einsatz, wenn die zu optimierende Funktion nur von einer Variablen abhängig ist. Zur Lösung gibt es zwei Methoden, wobei die erste direkt den Funktionswert (*direct-method*) verwendet und die zweite auf Ableitungen (*gradient-method*) basiert.

### 3.4.1. One-Variable-Optimisation

Ein Optimum nach der *gradient-method* ist gegeben, wenn (3.3) und (3.4) zutreffen.

$$f'(x) = 0 \quad (3.3)$$

$$f''(x) \begin{cases} < 0 & \text{wenn Maximum gesucht} \\ > 0 & \text{wenn Minimum gesucht} \end{cases} \quad (3.4)$$

Allgemein verwendet man die *secant-* oder die *Newton-Ralphson's method*.

### 3.4.1.1. Secant-Method

Die *secant-method* erfolgt entweder durch die Ableitung der Geradenfunktion oder durch die Taylor Reihe. Es handelt sich dabei um einen iterativen Prozess gemäß (3.5), bei dem zwei Anfangsbedingungen ( $x_0$  und  $x_1$ ) ausgewählt werden und anschließend der Nachfolger  $x_{i+1}$  berechnet wird. Dies wird so lange wiederholt, bis  $|f'(x_i)| = 0$  oder ausreichend klein ist.

$$x_{i+1} = x_i - \frac{f'(x_i)}{f(x_i) - f(x_{i+1})} \times (x_{i+1} - x_i) \quad (3.5)$$

Vorteile der *secant-method* sind, dass nur die erste Ableitung  $f'(x_i)$  benötigt wird und die Rechnung immer zu einer Lösung des Problems führt. Nachteilig ist hingegen, dass die Funktion nicht stets zum Minimum konvergiert und keine Fehlerschranken für die berechnete Iteration bestimmt werden. Das bedeutet, dass keine Aussage gemacht werden kann, wie nah man bereits am Minimum ist.

### 3.4.1.2. Newton-Ralphson's Method

Eine weitere Methode zur Erfüllung der ersten Bedingung ist durch die *Newton-Ralphson's method* gegeben. Diese Methode ist ebenfalls von der Taylor-Reihe abgeleitet und wird gemäß (3.6) iteriert.

$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)} \quad (3.6)$$

Die Abbruchbedingung nach der Angabe eines Startpunkts  $x_0$  ist, wie bei der *secant-method*, wenn  $|f'(x_i) = 0|$  oder ausreichend klein ist. Dabei werden beide Ableitungen der Funktion benötigt, wobei sich die Lösungsfunktion tangential immer weiter an das Minimum annähert. Die Vorteile bestehen darin, dass sie die wenigsten Iterationen benötigt, um das Minimum zu finden und dabei quadratisch konvergiert. Die Nachteile sind, dass die Methode fehlschlägt, wenn die zweite Ableitung  $|f'' = 0|$  ist (Division mit 0). Des Weiteren muss sich der initiale Punkt bereits nah am Minimum befinden.

### 3.4.2. N-Variable-Optimisation

Wird von mehr als nur einer Variablen einer Funktion das Optimum gesucht, spricht man von *n-variable-optimistation*. Die Schreibweise dieses Problems erfolgt dabei gemäß (3.7).

$$f(x) = f(x_1, x_2, x_3, \dots, x_n) \quad (3.7)$$

Die erste Ableitung einer Funktion mehrerer Unbekannter  $f'$  wird als Matrix der partiellen Ableitungen gemäß (3.8) formuliert.

$$f' = \begin{pmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \frac{\partial f}{\partial x_3} & \dots & \frac{\partial f}{\partial x_n} \end{pmatrix}^T \quad (3.8)$$

Die zweite Ableitung resultiert in der Hesse-Matrix gemäß (3.9). Die Hesse-Matrix entspricht der Transponierten der Jacobi-Matrix des Gradienten, wobei  $\frac{\partial^2 f}{\partial x_i \partial x_j}$  die zweiten partiellen Ableitungen bezeichnet.

$$f'' = H_f(x) := \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2^2}(x) & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \frac{\partial^2 f}{\partial x_n \partial x_2}(x) & \dots & \frac{\partial^2 f}{\partial x_n^2}(x) \end{pmatrix} \quad (3.9)$$

Damit ein Optimum gefunden werden kann, muss die erste Ableitung  $|f'| = 0$  sein. Die zweite Bedingung ist, dass die Eigenwerte der Hesse-Matrix positiv sein müssen.

Zur Lösung des Problems werden sogenannte *line-search*-Algorithmen verwendet. Ein *line-search*-Algorithmus ist ein iterativer Prozess, der eine Suchrichtung  $p_k$  und eine Schrittweite  $s$  benötigt. Jede Iteration erfolgt dabei gemäß (3.10), wobei man zwischen *weak-* und *perfect-line-search* unterscheidet. Der Unterschied besteht darin, dass im Gegensatz zur *weak-line-search* bei einer *perfect-line-search* die Schrittweite  $s$  durch Berechnung angepasst wird.

$$x_{k+1} = x_k + s * p_k \quad (3.10)$$

Für eine erfolgreiche *line-search* werden die drei Wolfe's Bedingungen definiert. Die erste, bekannt als *sufficient-decrease*, stellt gemäß (3.11) sicher, dass die Schrittweite nicht zu groß gewählt wird.

$$f(x_k + \alpha * p_k) < f(x_k) + c_1 * \alpha * \nabla^2 * f(x_k) * p_k \quad (3.11)$$

Die zweite Bedingung, *curvature-condition*, stellt gemäß (3.12) sicher, dass die Schrittweite ausreichend groß ist, um sich einer Lösung anzunähern.

$$f(x_k + \alpha * p_k)^T \geq c_2 * \nabla^2 * f(x_k)^T * p_k \quad (3.12)$$

Die letzte Bedingung wird als *strong Wolfe's condition* bezeichnet und stellt gemäß (3.13) sicher, dass sich die Suchrichtung  $p_k$  in Richtung eines Minimums befindet.

$$f(x_k + \alpha * p_k)^T \leq c_3 * \nabla^2 * f(x_k)^T * p_k \quad (3.13)$$

Im Folgenden werden die Suchstrategien *steepest-descent*, *Newton*s- und *quasi-Newton*s-*method* erläutert.

### 3.4.2.1. Steepest-Descent

Die erste Suchstrategie beginnt an einer Startposition  $x_0$ . Für jede Iteration  $k = 0, 1, 2, \dots, n$  wird der Gradient  $g_k = \nabla f$  bestimmt, die Richtung als  $p_k = -\nabla f$  festgelegt sowie überprüft, dass die Richtung zum Minima gemäß (3.14) gegeben ist.

$$p_k^T * g_k < 0 \quad (3.14)$$

Anschließend wird der nächste Punkt gemäß (3.15) bestimmt und die Schrittweite  $s$  gemäß (3.16) aktualisiert.

$$x_{k+1} = x_k + s * p_k^T \quad (3.15)$$

$$s = d/ds \quad (3.16)$$

Dieses Verfahren wird so lange wiederholt, bis  $|g_{k+1}| = 0$  ist. Die Vorteile bestehen darin, dass es die wenigsten Berechnungen erfordert, nie fehlschlägt und auf unterschiedliche Modelle skaliert werden kann. Nachteilig ist, dass die Anzahl der benötigten Iterationen nicht minimal ist.

### 3.4.2.2. Newton's Method

Die zweite Suchstrategie verwendet die Hesse-Matrix, um die Richtung der Suche zu bestimmen. Dazu werden nach Angabe eines Startwerts  $x_0$  für alle  $k = 0, 1, 2, \dots, n$  die Gradienten und die Hesse-Matrix  $H_k$  berechnet. Die Richtung der Suche erfolgt gemäß (3.17).

$$p_k = H_k^{-1} * g_k \quad (3.17)$$

Anschließend wird der neue Punkt  $x_{k+1}$  sowie die Schrittweite  $s$  wie beim *steepest-descent* berechnet. Die Suche endet, sobald  $|g_{k+1}| = 0$  ist.

Diese Strategie konvergiert schneller als *steepest-descent*. Die Suche kommt jedoch nicht zu einer Lösung, wenn der initiale Punkt zu weit vom Minimum entfernt ist. Des Weiteren müssen die Hesse-Matrix und ihre inverse berechnet werden.

### 3.4.2.3. Quasi-*Newtons Method*

Diese Suchstrategie entspricht der *Newtons Method*, jedoch wird eine approximierte Inverse der Hesse-Matrix  $H_k^{-1} = G_k$  verwendet. Nach Definition des Startpunkts  $x_0$  wird der Gradient  $g_k = \nabla f$  berechnet und  $G_k = I$  der Identitätsmatrix gesetzt. Die Suchrichtung errechnet sich gemäß (3.18). Die Berechnung des nächsten Punktes  $x_{k+1}$  und der Schrittweite  $s$  erfolgt wieder gemäß der *Newton's Method*.

$$p_k = -G_k * g_k \quad (3.18)$$

Im Anschluss wird die Annäherung der Hesse-Matrix angepasst. Dazu wird gemäß (3.19) die Differenz der Gradienten  $\gamma_k$  und gemäß (3.20) die Differenz der Punkte  $\delta_k$  berechnet.

$$\gamma_k = g_{k+1} - g_k \quad (3.19)$$

$$\delta_k = x_{k+1} - x_k \quad (3.20)$$

Die Annäherung der Hesse-Matrix berechnet sich gemäß (3.21).

$$G_{k+1} * \gamma_k = \delta_k \quad (3.21)$$

Das Verfahren wird anschließend so oft wiederholt, bis  $|g_{k+1}| = 0$  ist.

Als weitere Optimierungen existieren zwei Varianten zur Annäherung der Hesse-Matrix nach David-Fletcher-Powell (DFP) gemäß (3.22) und Broyden-Fletcher-Goldfarb-Shanno (BFGS) gemäß (3.23).

$$G_{k+1} = G_k - \frac{G_k \gamma_k \gamma_k^T G_k}{\gamma_k^T G_k \gamma_k} + \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} \quad (3.22)$$

$$G_{k+1} = G_k - \frac{G_k \gamma_k \delta_k^T + \delta_k \gamma_k^T G_k}{\delta_k^T \gamma_k} + \left[ 1 + \frac{\gamma_k^T G_k \gamma_k}{\delta_k^T \gamma_k} \right] \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} \quad (3.23)$$

Zusammenfassend kann gesagt werden, dass alle Suchstrategien entsprechend der Schrittweite variiert werden können. Welche Suchstrategie die beste ist, kann nicht anhand der Vor- oder Nachteile entschieden werden, sondern ist abhängig von der Funktion  $f$  und dem Ausgangspunkt.

## 3.5. Umsetzung der Kalibrierung im DIP-Framework

Die Kalibrierung wurde als zuschaltbarer Filter im DIP-Framework der Abteilung ULF implementiert.

### 3.5.1. Entwicklung der Softwarearchitektur

Dafür wurde die PCL aktualisiert und die entsprechenden Programmaufrufe an die neue Version angepasst. Es wurde ein Datentyp `Ext_Pt` angelegt, der den Anforderungen für die zu verwendenden Algorithmen entspricht. Der Datentyp ermöglicht die Speicherung der Punktkoordinaten, einer Oberflächen-Normalen und einer Sensorposition. Damit die Punktwolken in den unterschiedlichen Bezugssystemen dargestellt werden können, mussten gemäß Abbildung 3.7 einige Änderungen an der bestehenden Codebasis vorgenommen werden.

Zuerst wurde die Verzahnung zwischen Erstellung, Extraktion und Darstellung von Punktwolken aufgelöst und die Funktionalität der Darstellung in die Klasse `pointcloudVisualizer` verlegt. Jede Klasse, die Punktwolken in irgendeiner Art und Weise visualisieren möchte, kann nun von der Klasse `pointcloudVisualizer` erben und die Punktwolke in der geschützten Shared-Pointer-Variablen `cloudToDraw` speichern. Die Visualisierung erfolgt dabei in einem eigenen Thread.

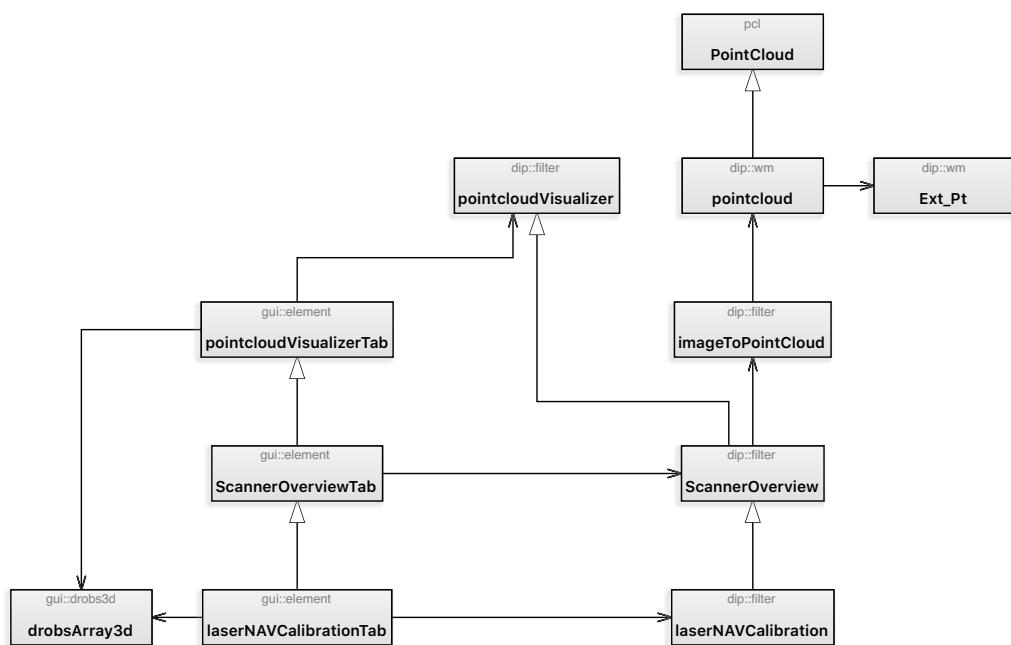


Abbildung 3.7.: Klassendiagramm zur Umsetzung der Kalibrierung im DIP

Des Weiteren wurde die Umwandlung von LiDAR-Scandaten in Punktwolken optimiert. In der bisherigen Implementierung wurden die Punktwolken redundant angelegt und mehrfach kopiert. Durch Verwendung von *shared-pointer*-Variablen konnte die Verarbeitung der Punktwolken effizienter gestaltet werden. Ebenfalls wurde die Klasse `imageToPointcloud` um die Variable `e_coordinate_system` erweitert, bei der es sich um eine Enumeration handelt. Diese Variable definiert das Bezugssystem, für das die Punktwolke aus den Scandaten berechnet werden soll. Zuvor wurden für jedes Bezugssystem eine eigene Punktwolke angelegt und bearbeitet, auch wenn von der aufrufenden Klasse nur die Punktwolke eines Bezugssystems verwendet wurde. Das Bezugssystem kann nun durch Setzen der Variablen variiert werden.

Als Nächstes wurde die Klasse `LaserNAVCalibration` erstellt. Dabei handelt es sich um den Filter, der gemäß des vorgestellten Ansatzes die einzelnen Funktionen aufruft. Die einzelnen Funktionen für Filterung, Sampling und Registrierung befinden sich außerhalb einer Klasse, ausgelagert in der Datei `pointcloudRegistration`. Die Klasse `LaserNAVCalibration` erbt zudem von der Klasse `ScannerOverview`, um die registrierten Punktwolken, den Ursprungsdatensatz sowie einzelne Zwischenprodukte während des Samplings darzustellen.

### 3.5.2. Implementierung der optimierten Lösung

Als nicht-lineare Optimierungsfunktion wird die Implementierung nach Gavin (2011) und Madsen et al. (1999) von Wuttke (2007) verwendet. Dabei handelt es sich um eine *n-variable-optimization*. Die Minimierung beginnt mit der Suchstrategie *steepest-descent* und geht anschließend in die *Newton-Gauß*-Methode über. Bei der *Newton-Gauß*-Methode handelt es sich um eine weitere Optimierung der *Quasi-Newton method*, die jedoch nur auf die Minimierung von nicht-linearen Problemen der kleinsten Quadrate angewendet werden kann.

Die Übergabeparameter sind die zu optimierende Montagepose. Sie besteht aus drei Translationen und drei Rotationen. Die Optimierungsfunktion benötigt ein bestimmtes, im

Idealfall ein überbestimmtes Gleichungssystem, sodass für sechs Unbekannte auch sechs Punktwolken-Paare benötigt werden.

Die Optimierungsfunktion berechnet für jedes Punktwolken-Paar den mittleren Punktstand. Dazu wird eine Initiallösung vorgegeben, die zum Beispiel vom Anwender gemessen oder geschätzt wird. Mit dieser Initiallösung werden nun beide Punktwolken transformiert. Dabei werden die Punkte vom Sensor- ins Trägerbezugssystem überführt. Anschließend wird die zeitlich erste Punktwolke mit der gemessenen Bewegung, die das UA zwischen den Aufnahmen erfahren hat, transformiert.

Bei einer korrekten Montagepose würde die erste Punktwolke nun mit der zweiten übereinstimmen. Ist dies nicht der Fall, bedeutet es, dass die Montagepose angepasst werden muss. Dafür werden die Punktwolkenabstände mit dem ICP bestimmt und in einem Fehlervektor gespeichert. Dieses Verfahren wird für jedes Punktwolken-Paar wiederholt. Anschließend werden alle Fehler aus dem Fehlervektor aufaddiert und die Wurzel gezogen. Dieser Wert dient für die Optimierungsfunktion als Richtwert.

Nach jedem Durchlauf werden die Übergabeparameter gemäß einer mathematischen Metrik verändert und der Fehlervektor erneut berechnet. Diese Veränderung ist unter anderem abhängig von der Schrittweite  $\epsilon$  und den vorherigen Fehlervektoren.

### 3.5.3. Entwicklung der graphischen Oberfläche

Durch die Vererbung der verschiedenen Filter kann zudem nicht nur die Funktionalität des Filters, sondern auch die graphische Oberfläche des Filters übernommen und erweitert werden. Gemäß Abbildung 3.8 ist die graphische Oberfläche des `LaserNAVCalibration`-Filters in drei Teile untergliedert.

Auf der [linken Seite](#) können die verschiedenen Filter- und Samplingfunktionen zur Kalibrierung zu- oder abgeschaltet werden. Dabei wurde Wert auf die gegenseitige Abhängigkeit von Filterfunktionen gelegt. Die Berechnung des Kovarianz-Samplings setzt zum Beispiel die Berechnung der Oberflächen-Normalen voraus. Deaktiviert der Benutzer die Berechnung der Oberflächen-Normalen, wird automatisch auch der Filter des Kovarianz-Samplings

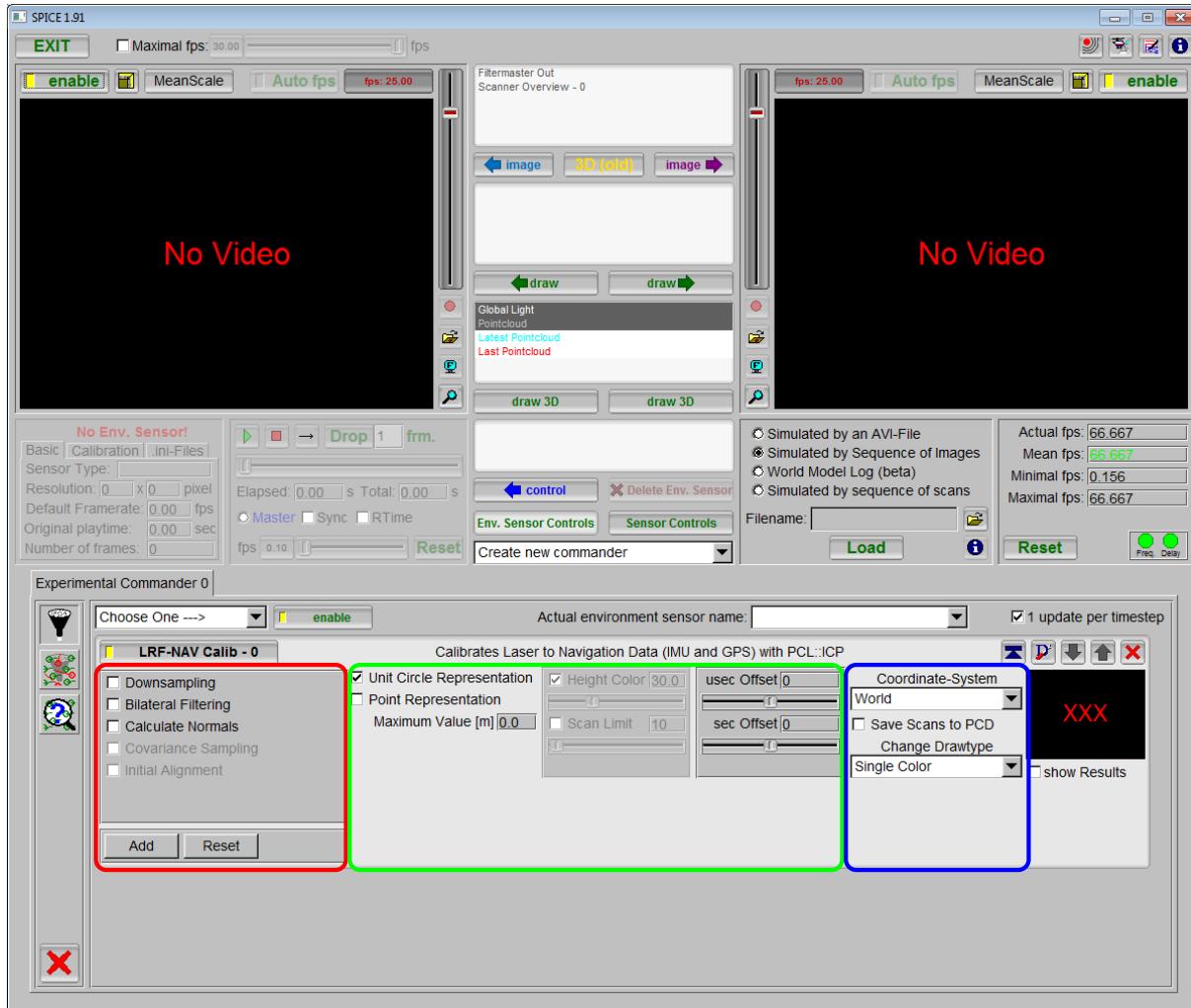


Abbildung 3.8.: Übersicht der graphischen Oberfläche des DIP-Frameworks mit hinzugeschaltetem LaserNAVCalibration-Filter (unterer Bildausschnitt)

deaktiviert. Des Weiteren befinden sich an dieser Stelle ein Add- und ein Reset-Button, wodurch der Ablauf der Kalibrierung gesteuert werden kann.

In der **Mitte** befinden sich die von der Klasse `ScannerOverview` geerbten Steuerelemente zur Visualisierung des Ursprungsdatensatzes. Dies ermöglicht die Parametrierung der Anzahl der anzuzeigenden Punktwolken und die Einstellung der Höhenskala.

Die **rechte Seite** ermöglicht allgemeine Einstellungen zur Visualisierung von Punktwolken, wie das darzustellende Bezugssystem und die Einfärbung der Punktwolke anhand

verschiedener Eigenschaften.

### 3.5.4. Anwendung des Filters zur relativen extrinsischen Kalibrierung

Um die Montagepose eines LiDAR-Sensors zu einer NAV-Sensorik zu kalibrieren, müssen die jeweiligen Scandaten und Navigations-Log-Dateien in das DIP-Framework geladen werden. Dabei ist sicherzustellen, dass die initiale Montagepose in der Initialisierungsdatei der Scandaten wie gewünscht gesetzt ist. Anschließend werden Schlüsselszenen durch den **Add**-Button zu einem Vektor aus Punktwolkenpaaren hinzugefügt. Dabei hat das Punktwolkenpaar einen zeitlichen Abstand von 100 ms. Der **Reset**-Button setzt die bisher gewählten Schlüsselszenen komplett zurück. Hat der Benutzer mindestens sechs Schlüsselszenen hinzugefügt, beginnt die nicht-lineare Optimierung.

Im Konsolen-Ausgabefenster kann nun die Variation der Parameter beobachtet werden. Der Algorithmus endet mit einer minimierten Lösung. Nach Erhalt der neuen Montagepose wird das Punktwolkenpaar mit dem größten Fehlerwert aus dem Vektor entfernt und der Benutzer hat die Möglichkeit, eine weitere Schlüsselszene hinzuzufügen. Durch das erneute Hinzufügen wird die Optimierung mit der nun berechneten Montagepose wiederum durchlaufen.

## 4. Validierung

Zur Validierung der Kalibrierlösung wurden zwei Experimente durchgeführt. Durch das erste Experiment *In-Flight* wird die Problematik der Kalibrierung an einem unbekannten Aufbau analysiert. Die Ergebnisse dienten als Anhaltspunkte für den vorgestellten Kalibrierungsansatz. Mit dem zweiten Experiment *Common-Ground* wurde die Implementierung der automatisierten Kalibrierung durch die Verwendung eines konstruierten und vermessenen Aufbaus validiert. Im Folgenden werden beide Experimente bezüglich ihres Aufbaus, des Ablaufs und den jeweiligen Resultaten erläutert.

### 4.1. Ausgangspunkt der Kalibrierung

Bei der Auswertung eines Filters zur Erkennung von Hindernissen anhand von Kamera und LiDAR-Daten wurden die Aufnahmen einer Kamera mit den Entfernungsinformationen eines LiDARs verknüpft. Dabei trat ein Effekt gemäß Abbildung 4.1 auf, der bei bestimmten Bewegungen dazu führte, dass die Hits des LiDARs fehlerhaft auf das Kamerabild projiziert wurden. Dadurch wurde die Hinderniserkennung stark gestört.

Für das menschliche Auge sind die fehlerhaften Detektionen an den Häuserwänden kaum zu erkennen. In der algorithmischen Analyse führen solche Messfehler zu gravierenden Rechenfehlern. Bisher ist es algorithmisch noch nicht möglich, solche perspektivischen Fehler zu erkennen und zu korrigieren. Alternativ entstand der Gedanke, dass die geschätzte und am Computer nachträglich optimierte Montagepose des LiDAR in bestimmten Situationen zu ungenau sei. Dadurch entstand der Wunsch nach einer automatisierten extrinsischen Kalibrierung zwischen LiDAR-Sensoren und NAV-Sensorik.

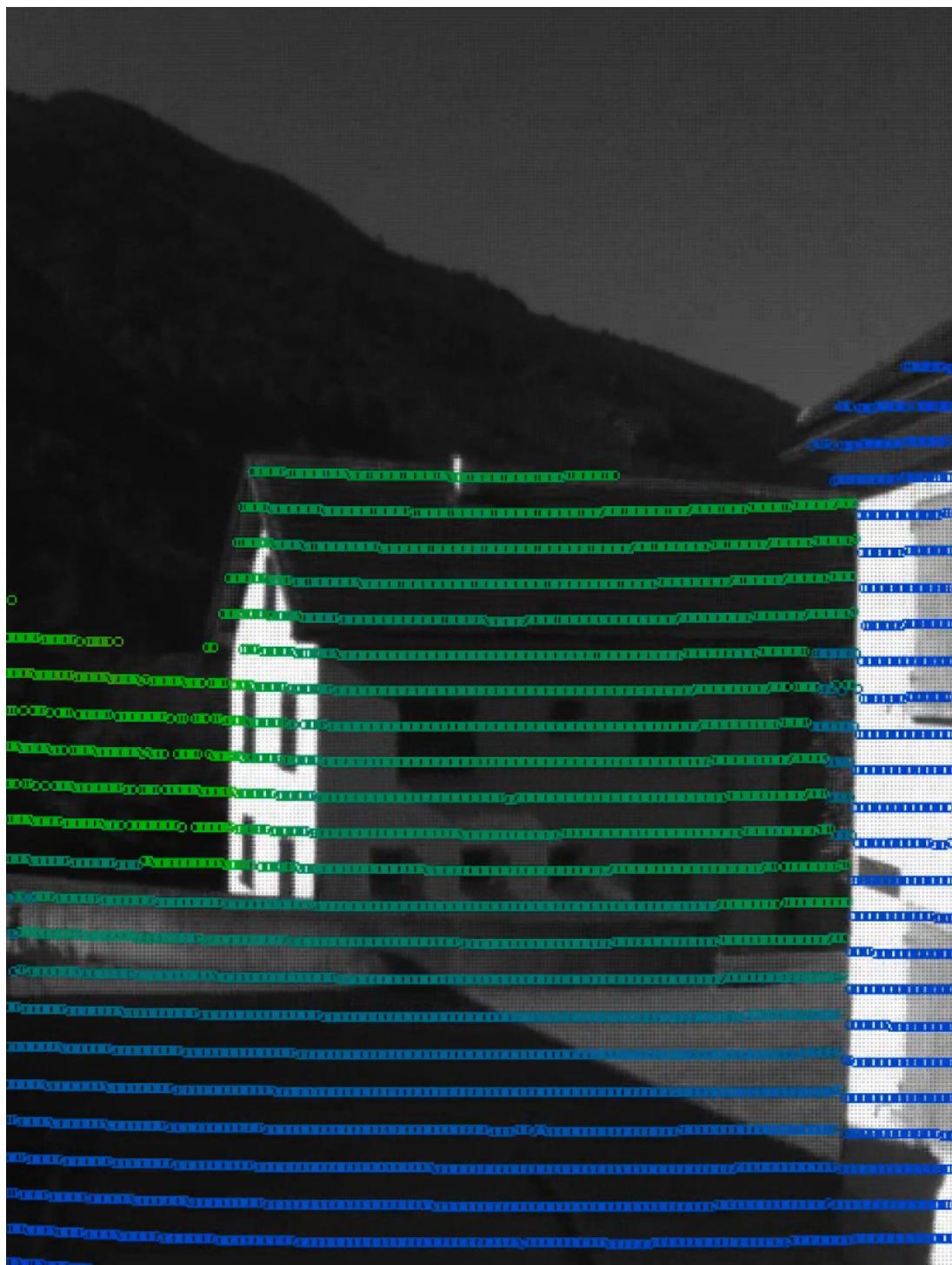


Abbildung 4.1.: Überlagerung von Kamera- und Laserscandaten zur Hinderniserkennung

## 4.2. Verwendete Sensoren

### 4.2.1. Der LiDAR-Sensor

Als LiDAR-Sensor kommt der Velodyne HDL-32e (im Folgenden als Velodyne bezeichnet) zum Einsatz. Entsprechend seines Namens besitzt der Velodyne 32 vertikal angeordnete Halbleiterlaser, die in der  $zx$ -Ebene einen FoV zwischen  $+10^\circ$  und  $-30^\circ$  ermöglichen. Gemäß Abbildung 4.2 rotiert diese Anordnung um  $360^\circ$  um seine  $y$ -Achse. Der Messbereich liegt bei 1 m bis 100 m mit einer Standardabweichung von  $\pm 2$  cm bei 25 m. Die Framerate beträgt dabei 10 Hz, wobei der LiDAR-Sensor je Scan circa 250.000 Bildpunkte als Entfernungsmessung liefert.

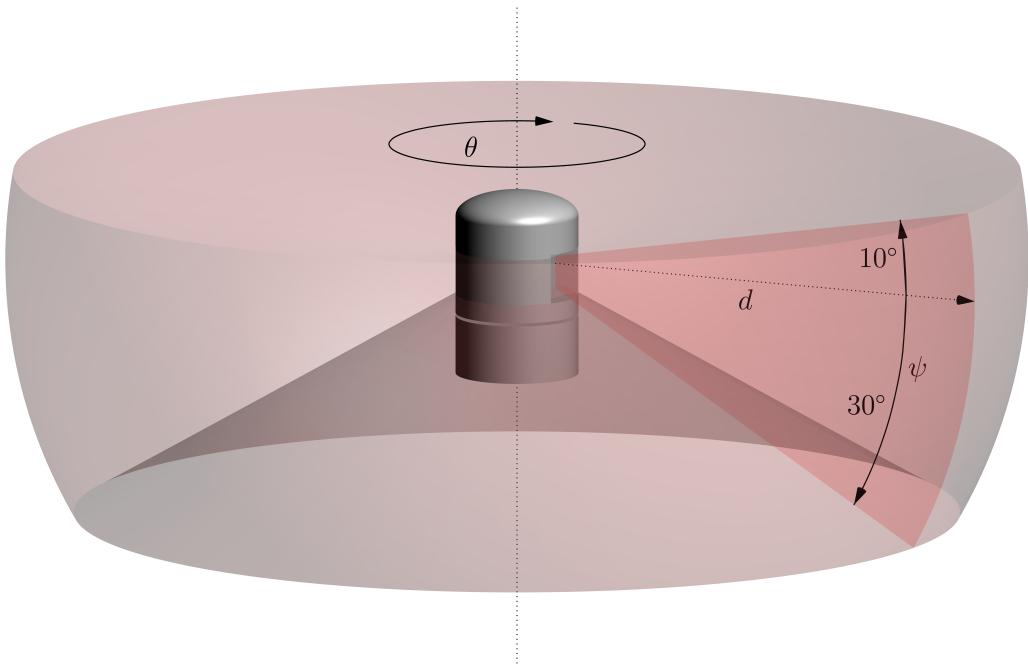


Abbildung 4.2.: Schematische Darstellung des Velodyne HDL-32e

Der LiDAR-Sensor misst den Rotationswinkel  $\theta$ , den Winkel  $\phi$ , die Distanz zum Objekt  $d$ , den Intensitätswert des jeweiligen Hits und einen intern gesetzten Zeitstempel. Die aufgenommenen Sensordaten werden per User Datagram Protocol (UDP) an den Rechner (Fahrzeug- beziehungsweise Flugrechner) weitergeleitet und dort von Kugelkoordinaten in

kartesische Koordinaten umgerechnet.

#### 4.2.2. Die inertiale Messeinheit

Als inertiale Messeinheit wird die iMar IMU iTraceRT-F400-Q gemäß Abbildung 4.3 verwendet. Die IMU bietet eine *Deep-Coupled* Sensorumgebung aus Inertial Navigation System (INS) und GNSS. Das INS wird durch Laserkreisel und eine unbekannte Anordnung von Beschleunigungssensoren realisiert; das GNSS unterstützt NAVSTAR-GPS-, GLONASS- sowie BeiDou-Satelliten.



Abbildung 4.3.: Produktbild der iMAR iTraceRT-F400 Q (iMAR (2013))

Die IMU liefert gemäß iMAR (2013) alle kinematischen Messungen wie Beschleunigung, Winkelgeschwindigkeiten, Lage, Heading, Geschwindigkeit und Position des Trägers mit einer Taktfrequenz von 400 Hz. Die Genauigkeit beträgt dabei 2 cm in der Position,  $0,01^\circ$  bezüglich der Lage,  $< 1 \text{ mg}$  in der Beschleunigung und  $0,02 \frac{\text{m}}{\text{s}}$  in der Geschwindigkeit. Die IMU ist über eine Universal Serial Bus (USB) 2.0 Schnittstelle mit dem Rechner verbunden.

### 4.3. Versuch - *In-Flight*

Der Versuch *In-Flight* repräsentiert den realen Einsatz der Kombination aus LiDAR und IMU am Forschungs-UA des DLR. Dabei sind die relativen Posen der Sensoren zueinander durch Mitarbeiter des DLR geschätzt und optimiert worden. Da dieser Versuch als Urheber des Wunsches nach einer automatisierten Kalibrierung zu betrachten ist, wurden die Messdaten des Versuches soweit möglich analysiert, um das Problem besser zu verstehen.

#### 4.3.1. Aufbau

Der Träger der Sensoren war in diesem Zusammenhang das UA Dragon 50 gemäß Abbildung 4.4 des Herstellers SwissDrones. Die Position und Lage der Sensoren zueinander wurde gemäß Tabelle 4.1 manuell vermessen und durch optische Analyse weiter optimiert.



Abbildung 4.4.: SwissDrones Dragon 50 im Flugversuch

Tabelle 4.1: Händisch ermittelte Rotationswinkel und Abstände der Sensoren auf der Trägerplattform.

Achse	Rotationswinkel [°]	Abstand [mm]
$x$	$-80,0 +/- 10,0$	$240,0 +/- 30,0$
$y$	$0,0 +/- 10,0$	$0,0 +/- 30,0$
$z$	$-90,0 +/- 10,0$	$135,0 +/- 30,0$

Damit das FoV des LiDAR in Flugrichtung nicht eingeschränkt war, wurde der LiDAR um  $10^\circ$  geneigt am Rumpf des UA befestigt.

### 4.3.2. Ablauf

Das Experiment fand am 29.09.2014 von 09:30 Uhr bis 11:00 Uhr auf einem Truppenübungsplatz in St. Luzisteig im Kanton Graubünden in der Schweiz statt. Dabei wurde das UA von Hand durch einen Sicherheitspiloten durch eine Anordnung von Häusern gesteuert und vom Bodenpersonal verfolgt. Das Experiment begann mit der Kalibrierung der inertialen Messeinheit durch das Fliegen von zwei übereinander gelegten Kreisen. Im Anschluss wurde das UA durch großteils translatorische Bewegungen durch die Umgebung gesteuert und anschließend auf dem Startplatz wieder gelandet. Somit genügt dieser Versuch der *loop-closing* Bedingung. Abbildung 4.5 zeigt eine 2-dimensionale Darstellung der Flugtrajektorie.

Die Montagepose des LiDAR führt unweigerlich dazu, dass das FoV zur Seite und nach hinten auf Grund des Fahrgestells und der Grundplatte eingeschränkt wurde. Da dies zu Fehlmessungen führte, wurden die resultierenden Daten durch einen Pass-Through-Filter bereinigt.

Im Anschluss wurden die generierten Punktwolken am Rechner betrachtet. Dabei fiel auf, dass im Datensatz gerade zu Beginn des Fluges kaum markante Hindernisse enthalten sind und die Punktwolken hauptsächlich aus ebenen Bodenaufnahmen bestehen. Beim Einflug

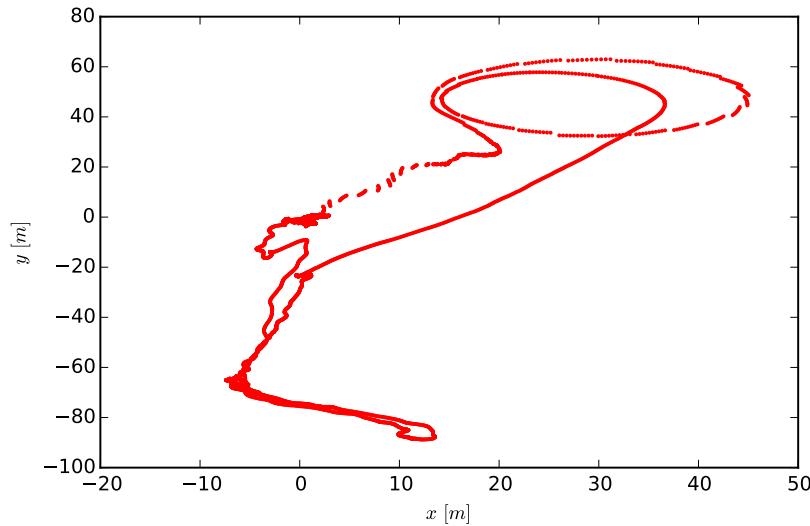


Abbildung 4.5.: Trajektorie des UA während des Versuchs *In-Flight*

in die Häusergegend sind sich bewegende Personen zu erkennen, sodass die Bedingung einer statischen Umgebung nicht zutrifft. Des Weiteren ist der Flug stellenweise holprig, sodass sich das Blickfeld horizontal schnell ändert.

Als Nächstes wurden die Punktwolken in verschiedenen Scan-Abständen mit dem ICP-Algorithmus registriert und die Fehlerwerte geplottet.

### 4.3.3. Ergebnisse

Die generierten Punktwolken dieses Versuchs sind speziell zu Beginn während der Kalibrierung der inertialen Messeinheit sehr bodenlastig, das heißt, es sind kaum Merkmale in Form von Häuserecken oder ähnlichem zu sehen. Dies wirkt sich direkt auf den ICP-Fehler aus. Zu Beginn befindet sich das UA in einer nicht markanten Umgebung und die Registrierung schlägt häufiger fehl. Ab der Hälfte des Fluges bewegt sich das UA in einer Anordnung von Häusern, was sich auch gemäß Abbildung 4.6 positiv auf den ICP-Fehler auswirkt.

Abbildung 4.7 zeigt eine Vergleichsübersicht der mittleren errechneten ICP-Fehler zwischen den Punktwolken für unterschiedliche Scan-Abstände.

Es ist zu erkennen, dass mit steigendem Scan-Abstand auch ein höherer mittlerer

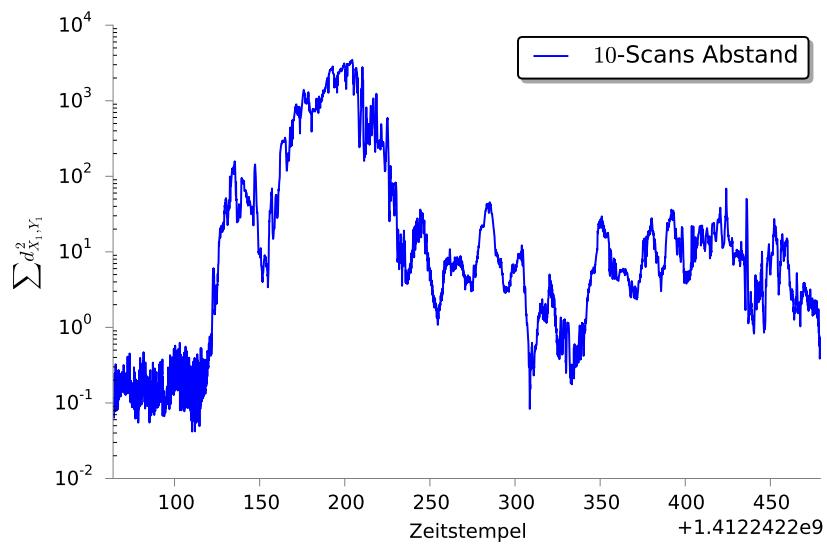


Abbildung 4.6.: Abhangigkeit zwischen dem ICP-Fehler und der Scan-Umgebung

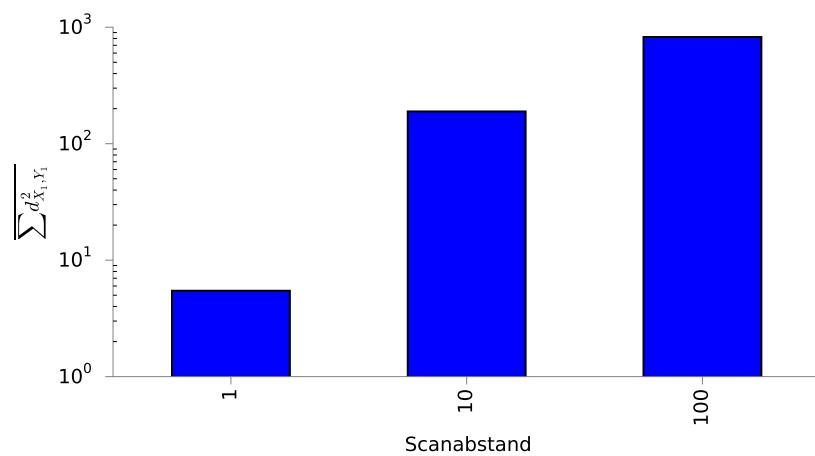


Abbildung 4.7.: Gegenuberstellung der mittleren ICP-Fehler

ICP-Fehler einhergeht. Der Versuch hat gezeigt, das das Resultat einer suxzessiven Registrierung von Punktwolken abhängig ist von der Struktur des Scans, den Abständen der zur Registrierung verwendeten Scans und der Position des LiDARs am Träger.

## 4.4. Versuch - *Common Ground*

Der Versuch *Common Ground* dient der Bestimmung der Genauigkeit und der Anwendbarkeit der Kalibrierungslösung.

### 4.4.1. Aufbau

Der Aufbau besteht aus den zur Kalibrierung benötigten Geräten gemäß Tabelle 4.2, die auf einer gemeinsamen Trägerplattform gemäß Abbildung 4.8 befestigt sind.

Tabelle 4.2: Geräteübersicht.

Anzahl	Gerät	Modell
1	LiDAR	Velodyne HDL-32e
1	IMU	iMAR iTrace RT-F400Q
2	NAVSTAR-GPS-Antennen	Novatel
1	Spannungsversorgung	DLR-Eigenbau
1	Flugcomputer	DLR-Eigenbau

Der Aufbau wurde bezüglich der Winkel und Abstände gemäß 4.3 kalibriert.

Tabelle 4.3: Rotationswinkel und Abstände der Sensoren auf der Trägerplattform.

Achse	Rotationswinkel [°]	Abstand [mm]
$x$	90,0 +/- 5,0	325,0 +/- 10,0
$y$	0,0 +/- 5,0	0,0 +/- 10,0

---

Achse	Rotationswinkel [°]	Abstand [mm]
<i>z</i>	90,0 +/- 5,0	20,0 +/- 10,0

---

#### 4.4.2. Ablauf

Zu Beginn wurde der Aufbau im Forschungstransporter des DLR verbaut. Dabei wurde Wert darauf gelegt, dem LiDAR ein offenes Blickfeld durch die geöffneten Ladeklappen zu ermöglichen. Der LiDAR benötigt keine weitere Inbetriebnahme, er liefert Umgebungsscans per Ethernet-Schnittstelle, sobald er mit einer Spannung versorgt wird.

Die IMU benötigt zur Bestimmung der Position mindestens eine NAVSTAR-GPS-Antenne, deren relative Montageposition gemäß Tabelle 4.4 vermessen wurde.

Tabelle 4.4: Montageabstände zwischen der GPS-Antenne und der IMU.

---

Achse	Abstand [m]
<i>x</i>	-3,03 +/- 0,10
<i>y</i>	0,71 +/- 0,10
<i>z</i>	1,175 +/- 0,10

---

Die Anleitung des Herstellers zur Inbetriebnahme der IMU war nicht eindeutig genug, sodass der Ablauf im Feldversuch ermittelt werden musste. Die verwendete IMU wird hauptsächlich für den Automobilbetrieb eingesetzt. Da für den Automobilbetrieb andere Koordinatensystem-Konventionen als im Bereich der Luft- und Raumfahrt verwendet werden, muss die IMU entsprechend konfiguriert werden. Dazu wurde die Ausrichtung der IMU und dem Träger (in diesem Fall das Forschungsfahrzeug) gemäß Tabelle 4.5 bestimmt.

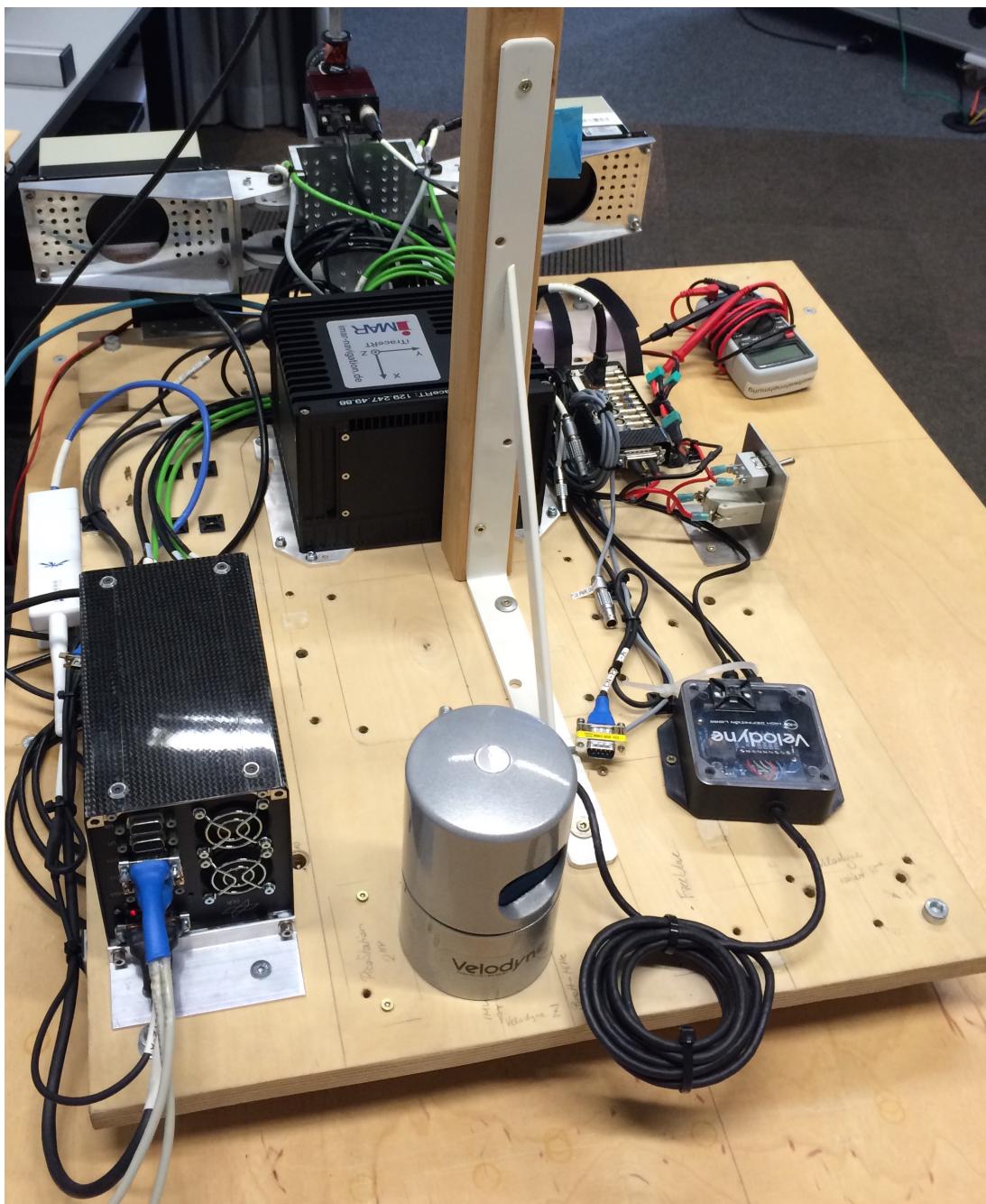


Abbildung 4.8.: Trägerplattform im Versuch *Common-Ground*

Tabelle 4.5: Ausrichtung der IMU im Fahrzeugbezugssystem.

Winkel	Betrag [°]
$\Phi$	0,0 +/- 5,0
$\Theta$	0,0 +/- 5,0
$\Psi$	90,0 +/- 5,0

Das Experiment wurde am 25. Juni 2015 von 15:00 Uhr bis 16:00 Uhr in der Wohngegend Hondelage bei Braunschweig durchgeführt. Die Lokalität und die Uhrzeit wurden gewählt, um während des Experiments eine großteils statische Umgebung zu ermöglichen, sodass die Laserscans nicht durch andere Verkehrsteilnehmer verfälscht werden konnten. Andererseits hätte die so erhaltene Dynamik die Berechnung der Montagepose erschwert. Das Experiment wurde dreimal durchgeführt.

Bei dem Versuch wurden insgesamt 2395 Laserscans in 95,8 Sekunden aufgenommen. Die Auflösung eines Laserscans betrug dabei  $2000 \times 32$  Bildpunkte, die in die Punktwolken übertragen wurden.

Abbildung 4.9 zeigt die aufgenommenen Punktwolken nach der Transformation ins Weltbezugssystem in einer Draufsicht. Die Farbe repräsentiert die relative Höhe der Punkte, wobei die Farben von Rot (nah am Boden) in Grün (circa 10 m über dem Boden) übergehen. Die Strecke des Experiments genügt der *Loop Closing*-Bedingung, sodass eine gleichmäßige Anzahl an Rotationen für die Auswahl der Schlüsselszenen zur Verfügung stehen (drei rechts- und vier Linkskurven). Für die translatorischen Schlüsselszenen stehen acht verschiedene Abschnitte zur Verfügung.

Nach dem Experiment wurden die Schlüsselszenen ausgewählt. Dafür wurden die Bewegungsdaten mit unterschiedlichen Scanabständen<sup>21</sup> (1, 2, 10 und 100) geplottet. Die Auswertung zeigt, dass bei einem Abstand von 100 die Differenz zwischen den Laserscans

<sup>21</sup> Ein Scanabstand bezeichnet die Anzahl an Scans, die zwischen zwei Scanpaaren ignoriert werden.



Abbildung 4.9.: Draufsicht des Versuchs *Common Ground* im Höhenprofil

zu groß war, um noch ausreichende Merkmale zur Registrierung zur Verfügung zu stellen. Bei den Abständen von 1 und 2 Laserscans war hingegen die Geschwindigkeit des Versuchsträgers zu gering, sodass die gemessenen Translationen und Rotationen zu nah aneinander lagen. Aus diesem Grund werden für die weitere Auswertung Abstände von 10 Scans zwischen den Schlüsselszenen verwendet.

Zur Auswahl der rotatorischen Schlüsselszenen wurden gemäß Abbildung 4.10a die Summe der quadratischen Wurzeln aus den einzelnen Winkeln  $\Phi_d$ ,  $\Theta_d$  und  $\Psi_d$  gebildet. Anschließend wurden die Maxima bestimmt, absteigend sortiert und die oberen drei abgespeichert. Für die Translationen wurde gemäß Abbildung 4.10b auf Basis der Wurzel der quadratischen Summen aus den einzelnen Koordinatenanteilen  $x_d$ ,  $y_d$  und  $z_d$  ebenso verfahren. Aus dieser Analyse wurden die Punktwolken-Paare 42, 79 und 200 für die Rotationen und 137, 167 und 190 für die Translationen als Schlüsselszenen ausgewählt.

Der Ansatz wurde dabei unter Vorgabe der gemessenen, gemäß Tabelle 4.3, und der falschen (alle Winkel und Translationen wurden auf 0 gesetzt) Lösung validiert. Dabei

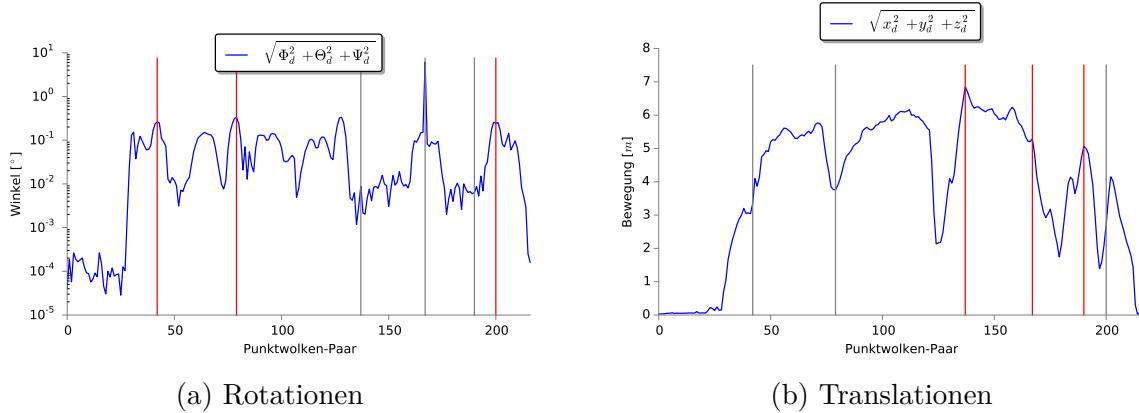


Abbildung 4.10.: Analyse der Punktwolken-Paare auf markante Bewegungen

wurden die Kalibrierungen einmal im Welt- und im Sensorbezugssystem durchgeführt sowie die Punktwolken jeweils (a) ohne Filterung betrachtet, beziehungsweise (b) durch das Voxel-Grid, (c) durch das Kovarianz-Downsampling und (d) Voxel-Grid + Kovarianz-Downsampling reduziert. Die Parameter für die Filter- und Samplingfunktionen entsprachen denen von Stellmacher (2014) ermittelten Standardwerten.

#### 4.4.3. Ergebnisse

Abbildung 4.11 zeigt die Gegenüberstellung der 16 durchgef hrten Messungen. Es ist zu erkennen, dass unter der Vorgabe der korrekten Montagepose die Kalibrierung der Pose im Sensor- wie auch im Weltbezugssystem zur korrekten L sung konvergiert. Im Sensorbezugssystem ist die L sung der Position in Bezug auf die  $x$ - und  $z$ -Achse falsch. Des Weiteren ist zu erkennen, dass die Kalibrierung im Sensorbezugssystem unter der korrekten Vorgabe der L sung im Durchschnitt h here ICP-Fehler vorweist als im Weltbezugssystem.

Bei der falschen Vorgabe ist zu erkennen, dass die Lösung der Lage im Sensor- wie auch im Weltbezugssystem in die korrekte Richtung geht, jedoch um mehrere Größenordnungen zu groß ist. Auf den ersten Blick erscheint dies als eine falsche Lösung. Bei genauerer Betrachtung der Messdaten fällt hingegen auf, dass es sich dabei um positive oder negative Vielfache der korrekten Lösung handelt. Bezüglich der Translation bietet die Kalibrierung

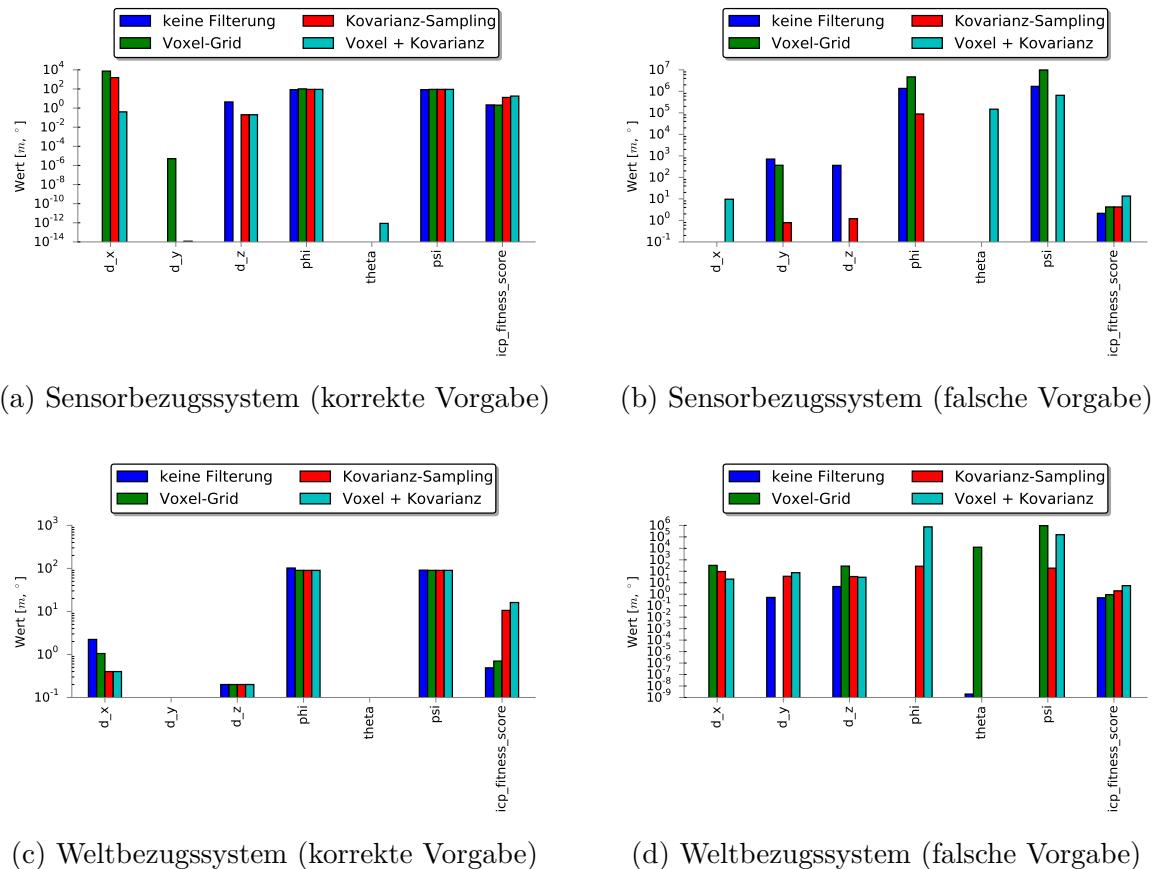


Abbildung 4.11.: Gegenüberstellung der berechneten Montageposen im Sensor- und Weltbezugssystem unter der Vorgabe einer falschen und der korrekten Montagepose

im Sensorbezugssystem die besten Ergebnisse.

Bezüglich der Filterfunktionen kann keine eindeutige Aussage getroffen werden, welche besser geeignet ist. Es ist zu beobachten, dass außer im Weltbezugssystem mit falscher Vorgabe die Kalibrierung ohne Filterung rotatorisch sehr gute Ergebnisse liefert. Auffällig ist, dass die gleichzeitige Anwendung des Kovarianz-Downsampling und des Voxel-Grids im Sensorbezugssystem starke Fehler im Bezug auf  $\Theta$  aufweist. Unter der falschen Vorgabe ist dieser Winkel eine komplette Drehung, sodass er zumindest korrigierbar ist. Die Position kann mit keiner der vorgestellten Methoden eindeutig bestimmt werden.

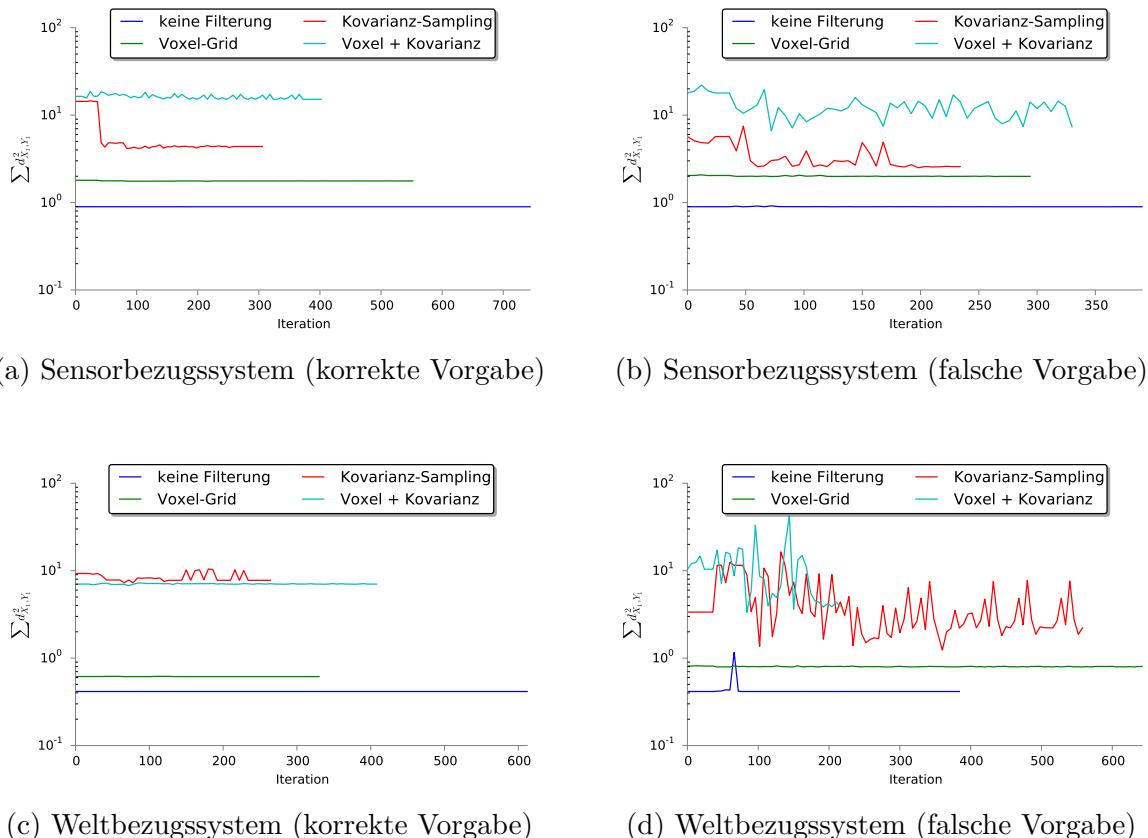


Abbildung 4.12.: Gegenüberstellung der Konvergenz über die Anzahl der benötigten Iterationen und des fortlaufenden ICP-Fehlers

Die Abbildung 4.12 zeigt die Gegenüberstellung der Konvergenz der verschiedenen

Filterungen gegenüber der benötigten Anzahl an Iterationen. Es ist zu erkennen, dass im Falle der korrekten Vorgabe alle Filterungen schneller konvergieren als unter der falschen Vorgabe. Des Weiteren fällt auf, dass unter der falschen Vorgabe größere Optimierungsschritte ausgeführt werden. Daraus resultiert ein deutlich schwankender ICP-Fehler, der jedoch erwartungsgemäß mit fortschreitenden Iterationen abflacht. Bis auf die Außnahme des Weltbezugssystems unter der falschen Vorgabe benötigt die Methode ohne Filterung deutlich mehr Iterationen bis zur Konvergenz. Am schnellsten konvergiert die Kalibrierung unter Einsatz der kombinierten Filterung.

## 4.5. Ergebnisanalyse

Es kann festgehalten werden, dass die Kalibrierung unter Vorgabe einer fast korrekten Montagepose im Weltbezugssystem die besten Ergebnisse erzielt hat. Ebenfalls sind die Rotationen in Form von Vielfachen eindeutiger zu bestimmen. Bezuglich der Translationen wiesen die Messergebnisse einige Schwierigkeiten auf.

Welche Filterfunktion die besten Ergebnisse liefert, ist nicht erkennbar. Des Weiteren wurden die Filterfunktionen mit ihren Standardparametern verwendet, die bezüglich der vorliegenden LiDAR-Scans optimiert werden können. Die zuverlässigsten Ergebnisse liefert die Kalibrierung ohne jegliche Filterung, zumindest, wenn eine korrekte Vorgabe zur Montagepose gegeben ist.

Des Weiteren gilt es zu beachten, dass auch die Trägerplattform sowie die Sensoren im Allgemeinen Fehler aufweisen. Besonders beim Versuch *In-Flight* sind zudem Auswirkungen durch den Flugpfad und der Umgebung nicht auszuschließen. Ebenfalls wirkt sich die Auswahl der Schlüsselszenen auf die Kalibrierung aus, da zu kleine oder zu große Bewegungsänderungen zu Schwierigkeiten in der Registrierung führen.

Alle aufgenommenen und ausgewerteten Daten können in einem iPython Notebook, das sich auf dem beigelegten Datenträger und unter dem GitHub-Repository <https://github.com/gismo141/mastersthesis> befindet, nachverfolgt und weiterführend ausgewertet werden.

## 5. Fazit und Ausblick

In dieser Arbeit wurde ein Ansatz zur automatisierten relativen extrinsischen Kalibrierung von bildgebenden Sensoren zu einer NAV-Sensorik vorgestellt. Ziel war es, durch den Vergleich von paarweise registrierten Punktwolken und den parallel aufgezeichneten Bewegungsdaten eine Montagepose zu bestimmen.

Da der ICP-Algorithmus ohne initiale Ausrichtung in ein lokales Minimum fällt, wurde die Registrierung der Punktwolken in einem nicht-linearen Optimierungsalgorithmus nach Levenberg-Marquardt geschachtelt.

Diese Optimierungsfunktion benötigt für jede Unbekannte einen Startwert, der über einen Liniensuchalgorithmus, zu Beginn *steepest-descent* und anschließend *Newton-Gauß*, optimiert wird. Dazu werden Datensätze durch den Benutzer ausgewählt, die eine markante Bewegung des Trägers in Form von Rotationen und Translationen aufweisen.

Grundlage für die Optimierung ist dabei eine Fehlermetrik. Diese Fehlermetrik entspricht der mittleren euklidischen Distanz zwischen der Punktwolke vor der Bewegung und der Punktwolke nach der Bewegung.

Da die vorliegenden Datensätze die Eigenschaft einer ungleichmäßig verteilten Punktwolke aufweisen, wurden einige Funktionen zur Filterung und für das Downsampling implementiert. Alle vorgestellten Funktionen wurden entsprechend ihrer Standardparameter verwendet.

Die Ergebnisse zeigen, dass trotz Verwendung der Funktionen zur Optimierung der Punktwolken sowie der geschachtelten Optimierung der Montagepose kein globales Minimum garantiert werden kann. Für die weitere Betrachtung der Thematik sollte besonderes Augenmerk auf die Parametrisierung der Filter- und Samplingfunktionen gelegt werden. Am vielversprechendsten ist dabei das *Covariance-Sampling*, da es die Möglichkeit bietet,

planare Flächen zu entfernen und die Punktwolke auf markante Stellen zu reduzieren. Dadurch werden geometrisch stabilere Punktwolken geschaffen und lokale Minima vermieden.

Des Weiteren wurde beobachtet, dass die gleichzeitige Optimierung von Translation und Rotation nicht ideal ist, da die Rotationen einen größeren Einfluss auf das Resultat der Optimierung hatten als die Translationen. Es wird empfohlen, für zukünftige Anwendungen die Optimierung der Translation und Rotation getrennt zu betrachten oder die Parameter unterschiedlich zu gewichten.

Bildgebende Sensoren bieten die Möglichkeit, die Umgebung nicht nur zu vermessen, sondern sie auch zu kategorisieren. Da die Automatisierung von Abläufen immer weiter voranschreitet, wird die Extraktion von Bewegungen meiner Meinung nach in Zukunft einen sehr hohen Stellenwert einnehmen.

Bereits heute werden Fahrzeugführer durch unterschiedliche Assistenzsysteme unterstützt. Im Kraftfahrzeubereich zählen zum Beispiel Geschwindigkeitsregelung, Spur- und Abstandsassistenten sowie Navigationssysteme bereits zum Repertoire der Werksausstattung. In der Luftfahrt halten seit einigen Jahren Autopiloten sowie Systeme zur automatisierten Landung Einzug. Im Bereich der Schifffahrt werden Containerschiffe bereits vollautomatisiert durch eine minimale Besatzung gefahren.

Auch der Gütertransport befindet sich im Wandel zur Automatisierung. So forschen Versandunternehmen wie Amazon bereits an selbstfliegenden UA zur individuellen Paketzustellung; auch in der Überwachung durch Polizeibehörden werden vermehrt automatisierte Systeme eingesetzt.

Die vorliegende Arbeit legt mit der automatisierten Kalibrierung einen Grundstein für zukünftige Entwicklungen.

## Literaturverzeichnis

*Andert, Franz* 2011: Bildbasierte Umgebungserkennung für autonomes Fliegen (DLR eV: PhD thesis).

*Bae, Kwang-Ho/Lichti, Derek* 2007: On-site self-calibration using planar features for terrestrial laser scanners, in: Proceedings of the ISPRS Workshop on Laser Scanning and SilviLaser, 14-19.

*Bellekens, Ben/Spruyt, Vincent/Berkvens, Rafael/Weyn, Maarten* 2014: A Survey of Rigid 3D Pointcloud Registration Algorithms, in: Fourth International Conference on Ambient Computing, Applications, Services and Technologies, 8-13.

*Belshaw, Michael Sweeney/Greenspan, Michael A.* 2009: A high speed iterative closest point tracker on an FPGA platform, in: Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, 1449-1456.

*Bentley, Jon Louis* 1975: Multidimensional Binary Search Trees Used for Associative Searching, in: Commun. ACM 18: 9, 509-517, in: <http://doi.acm.org/10.1145/361002.361007>.

*Benzerrouk, Hamza/Salhi, Hassen/Nebylov, Alexander* 2013: Adaptive „Cubature and Sigma Points“ Kalman Filtering Applied to MEMS IMU/GNSS Data Fusion during Measurement Outlier, in: Journal of Sensor Technology 3: 04, 115.

*Besl, Paul J./McKay, Neil D.* 1992: Method for registration of 3-d shapes, in: Robotics-dL tentative, 586-606.

*Boughorbel, Faysal/Mercimek, Muharrem/Koschan, Andreas/Abidi, Mongi* 2010: A new method for the registration of three-dimensional point-sets: The Gaussian fields framework, in: Image and Vision Computing 28: 1, 124-137.

*Browell, EV/Ismail, S./Grant, WB* 1998: Differential absorption lidar (DIAL) measure-

ments from air and space, in: Applied Physics B 67: 4, 399-410.

*Chan, Ting On/Lichti, Derek D./Belton, David* 2015: A rigorous cylinder-based self-calibration approach for terrestrial laser scanners, in: ISPRS Journal of Photogrammetry and Remote Sensing 99: 84-99.

*Davison, Andrew J./Reid, Ian D./Molton, Nicholas D./Stasse, Olivier* 2007: MonoSLAM: Real-time single camera SLAM, in: Pattern Analysis and Machine Intelligence, IEEE Transactions on 29: 6, 1052-1067.

*Dawes, Beman/Abrahams, David/Rivera, Rene* 1998: boost C++ libraries, in: <http://www.boost.org>.

*Delmerico, Jeff* 2013: PCL Tutorial: The Point Cloud Library By Example, in: [http://www.jeffdelmerico.com/wp-content/uploads/2014/03/pcl\\_tutorial.pdf](http://www.jeffdelmerico.com/wp-content/uploads/2014/03/pcl_tutorial.pdf).

*Demtröder, Wolfgang* 2013: Experimentalphysik 2: Elektrizität und optik, 6. Aufl.

*Droeschel, David/Behnke, Sven* 2011: 3D body pose estimation using an adaptive person model for articulated ICP, in: Intelligent Robotics and Applications, 157-167.

*Edwan, Ezzaldeen* 2013: Novel approaches for improved performance of inertial sensors and integrated navigation systems (Universität Siegen: PhD thesis).

*Gavin, Henri* 2011: The levenberg-marquardt method for nonlinear least squares curve-fitting problems, in: Department of Civil and Environmental Engineering, Duke University: 1-15.

*Gelfand, Natasha/Ikemoto, Leslie/Rusinkiewicz, Szymon/Levoy, Marc* 2003: Geometrically stable sampling for the ICP algorithm, in: 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on, 260-267.

*Goermann, Lukas* 2004: Objektorientierte Bildverarbeitungsalgorithmen zum relativen Hovern eines autonomen Helikopters (Fachhochschule Braunschweig/Wolfenbüttel: Diplomarbeit), Braunschweig.

*Guennebaud, Gaël/Jacob, Benoît/others* 2010: Eigen v3, in: [http://eigen.tuxfamily.org/index.php?title=Main\\_Page](http://eigen.tuxfamily.org/index.php?title=Main_Page).

*Guth, Olaf* 2004: Biologisch inspirierte Bildverarbeitungsalgorithmen zur Realisierung

eines Geländefolgefloges für einen autonomen Kleinhubschrauber (Fachhochschule Braunschweig/Wolfenbüttel: Diplomarbeit), Braunschweig.

*Hervier, Thibault/Bonnabel, Silvere/Goulette, François* 2012: Accurate 3D maps from depth images and motion sensors via nonlinear Kalman filtering, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 5291-5297.

*iMAR, GmbH* 2013: iMAR iTraceRT-F400-Q.

*Kern, Fredie* 2003: Automatisierte modellierung von bauwerksgeometrien aus 3D-laserscanner-daten.

*Köhne, Anja/Wößner, Michael* 2008: Fehlerquellen bei GPS, in: <http://www.kowoma.de/gps/Fehlerquellen.htm>.

*Krause, Stefan/Evert, Robert* 2012: Remission based improvement of extrinsic parameter calibration of camera and laser scanner, in: 12th International Conference on Control Automation Robotics

& Vision (ICARCV), 829-834.

*Kuchling, Horst* 2011: Taschenbuch der Physik, 20. Aufl.

*Lemoine, FG/Kenyon, SC/Factor, JK/Trimmer, RG/Pavlis, NK/Chinn, DS/Cox, CM/Klosko, SM/Luthcke, SB/Torrence, MH/others* 1998: The development of the joint NASA GSFC and NIMA geopotential model EGM96: NASA Goddard Space Flight Center, in: Greenbelt, Maryland 20771: 298.

*Levinson, Jesse Sol* 2011: Automatic laser calibration, mapping, and localization for autonomous vehicles.

*Lindenberger, Joachim* 1993: Laser-Profilmessungen zur topographischen Geländeaufnahme (Bayerische Akademie der Wissenschaften: PhD thesis).

*Lowe, David G.* 1999: Object recognition from local scale-invariant features, in: Computer vision, 1999. the proceedings of the seventh iEEE international conference on, Band 2, 1150-1157.

*Maddern, Will/Harrison, Alastair/Newman, Paul* 2012: Lost in translation (and rotation): Rapid extrinsic calibration for 2D and 3D LIDARs, in: Robotics and Automation

(ICRA), 2012 IEEE International Conference on, 3096-3102.

*Madsen, Kaj/Bruun, Hans/Tingleff, Ole* 1999: Methods for non-linear least squares problems.

*May, Stefan/Droeschel, David/Holz, Dirk/Wiesen, Christoph/Fuchs, Stefan/others* 2008: 3D pose estimation and mapping with time-of-flight cameras, in: International Conference on Intelligent Robots and Systems (IROS), 3D Mapping workshop, Nice, France.

*Mendes, Abel/Bento, Luis Conde/Nunes, Urbano* 2004: Multi-target detection and tracking with a laser scanner, in: Intelligent Vehicles Symposium, 2004 IEEE, 796-801.

*Michelson, Albert Abraham* 1925: The Effect of the Earth's Rotation on the Velocity of Light, I., in: The Astrophysical Journal 61: 137.

*Muja, Marius/Lowe, David G.* 2008: FLANN - Fast Library for Approximate Nearest Neighbors, in: <http://www.cs.ubc.ca/research/flann/>.

*Muja, Marius/Lowe, David G.* 2012: Fast Matching of Binary Features, in: 2012 Ninth Conference on Computer and Robot Vision (CRV), 404-410.

*Muja, Marius/Lowe, David G.* 2014: Scalable nearest neighbor algorithms for high dimensional data, in: IEEE Transactions on Pattern Analysis and Machine Intelligence 36: 11, 2227-2240.

*Rusinkiewicz, Szymon/Levoy, Marc* 2001: Efficient variants of the ICP algorithm, in: Third International Conference on 3-D Digital Imaging and Modeling, 2001. Proceedings., 145-152.

*Rusu, Radu Bogdan/Blodow, Nico/Beetz, Michael* 2009: Fast point feature histograms (FPFH) for 3D registration, in: IEEE International Conference on Robotics and Automation, 2009. ICRA'09., 3212-3217.

*Rusu, Radu Bogdan/Cousins, Steve* 2011: 3D is here: Point Cloud Library (PCL), in: 2011 IEEE International Conference on Robotics and Automation (ICRA), 1-4.

*Schroeder, Will J./Lorensen, Bill/Martin, Ken* 2004: The Visualization Toolkit.

*Sheehan, Mark/Harrison, Alastair/Newman, Paul* 2011: Self-calibration for a 3D laser,

in: The International Journal of Robotics Research: 0278364911429475.

*Stellmacher, Sebastian* 2014: Navigationsdatenkorrektur durch 3D Registrierung (Otto-von-Guericke-Universität Magdeburg: Master's thesis).

*Talaya, J./Alamus, R./Bosch, E./Serra, A./Kornus, W./Baron, A.* 2004: Integration of a terrestrial laser scanner with GPS/IMU orientation sensors, in: Proceedings of the XXth ISPRS Congress, Band 35, 1049-1055.

*The National Coordination Office for Space-Based Positioning, Navigation, and Timing* 2013: Selective availability, in: <http://www.gps.gov/systems/gps/modernization/sa/>.

*The United States Government* 1995: Global Positioning System Standard Positioning Service Signal Specification, 2. Aufl.

*Wulfmeyer, Volker/Bösenberg, Jens* 1998: Ground-based differential absorption lidar for water-vapor profiling: assessment of accuracy, resolution, and meteorological applications, in: Applied Optics 37: 18, 3825-3844.

*Wuttke, Joachim* 2007: lmfit, in: <http://apps.jcns.fz-juelich.de/doku/sc/lmfit>.

*Yang, Jiaolong/Li, Hongdong/Jia, Yunde* 2013: Go-ICP: solving 3D registration efficiently and globally optimally, in: Computer Vision (ICCV), 2013 IEEE International Conference on, 1457-1464.