



Masterarbeit

# Navigationsdatenkorrektur durch 3D Registrierung

Autor:

Sebastian Stellmacher

21. November 2014

Erstprüfer:

Prof. Dr. Frank Ortmeier, OvGU

Zweitprüfer:

Dr.-Ing. Gordon Strickert, DLR

Wissenschaftliche Betreuung:

Dipl.-Wirt.-Inf. Stefan Krause, DLR

**Stellmacher, Sebastian**

Matrikel-Nr.: 201821

*Navigationsdatenkorrektur durch 3D Registrierung*

Masterarbeit, Otto-von-Guericke-Universität Magdeburg, 2014.

# Kurzfassung / Abstract

Für den sicheren und autonomen Betrieb unbemannter Luftfahrzeuge in unbekannten Gebieten ist die Erkennung von Hindernissen von großer Bedeutung. Dafür sind die Luftfahrzeuge mit verschiedenen Sensoren, wie Laserscanner und Navigationssysteme, ausgestattet. Bei der Verwendung dieser Laserscanner werden Punktwolken von der Umgebung generiert. Diese Punktwolken werden sukzessive in einen Octree überführt, der so als Repräsentation der globalen Hinderniskarte dient. Die Verortung der einzelnen Punktwolken findet mittels GPS und Inertialplattform statt. Da den Daten dieser Navigationssysteme Fehler anhaften, ist ein Verfahren entwickelt worden, das eine Korrektur anhand der Scandaten vornimmt. Dieses Verfahren basiert auf dem Iterative Closest Point Algorithmus und führt eine 3D Registrierung der Punktwolken mit dem globalen Octree durch. Um dieses Verfahren in naher Echtzeit verwenden zu können, finden unter anderem Strategien zur Reduzierung der Punktmengen Anwendung.

In order to ensure safe and autonomous operations of unmanned aircrafts in unknown territory, the detection of obstacles is of great importance. Therefore, aircrafts are equipped with various sensors, like laser scanners and navigation systems. By utilising a laser scanner, point clouds of the environment are generated. These point clouds are successively transferred into an octree data structure that is used as a global obstacle map. The positioning of the point clouds is done via GPS and inertial measurement unit. Since the data of these navigation systems is not error-free, a method is developed that is able to correct the errors by using the scan data. This method is based on the Iterative Closest Point algorithm and performs a 3D registration of the point clouds with the global octree. In order to run this method in near real-time, strategies for reducing the point sets are applied.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	viii
<b>Tabellenverzeichnis</b>	x
<b>Abkürzungsverzeichnis</b>	xi
<b>1 Einführung</b>	1
1.1 Aufgabenstellung . . . . .	2
1.2 Anforderungen . . . . .	2
1.3 Gliederung der Arbeit . . . . .	3
<b>2 Grundlagen</b>	4
2.1 Sensorik . . . . .	4
2.1.1 Laserscanner . . . . .	4
2.1.2 GPS . . . . .	6
2.1.3 Inertialplattform . . . . .	10
2.2 Datenstrukturen . . . . .	12
2.2.1 Punktwolke . . . . .	12
2.2.2 Baumstrukturen . . . . .	12
2.3 Mathematische Grundlagen . . . . .	16
2.3.1 Statistik . . . . .	16
2.3.2 Matrix Algebra . . . . .	17
2.3.3 Hauptkomponentenanalyse . . . . .	20
2.4 Geometrische Grundlagen . . . . .	22
2.4.1 Herleitung der Sensorkoordinaten . . . . .	23
2.4.2 Koordinatentransformation . . . . .	24
<b>3 Analyse</b>	25
3.1 Feine Registrierung . . . . .	26
3.2 Verwendung des Octrees zur Registrierung . . . . .	33
3.3 Themenverwandte Arbeiten . . . . .	34
3.4 Ablauf des entwickelten Verfahrens . . . . .	37
<b>4 Selektion von Punkten</b>	39
4.1 Reduzierung der Punkte im Scandatensatz . . . . .	39
4.1.1 Berechnung von Normalvektoren . . . . .	40
4.1.2 Geometrische Stabilität . . . . .	42
4.1.3 Analyse verschiedener Sampling Verfahren . . . . .	44

---

4.1.4 Covariance Sampling . . . . .	48
4.2 Eingrenzung des Suchraums . . . . .	50
<b>5 Iterative Closest Point</b>	<b>53</b>
5.1 Bestimmung von korrespondierenden Punkten . . . . .	54
5.1.1 Analyse verschiedener Methoden . . . . .	54
5.1.2 $k$ D-Baum . . . . .	57
5.2 Wichtung von Punktpaaren . . . . .	58
5.2.1 Analyse verschiedener Wichtungsmethoden . . . . .	58
5.3 Verwerfung von Punktpaaren . . . . .	60
5.3.1 Analyse verschiedener Verwerfungsmethoden . . . . .	60
5.3.2 RANSAC . . . . .	63
5.4 Auswahl der zu minimierenden Fehlermetrik . . . . .	65
5.4.1 Singulärwertzerlegung . . . . .	65
<b>6 Evaluierung</b>	<b>68</b>
6.1 Versuchsaufbau . . . . .	68
6.2 Bewertungsprinzip . . . . .	73
6.3 Auswertung . . . . .	77
6.3.1 Weitere Octreeauflösungen . . . . .	82
6.3.2 Visuelle Darstellung der Ergebnisse . . . . .	86
6.3.3 Betrachtung der Höhenkorrektur . . . . .	88
6.3.4 Überprüfung der nahen Echtzeitanforderung . . . . .	89
<b>7 Zusammenfassung und Ausblick</b>	<b>93</b>
<b>A Herleitungen</b>	<b>95</b>
A.1 Herleitung der Point-to-Plane Minimierung . . . . .	95
A.2 Beweis zur Anwendbarkeit der SWZ zur Rotationsfindung . . . . .	97
<b>B Weitere Ergebnisse</b>	<b>99</b>
B.1 Absolute Werte . . . . .	99
B.2 Visuelle Ergebnisse . . . . .	103
B.3 Vergleich der Fehlermetriken . . . . .	104
<b>C Beispiel zur Suche des nächsten Nachbarn in einem <math>k</math>D-Baum</b>	<b>106</b>
<b>Literaturverzeichnis</b>	<b>109</b>

# Abbildungsverzeichnis

2.1	Technischer Aufbau eines Zeilenscanners. Nach [PB07]	5
2.2	Reflexionsarten (der emittierte Laserstrahl ist in blau hervorgehoben). Nach [Ker03]	6
2.3	Der Schnittpunkt dreier Kugeln zur Bestimmung der Position auf der Erde. Nach [Zog11]	7
2.4	Geometrische Darstellung des DOP-Wert. FB steht für Fehlerbereich und E für Entfernung. Nach [Zog11]	9
2.5	Binärer Suchbaum der Höhe 3	13
2.6	Beispiel eines Octree, bei dem weiße Zellen als frei, Magentafarbene als belegt, und freie Zellen als unbekannt definiert sind. Nach [WHB <sup>+</sup> 10]	13
2.7	Darstellung verschiedener Octree Auflösungen anhand des ca. 19 cm hohen Stanford-Hasen	14
2.8	Beispiel eines Kd-Baums. Nach [BCKO08]	15
2.9	Darstellung der $y=x$ Spiegelachse und den Eigenvektoren der Spiegelmatrix	19
2.10	Darstellung der Beispieldaten und dessen Hauptkomponenten	21
2.11	Darstellung der verwendeten Koordinatensysteme. Nach [And11]	22
2.12	Umrechnung zwischen Kugelkoordinaten und kartesischen Koordinaten	23
3.1	Angleichung einer Eingangspunktwolke (rot) an eine Zielpunktwolke (blau) durch den ICP. Bei ausgewählten Punkten der Eingangspunktwolke sind die nächsten Nachbarn mit einer Linie und die korrekten Korrespondenzen mit einer gepunkteten Linie hervorgehoben.	29
3.2	Beispiel der Point-to-plane Fehlermetrik	30
3.3	Beispielregistrierung eines Scandatensatz als Punktfolge (rot) mit der globalen Hinderniskarte als Octree in 2D. Belegte Knoten sind blau markiert.	34
3.4	Ablaufdiagramm des entwickelten Ansatzes	37
4.1	Kamera- und Laserscannerdaten auf einem Teil des DLR Geländes	41
4.2	Geometrisch instabile 3D Formen. Unter jeder Form steht die Anzahl und der Typ der instabilen Transformationen. Nach [GIRL03]	43
4.3	Vergleich verschiedener Sampling Methoden anhand des Stanford-Hasen. Die gesamte Punktfolge ist in schwarzen Punkten dargestellt und die abgetasteten Punkte sind größer und magentafarben hervorgehoben.	46

4.4	Vergleich verschiedener Sampling Methoden anhand eines Scandatensatzes, der auf dem DLR-Gelände aufgenommen wurde. Es handelt sich um eine Vogelperspektive, die am unteren Rand Vegetation und Teile von drei PKWs zeigt, am oberen Rand eine Mauer und links davon einen Transporter. Die gesamte Punktwolke ist in schwarzen Punkten dargestellt und die abgetasteten Punkte sind größer und magentafarbend hervorgehoben. . . . .	47
4.5	Vogelperspektive des horizontalen Sichtfelds des Laserscanners (grün). Irrelevante Bereiche sind in rot dargestellt . . . . .	51
5.1	Beispiel der Operationen Teilen und Schrumpfen des Box-Dekompositionsbäums [AMN <sup>+</sup> 94] . . . . .	56
5.2	Darstellung mehrerer Methoden zum Verwerfen von Korrespondenzen. Verworfene Punktpaare sind als gepunktete Linie gekennzeichnet.	61
6.1	midiARTIS auf dem Sensorträger. 1) HDL-32E Laserscanner 2) Bildverarbeitungsrechner 3) Flugrechner 4) Telemetrie 5) GPS 6) IMU 7) Stromversorgung . . . . .	69
6.2	Versuchsstrecke auf dem DLR-Gelände . . . . .	70
6.3	Referenzoctree der Versuchsstrecke mit einer Auflösung von 1 m . . . . .	72
6.4	Referenzoctree der Versuchsstrecke im Vergleich zum Satellitenbild . . . . .	72
6.5	Darstellung der vier Gütekriterien. Die belegten Knoten des Referenzoctree sind blau und die des rekonstruierten Octrees rot umrandet . . . . .	74
6.6	Beispiel der Hausdorff-Distanz. Die weiteste Distanz in beide Richtungen ist jeweils durch einen größeren Pfeil gekennzeichnet. . . . .	76
6.7	Prozentuale Veränderung des Matthews Korrelationskoeffizient bei einer Octreeauflösung von 0,5 m . . . . .	78
6.8	Prozentuale Veränderung der Jensen-Shannon Divergenz bei einer Octreeauflösung von 0,5 m . . . . .	79
6.9	Prozentuale Veränderung der durchschnittlichen Hausdorff-Distanz bei einer Octreeauflösung von 0,5 m . . . . .	80
6.10	Unterschiedliche Registrierungsergebnisse bei verschiedenen Ausgangsfehlern . . . . .	81
6.11	Prozentuale Veränderung des Matthews Korrelationskoeffizient bei weiteren Octreeauflösungen . . . . .	82
6.12	Prozentuale Veränderung der Jensen-Shannon Divergenz bei weiteren Octreeauflösungen . . . . .	83
6.13	Prozentuale Veränderung der durchschnittlichen Hausdorff-Distanz bei weiteren Octreeauflösungen . . . . .	84
6.14	Unterschiedliche Registrierungsergebnisse bei verschiedenen Auflösungen . . . . .	85
6.15	Vergleich des rekonstruierten Octrees mit dem Referenzoctree . . . . .	87
6.16	Vergleich des rekonstruierten Octrees mit dem Referenzoctree im Nahbereich . . . . .	88
6.17	Vergleich des rekonstruierten Octrees mit dem Referenzoctree in der Seitenansicht . . . . .	88
6.18	Test . . . . .	89
6.19	Zeitmessungen bei den Octreeauflösungen 0,5, 1 und 2 m . . . . .	90

6.20 Einfluss der Anzahl der Scanpunkte auf die Zeitmessungen . . . . .	91
6.21 Anzahl der gefilterten Octree-Zentren bei den Octreeauflösungen 0,5, 1 und 2 m . . . . .	92
B.1 Vergleich von rekonstruierten Octrees mit dem Referenzoctree . . . . .	103
B.2 Vergleich von rekonstruierten Octrees mit dem Referenzoctree im Nahbereich . . . . .	104
B.3 Vergleich von rekonstruierten Octrees mit dem Referenzoctree in der Seitenansicht . . . . .	104
C.1 Suche des nächsten Nachbarn im <i>kD</i> -Baum (1. Schritt) . . . . .	106
C.2 Suche des nächsten Nachbarn im <i>kD</i> -Baum (2. Schritt) . . . . .	107
C.3 Suche des nächsten Nachbarn im <i>kD</i> -Baum (3. Schritt) . . . . .	107
C.4 Suche des nächsten Nachbarn im <i>kD</i> -Baum (4. Schritt) . . . . .	108

# Tabellenverzeichnis

2.1	GPS Ortungsgenauigkeit [Zog11] . . . . .	8
2.2	IMU Klassifizierung anhand der Fehlerparameter [Edw13] . . . . .	11
2.3	Zweidimensionaler Beispieldatensatz für die PCA . . . . .	20
4.1	Konditionszahlen der Scandatensätze Stanford-Hase und DLR-Gelände	46
6.1	Konfusionsmatrix bei binärer Betrachtung der Octreeknoten . . . . .	73
6.2	MCC Ausgangsfehler der drei Durchläufe bei einer Octreeauflösung von 0,5 m . . . . .	78
6.3	JSD Ausgangsfehler der drei Durchläufe bei einer Octreeauflösung von 0,5 m . . . . .	79
6.4	DHD Ausgangsfehler der drei Durchläufe bei einer Octreeauflösung von 0,5 m . . . . .	79
6.5	MCC Ausgangsfehler der drei Durchläufe bei den Octreeauflösung von 1 und 2 m . . . . .	82
6.6	JSD Ausgangsfehler der drei Durchläufe bei den Octreeauflösung von 1 und 2 m . . . . .	83
6.7	DHD Ausgangsfehler der drei Durchläufe bei den Octreeauflösung von 1 und 2 m . . . . .	84
6.8	Absolute Veränderung der DHD in Metern für die drei Durchläufe . .	86
B.1	Absolute Werte des Matthews Korrelationskoeffizient bei einer Octreeauflösung von 0,5 m . . . . .	99
B.2	Absolute Werte der Jensen-Shannon Divergenz bei einer Octreeauflösung von 0,5 m . . . . .	100
B.3	Absolute Werte der durchschnittlichen Hausdorff-Distanz bei einer Octreeauflösung von 0,5 m . . . . .	100
B.4	Absolute Werte des Matthews Korrelationskoeffizient bei einer Octreeauflösung von 1 m . . . . .	100
B.5	Absolute Werte der Jensen-Shannon Divergenz bei einer Octreeauflösung von 1 m . . . . .	101
B.6	Absolute Werte der durchschnittlichen Hausdorff-Distanz bei einer Octreeauflösung von 1 m . . . . .	101
B.7	Absolute Werte des Matthews Korrelationskoeffizient bei einer Octreeauflösung von 2 m . . . . .	101
B.8	Absolute Werte der Jensen-Shannon Divergenz bei einer Octreeauflösung von 2 m . . . . .	102

B.9	Absolute Werte der durchschnittlichen Hausdorff-Distanz bei einer Octreeauflösung von 2 m . . . . .	102
B.10	Vergleich der Point-to-Point und Point-to-Plane Fehlermetrik beim normalen Anwendungsfall . . . . .	105
B.11	Vergleich der Point-to-Point und Point-to-Plane Fehlermetrik beim Durchlauf mit translativen Fehlern . . . . .	105
B.12	Vergleich der Point-to-Point und Point-to-Plane Fehlermetrik beim Durchlauf mit rotatorischen Fehlern . . . . .	105

# Abkürzungsverzeichnis

3D-NDT	3D Normal Distribution Transform
CEP	Circular Probable Error (Kreisfehlerwahrscheinlichkeit)
DGPS	Differential Global Positioning System
DHD	Durchschnittliche Hausdorff-Distanz
DLR	Deutschen Zentrum für Luft- und Raumfahrt e. V.
DOP	Dilution-of-Precision (Verringerung der Genauigkeit)
FN	False Negative
FP	False Positive
GNSS	Globales Navigationssatellitensystem
GPS	Global Positioning System
HDL-32E	Velodyne High Definition LiDAR mit 32 simultan-operierenden Laserstrahlen
ICP	Iterative Closest Point Algorithmus
IMU	Inertial Measurement Unit (Inertialplattform)
JSD	Jensen-Shannon Divergenz
KLD	Kullback-Leibler Divergenz
LiDAR	Light detection and ranging
MCC	Matthews Korrelationskoeffizient
ND	Durchlauf normaler Anwendungsfall
NED	North East Down (Koordinatensystemkonvention)
PCA	Principal Components Analysis (Hauptkomponentenanalyse)
PR	Precision Rate

---

RANSAC	Random Sample Consensus
RD	Referenzdurchlauf
RFD	Durchlauf mit rotatorischen Fehlern
RMS	Root Mean Square
RR	Recall Rate
SLAM	Simultaneous Localization and Mapping (Simultane Lokalisierung und Kartenerstellung)
SWZ	Singulärwertzerlegung
TFD	Durchlauf mit translativen Fehlern
TINN	Triangle Induced Nearest Neighbor
TN	True Negative
TP	True Positive
UA	Unmanned Aircraft (Unbemanntes Luftfahrzeug)
WLAN	Wireless Local Area Network (Drahtloses lokales Netzwerk)

# 1. Einführung

Unbemannte Luftfahrzeuge ([UA](#)) werden im Bereich der Flugrobotik angesiedelt und sind in der Lage Missionen zu übernehmen, die für die Piloten der bemannten Luftfahrzeuge gefährlich oder monoton wären. Diese zum Teil sehr komplexen Missionen beinhalten unter anderem Flüge in Katastrophengebieten, Inspektionsarbeiten an Versorgungsleitungen oder das Manövrieren zwischen Hochhäusern bei Verkehrsüberwachungen [[FJ08](#)].

Das Institut für Flugsystemtechnik des Deutschen Zentrums für Luft- und Raumfahrt e. V. ([DLR](#)) am Standort Braunschweig führt Forschungsarbeiten in diesem Gebiet durch. Die Abteilung für Unbemannte Luftfahrtfahrzeuge konzentriert sich vor allem auf die Umweltwahrnehmung, Flugregelung und Missionsplanung für [UAs](#). Hierfür stehen verschiedene Arten von Flugversuchsträgern zur Verfügung, wobei es sich unter anderem um Flächenflugzeuge und Drehflügler handelt. Ziel ist der sichere und autonome Betrieb dieser Flugversuchsträger in unbekannten Umgebungen. Um den Anforderungen der Missionen gerecht zu werden, müssen die Luftfahrzeuge in der Lage sein, Hindernisse in unbekannten Gebieten wahrzunehmen und autonome Ausweichmanöver zu fliegen.

Die Pfadplanungsansätze arbeiten unter anderem mit globalen Hinderniskarten, um dem [UA](#) eine kollisionsfreie Fortbewegung zu ermöglichen. Die an den Versuchsträger montierten Umweltsensoren, wie Stereokameras und Laserscanner, sind in der Lage Hindernisse zu detektieren. Die Inklusion der lokalen Hindernisdaten, welche sich im Sensorkoordinatensystem befinden, in die globale Karte erfordert eine Transformation unter Verwendung der bekannten Sensorpose. Die Pose, definiert durch Position und Lage des Sensors, wird anhand eines globalen Navigationssatellitensystems ([GNSS](#)) und einer Inertialplattform ([IMU](#)) ermittelt. Dabei können mögliche Abweichungen, beispielsweise durch Rauschen oder Sprünge, die den Navigations-

daten der **GNSS** und der **IMU** anhaften, zu einer fehlerhaften Verortung der lokalen Sensorhindernisdaten in die Karte führen. Die Konsequenz dieser falschen Verortung kann eine Kollision des Vehikels mit einem Hindernis sein.

## 1.1 Aufgabenstellung

Um den Einfluss der Abweichungen der Navigationsdaten auf die Verortung von Hindernissen zu reduzieren, wird im Rahmen dieser Masterarbeit ein Ansatz zur Navigationsdatenkorrektur konzipiert und umgesetzt. Dieser Ansatz bezieht die jeweils aktuelle Sensormessung, welche als Punktwolke vorliegt, mit in die Schätzung der Sensorpose ein. Eine Möglichkeit ist die 3D Registrierung, bei der die Abweichung zwischen Punktwolken gesucht wird. Auf Basis dieser Abweichung kann eine translativ und rotatorische Verschiebung geschätzt werden. Um eine Ausgangslösung für die Abweichungsschätzung zu erhalten, wird der aktuelle Sensordatensatz mit den bereitgestellten Navigationsdaten der **GNSS** und **IMU** transformiert.

Da die globale Hinderniskarte jedoch nicht als Punktwolke sondern in Form eines Octree vorliegt, werden für die Schätzung nicht zwei identische Datenstrukturen verwendet. Daher soll der Registrierungsansatz erweitert werden, um einen Vergleich zwischen Punktwolke und Octree zu ermöglichen. Dabei muss die Abweichung berücksichtigt werden, die dem Aufbau des Octrees anhaften. Weiterhin ist zu analysieren welche Genauigkeitsunterschiede bei unterschiedlichen Auflösungen des Octrees entstehen. Darüber hinaus soll untersucht werden welche minimale Auflösung der Octree aufweisen darf um eine optimale Schätzung zu ermöglichen.

Da die globale Karte generell größer ist als der einzelne Sensordatensatz, soll der Suchraum vor dem Einsatz des Registrierungsalgorithmus stark eingeschränkt werden, um eine echtzeitnahe Umsetzung zu ermöglichen. Es sollen daher Ansätze präsentiert werden, mit denen eine Reduktion des Suchraums realisiert werden kann. Außerdem soll aufgrund der Größe der einzelnen Sensordatensätze untersucht werden, ob durch eine Reduktion dieser Daten eine Performanzsteigerung bei einer vergleichbaren Genauigkeit für die Navigationsdatenkorrektur erreicht werden kann.

## 1.2 Anforderungen

Basierend auf der Aufgabenstellung lassen sich Anforderungen ableiten, welche nachfolgend in einer nummerierten Auflistung dargestellt werden. Dies bietet die Möglichkeit im späteren Verlauf dieser Arbeit auf die Anforderungen Bezug nehmen zu können.

1. Optionale Korrektur der Navigationsdaten zur verbesserten Verortung der Laserscanner-Daten. Verifikation durch visuelle Betrachtung und Anwendung eines geeigneten Bewertungsprinzips.
2. Der entwickelte Ansatz soll mit Punktwolken und Octrees sowie einer Kombination aus beiden funktionieren
3. Echtzeitnahe (1 - 2 Hz) Umsetzung
4. Soll eine Einschränkung des Suchraums unabhängig vom Laserscanner-Typs ermöglichen

### 1.3 Gliederung der Arbeit

Nachdem in diesem Kapitel eine Einführung in die Thematik und die Aufgabenstellung gegeben wurde, beschreibt das nächste Kapitel die verwendeten Sensoren sowie die theoretischen Grundlagen. In [Kapitel 3](#) wird die Problemstellung analysiert und eine geeignete Registrierungsmethode ausgewählt. Zudem wird die Verwendung des Octrees bei der Registrierung thematisiert und ein Ablauf für das Verfahren festgelegt. Anschließend wird in [Kapitel 4](#) geschildert welche Punkte für die Registrierung verwendet werden und wie der Suchraum für die globale Karte eingeschränkt wird. In [Kapitel 5](#) wird detailliert auf den Registrierungsalgorithmus eingegangen. Dabei werden verschiedene Methoden zur Verbesserung und Beschleunigung betrachtet und für die Verwendung in dieser Arbeit ausgewählt. Im darauffolgenden Evaluationskapitel wird ein Versuch beschrieben, der im Rahmen dieser Arbeit durchgeführt wurde. Anhand der Daten dieses Versuchs wird eine Bewertung des gewählten Registrierungsansatzes durchgeführt. Abschließend wird in [Kapitel 7](#) eine Zusammenfassung der Arbeit und ein Ausblick gegeben.

## 2. Grundlagen

### 2.1 Sensorik

Damit sich unbemannte Luftfahrzeuge autonom bewegen können, werden sie mit verschiedenen Sensoren ausgestattet. Zum einen erlauben Umgebungssensoren, wie Laserscanner, die Wahrnehmung lokaler Hindernisse. Zum anderen kann durch einen GNSS-Sensor und einer IMU eine globale Bestimmung der Pose des Flugversuchsträgers erfolgen. Nachfolgend wird die Funktionsweise der verwendeten Sensoren erläutert.

#### 2.1.1 Laserscanner

Laserscanner sind aktive Sensoren, die anhand von ausgesendeten Laserstrahlen, die Distanz zu Objekten in ihrer Umgebung bestimmen können. Sofern die gepulsten Strahlen auf ein Objekt treffen, werden sie reflektiert. Der Laserscanner ist auf diese Weise in der Lage die Strahlen wieder zu empfangen. Anhand der Lichtlaufzeit dieses Vorgangs lassen sich mit Hilfe der Lichtgeschwindigkeit Entfernungsmessungen durchführen, da die Laufzeit sich proportional zu der Distanz verhält. Dieses Verfahren wird auch Light Detection and Ranging ([LiDAR](#)) genannt und ähnelt der Radar-Technologie, bei der Radiowellen anstelle von Licht verwendet werden. [OC12]

Laserscanner tasten die Umgebung größtenteils zeilenweise ab, indem in einer horizontalen Zeile sequentiell Laserstrahlen in einem bestimmten Winkelabstand versendet werden. Für jeden dieser Strahlen kann anhand der Distanz, dem Austrittswinkel und der Position des Sensors, ein Punkt in Kugelkoordinaten berechnet werden. Bei Laserscannern mit mehreren Scanzeilen wird zudem der vertikale Winkel zur

korrekten Bestimmung des Messpunkts benötigt. Diese Punkte können in ein dreidimensionales kartesisches Koordinatensystem umgerechnet und anschließend zu einer Punktfolge zusammengefasst werden (siehe Abschnitt 2.4.1). Diese Punktfolge dient als Repräsentation des gescannten Bereichs. [OC12]

Das zeilenweise Scannen kann technisch auf verschiedene Arten umgesetzt werden. Abbildung 2.1 veranschaulicht ein optisch-mechanisches Prinzip anhand eines rotierenden Spiegels und eines Motors. Weitere Verfahren nutzen einen schwingenden Spiegel oder einen Facettenspiegel. [PB07]

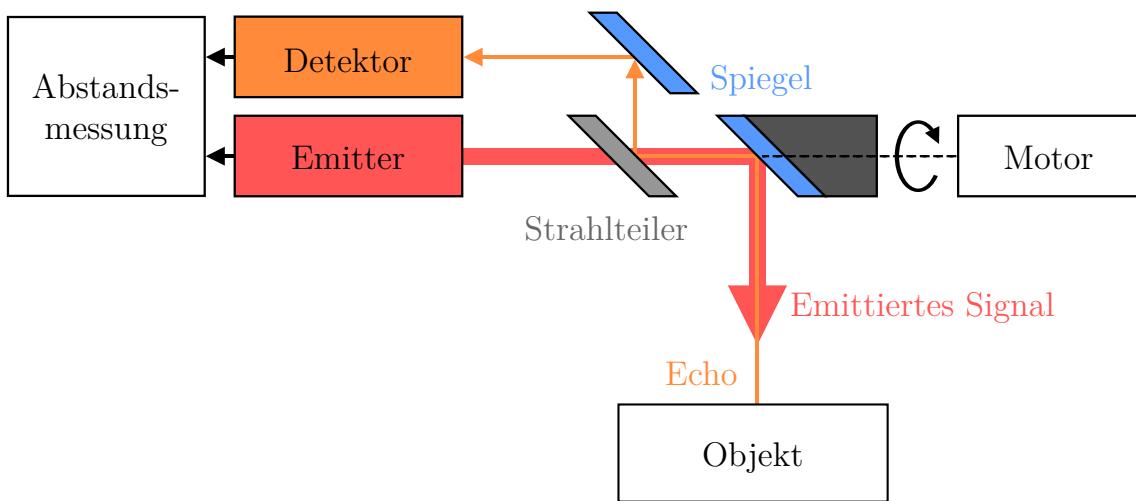


Abbildung 2.1: Technischer Aufbau eines Zeilenscanners. Nach [PB07]

Das Lasermesssystem sendet einen Laserstrahl, welcher von einer Spiegelkonstruktion um  $45^\circ$  abgelenkt wird. Der Spiegel wird mit Hilfe eines Motors in festen Winkelabständen um eine Achse rotiert. Diese Winkelabstände definieren zugleich die Winkelauflösung des Laserscanners. Nach dem Aussenden des Laserstrahls, kehrt das reflektierte Licht wieder zurück und wird durch einen Strahlteiler und einem zusätzlichen Spiegel zurück zum Lasermesssystem geleitet. Dieses Verfahren beschreibt das Scannen von einer Zeile, wobei der Spiegel für weitere Zeilen gekippt werden kann, also nicht mehr einen Winkel von  $45^\circ$  aufweist [PB07]. Ist der Laserscanner nicht stationär muss beachtet werden, dass jeder Messpunkt eines Scanvorgangs aufgrund des sequentiellen Abtastens von einer anderen Position aufgenommen wird.

Bei der Reflexion der Laserstrahlen kann zwischen mehreren Arten unterschieden werden, welche auf den Eigenschaften der gemessenen Oberfläche basieren und Einfluss auf die Genauigkeiten des Messvorgangs haben. Hierbei handelt es sich um die reguläre Reflexion (2.2a), die diffuse Reflexion (2.2b) und die diffuse Reflexion mit Spiegelung (2.2c). Die reguläre Reflexion tritt bei perfekt spiegelnden Oberflächen auf, bei denen der Einfallswinkel gleich dem Ausfallwinkel ist. Dies bietet für das Scanverfahren Probleme, da das reflektierte Licht den Laserscanner nicht wieder er-

reicht und somit keine Entfernung gemessen werden kann. Die reguläre Reflexion kann unter anderem bei Glas-, Metall- und Wasseroberflächen auftreten. Bei der diffusen Reflexion, die bei rauen Oberflächen zustande kommt, werden die Strahlen gleichmäßig in alle Richtungen zurückgestrahlt. In der Natur tritt jedoch vorrangig die dritte Reflexionsart auf, die eine Mischung aus regulärer und diffuser Reflexion entspricht. [Ker03]

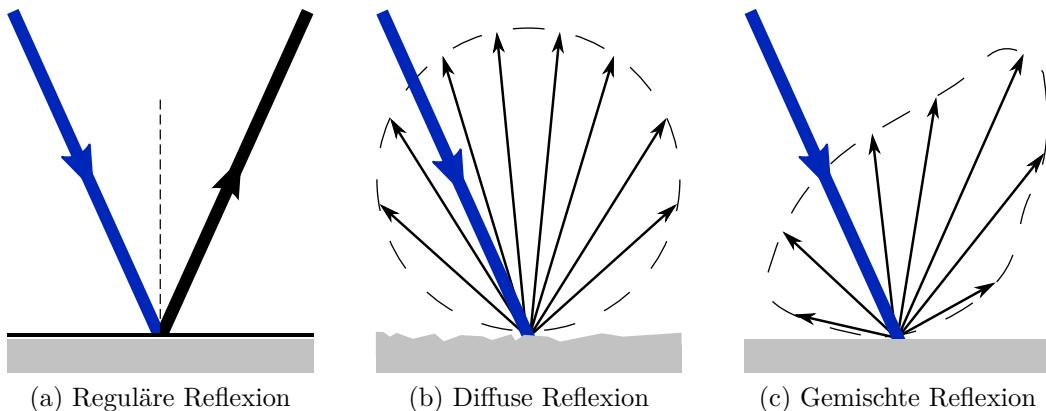


Abbildung 2.2: Reflexionsarten (der emittierte Laserstrahl ist in blau hervorgehoben). Nach [Ker03]

Weitere Messungenauigkeiten können auftreten wenn ein Strahl von mehreren Oberflächen reflektiert wird. Aufgrund dieses verlängerten Weges trifft die Proportionalität zwischen Lichtlaufzeit und Distanz zum Objekt nicht mehr zu. Daher entstehen grobe Ausreißer innerhalb der Scandaten. [Ker03]

Die Vorteile von Laserscannern gegenüber anderen Umgebungssensoren, wie beispielsweise Stereokameras und Time-of-Flight-Kameras, liegen in der konstanten Tiefenauflösung und der Genauigkeit im Zentimeterbereich. Jedoch sind sowohl der Zeitaufwand zur Erfassung einer kompletten Szene als auch die Kosten eines Laserscanners deutlich höher. Im Gegensatz zu Kameras sind Laserscanner jedoch nicht auf ausreichende Beleuchtung der Umgebung angewiesen. [Sch11]

Aufgrund dieser Vorteile finden Laserscanner Anwendung in unterschiedlichen Bereichen. Neben der Umgebungserkennung für mobile unbemannte Vehikel, werden sie beispielsweise zur Objektvermessung bei Bauwerken [Ker03] und zur Verformungsdetektion bei Staudämmen [PB07] verwendet.

### 2.1.2 GPS

Das Global Positioning System (**GPS**) ist ein satellitengestütztes Ortungs- und Navigationssystem und gehört somit zu den Global Navigation Satellite Systems. Mit Hilfe von 31 Satelliten, die in einer Höhe von über 20.000 km auf sechs verschiede-

nen Umlaufbahnen um die Erde kreisen, ist es weltweit möglich den Standort mit einer Genauigkeit von ca. 1 mm bis 12 m und die Uhrzeit mit einer Genauigkeit von ca. 5 bis 40 ns zu bestimmen. Um dies zu ermöglichen, sind die Umlaufbahnen um  $55^{\circ}$  zum Äquator geneigt, sodass von jeder Position auf der Erde eine Funkverbindung zu mindestens vier Satelliten gewährleistet ist. Jeder Satellit ist mit bis zu vier Atomuhren ausgestattet und sendet ständig seine aktuelle Position und Uhrzeit. Anhand der Laufzeit des Signals zum GPS-Empfänger und der Lichtgeschwindigkeit, lässt sich die Entfernung des Satelliten berechnen. Demnach kann, wie in Abbildung 2.3 dargestellt, eine Kugel um die Satelliten platziert werden, dessen Radius der ermittelten Entfernung zum Empfänger entspricht. Der Schnittpunkt der Kugeln bestimmt die Position des GPS-Empfängers. Theoretisch genügen bereits drei Satelliten zur Positionsbestimmung. Dies ist jedoch nur der Fall wenn der Empfänger selbst über eine Atomuhr verfügt, da sonst die Laufzeit nicht richtig kalkuliert werden kann und somit ein Zeitfehler auftritt. Wobei ein Zeitfehler von 1  $\mu\text{s}$  bereits einem horizontalen Entfernungsfehler von 300 m entspricht. [Zog11]

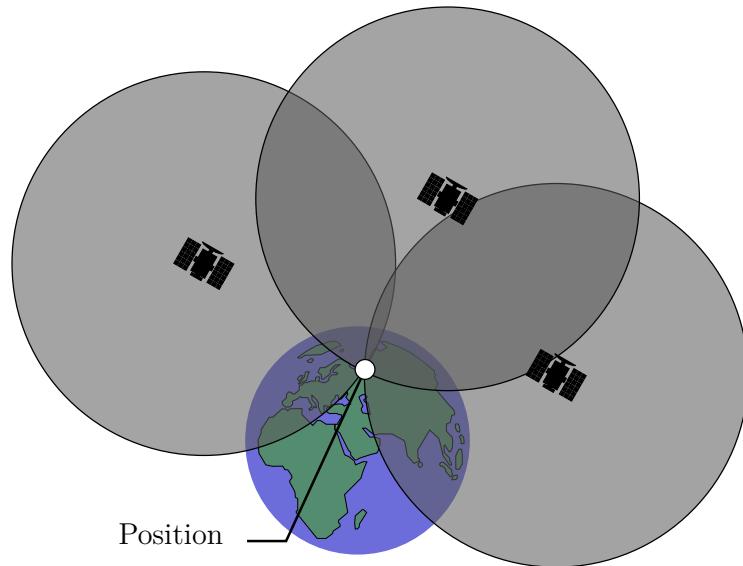


Abbildung 2.3: Der Schnittpunkt dreier Kugeln zur Bestimmung der Position auf der Erde. Nach [Zog11]

## Fehlerarten

Auf Grundlage der Arbeit von Zogg [Zog11] werden in diesem Abschnitt die verschiedenen Fehlerarten bei der Positionsbestimmung mit dem Global Positioning System beschrieben und mit den dazugehörigen horizontalen Fehlern in Tabelle 2.1 zusammengefasst.

Da selbst die momentan präzisesten Zeitgeber, die Atomuhren, eine geringfügige Abweichung von der tatsächlichen Zeit nicht verhindern können, entsteht bei

der Verwendung der Satellitenuhren ebenfalls ein Fehler bei der Positionsbestimmung. Beispielsweise beträgt die Abweichung der Atomuhr CS2 der Physikalischen Technischen Bundesanstalt nach einem Jahr 12 Milliardstel Sekunden (12 Nanosekunden) [Bun12], was einem Fehler im Bereich von ca. 3 - 4 m zur Folge hat. Um diesen Fehler gering zu halten, werden die Atomuhren an Bord der Satelliten regelmäßig synchronisiert. Eine weitere Fehlerquelle ist die Position des Satelliten in der Umlaufbahn, da diese nur mit einer Genauigkeit von ca. 1 – 5 m bekannt ist. Außerdem ist die Annahme, dass sich die Satellitensignale konstant mit der Lichtgeschwindigkeit bewegen nicht immer korrekt. Dies trifft nur in einem Vakuum zu. Jedoch werden beim Durchqueren der Ionosphäre (60 – 1.000 km oberhalb der Erde) und Troposphäre (0 – 15 km oberhalb der Erde) die Signale durch die Luftfeuchte und die Dichte der ionisierten Gasmoleküle verlangsamt. Dies führt zu einem Positionsfehler.

Der verwendete GPS-Empfänger trägt ebenfalls zur Ungenauigkeit bei der Messung der Laufzeit bei, indem er den Zeitpunkt des ankommenden Signals nicht genau bestimmen kann. Befindet sich der Empfänger außerdem in der Nähe von hohen Hindernissen, wie Gebäuden oder Bäumen, kann es durch terrestrische Reflexionen zum Mehrwegempfang kommen. Dabei wird das Signal von dem Hindernis abgelenkt und gelangt zeitversetzt zum Empfänger. Das Signal wird nun durch die Überlagerung mit dem reflektierten Signal verfälscht.

Fehlerursache	Fehler mit GPS
Satellitenuhren	1,5 m
Satellitenposition	1,5 m
Einfluss der Ionosphäre	3,0 m
Einfluss der Troposphäre	0,7 m
Einfluss des GPS-Empfängers	0,5 m
Mehrwegempfang	1,0 m
<b>Totaler RMS-Fehler</b>	<b>4,0 m</b>

Tabelle 2.1: GPS Ortungsgenauigkeit [Zog11]

Wie in Tabelle 2.1 dargestellt, lässt sich aus den hier aufgeführten Fehlern ein RMS-Fehler berechnen, welcher den quadratischen Mittelwert aller Fehler bezeichnet. Des Weiteren hat die Satellitenkonstellation einen großen Einfluss auf den Gesamtfehler. Je näher die Satelliten beieinander stehen, desto ungenauer ist die Messung. Diese Messungsgenauigkeit wird mit einer skalaren Größe, dem Dilution-of-Precision-Wert (DOP-Wert), ausgedrückt.

Verdeutlicht wird der Einfluss der geometrischen Anordnung der Satelliten in Abbildung 2.4. In 2.4a und 2.4b sind jeweils zwei Satelliten in verschiedenen Stellungen zueinander abgebildet. In der vorherigen Abbildung 2.3 wurde zur Vereinfachung

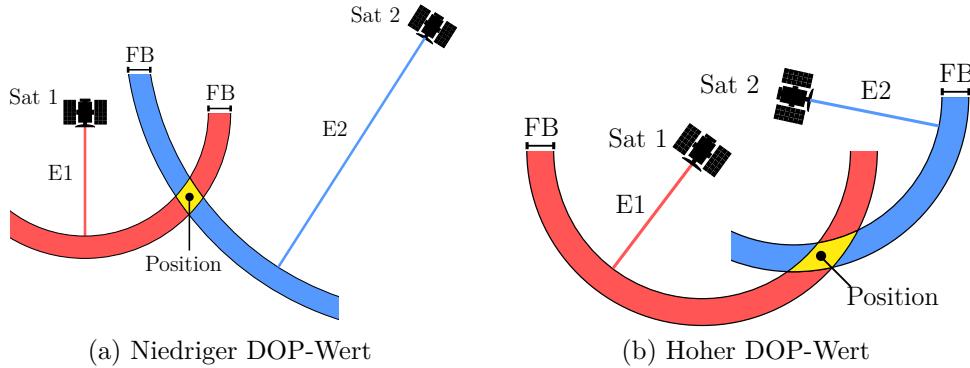


Abbildung 2.4: Geometrische Darstellung des DOP-Wert. FB steht für Fehlerbereich und E für Entfernung. Nach [Zog11]

davon ausgegangen, dass keine Messungenauigkeiten bestehen und daher nur die Oberfläche der Kugel betrachtet werden muss. Durch die in diesem Kapitel aufgeführten Fehler wird jedoch deutlich, dass diese Ungenauigkeiten als Varianz mit einbezogen werden müssen und daher nicht nur die Oberfläche betrachtet werden kann. Die Fehlerbereiche um die Entfernungsradien der beiden Satelliten überschneiden sich und ergeben somit den Gesamtfehlerbereich bei der Positionsbestimmung. Dieser Bereich wird in der Mathematik auch als schleifender Schnitt bezeichnet. Auf der rechten Abbildung ist zu sehen, dass dieser Bereich deutlich größer ist, da die Satelliten näher positioniert sind. Somit ist der **DOP**-Wert ein Faktor von dem der Gesamtfehler proportional abhängig ist.

$$\text{Gesamtfehler} = \text{DOP} \cdot \text{RMS-Fehler}$$

Der tatsächliche Wert des **DOP**-Wert wird durch die Volumengröße eines Tetraeders bestimmt, welches die Positionen der Satelliten sowie die Empfängerposition als Eckpunkte verwendet. Je kleiner das eingeschlossene Volumen, desto größer der **DOP**-Wert.

## DGPS

Zur Verbesserung der Genauigkeit von **GPS** gibt es verschiedene Ansätze. Eine Möglichkeit ist der Einsatz eines Differential Global Positioning System (**DGPS**), das auf der Laufzeitmessung basiert. Hierbei werden **GPS**-Referenzstationen zur Hilfe genommen, die ihre exakte Position kennen. Somit können diese Stationen die Differenz zwischen der tatsächlichen und der durch die Laufzeit berechneten Entfernung zu den sichtbaren Satelliten berechnen. Die Differenz für jeden Satelliten kann nun als Korrekturwert verwendet werden, der über Funk an **GPS**-Empfänger im Umkreis von 200 km übermittelt wird. Anhand der Korrekturwerte lässt sich nun die Position des Empfängers genauer bestimmen. Infolgedessen reduzieren sich die Fehlerarten

auf das Empfängerrauschen und den Mehrwegeempfang, wodurch der RMS-Fehler einen Wert von ca. 1,2 m erreicht. [Zog11]

### 2.1.3 Inertialplattform

Eine Inertialplattform kombiniert verschiedene Inertialsensoren zum Bestimmen der relativen Position und Ausrichtung des Sensorträgers im Raum. Zur Bestimmung der sechs Freiheitsgrade werden typischerweise drei Beschleunigungssensoren und drei gyroskopische Sensoren verbaut. Die drei Beschleunigungssensoren sind dabei orthogonal angeordnet, sodass die lineare Beschleunigung für die  $x$ -,  $y$ - und  $z$ -Achse gemessen werden kann. Anhand der Signale der Sensoren wird durch zweifache Integration die translatorische Bewegung berechnet. Die gyroskopischen Sensoren sind nach demselben Prinzip angebracht und messen den Roll-, Nick- und Gierwinkel, aus denen sich durch einfache Integration die Rotationsbewegung bestimmen lässt. [TW04]

#### Fehlerarten

IMUs sind anfällig gegenüber verschiedenen Arten von Fehlern, die in [Inc14] beschrieben sind und folgend aufgeführt werden. Problematisch bei diesen Fehlern ist, dass durch die einfache Integration bei der Rotationsbewegung aus einem nicht-korrigierten Fehler ein linearer Fehler bei der Geschwindigkeit und ein quadratischer Fehler bei der Positionsbestimmung entsteht. Bei Fehlern der Beschleunigungssensoren entstehen sogar quadratische Fehler bei der Geschwindigkeit und kubische Fehler bei der Positionsbestimmung.

Außerdem haftet der Inertialplattform ein sogenannter Bias an, welcher die Ungenauigkeit der Messung zum wahren Wert beschreibt. Befindet sich die IMU über einen längeren Zeitraum in einer stationären Position kann der Bias gemessen werden, in dem die Beschleunigungswerte der vertikalen Achse mit der Beschleunigung der Schwerkraft verglichen werden. Die Differenz, die bei jedem Neustart der IMU neu berechnet werden muss, kann dann von den Messwerten subtrahiert werden. Aufgrund von mechanischer Belastung und Temperaturänderungen in der Umgebung oder durch Erhitzen des Sensors verändert sich dieser initiale Bias jedoch mit der Zeit, was zu Fehlern führen kann.

Eine weitere Fehlerquelle ist der Skalierungsfaktor, welcher die Relation zwischen Input und Output beschreibt. Beispielsweise liegt bei einem Input von  $10 \text{ m/s}^2$  und Output von  $10.2 \text{ m/s}^2$  der Skalierungsfaktor-Fehler bei 2%. Demnach unterliegt der Output einem linearen Effekt, der sich proportional zum Input verhält, aber anders

skaliert ist. Bei hoher Beschleunigung oder Rotation kann der Skalierungsfaktor sich nicht-linear verändern, was zu Problemen bei der Korrektur führt.

Zudem liegt beim Messen von konstanten Signalen zufälliges Rauschen vor, welches einen Random Walk-Fehler<sup>1</sup> darstellt. Der Messwert eines stationären Beschleunigungssensors sollte 0 ergeben. Durch das Rauschen liegt der statistische Durchschnitt der Signale bei 0 aber der momentan gemessene Wert variiert leicht. Bei der darauf folgenden Integration vergrößert sich der Fehler weiter und führt zu einer ungenauen Berechnung.

Nach Edwan [Edw13] lassen sich IMUs nicht anhand von Technologien sondern anhand der Fehler in Klassen einteilen. Tabelle 2.2 zeigt die 5 verschiedenen Klassen Consumer, Automotive, Tactical, Navigation II und Navigation I sowie die typischen Fehlerbereiche die ihnen anhaften. Diese Kategorisierung ist eine Erweiterung der Arbeit von Jekeli [Jek01], der zunächst eine Einteilung in Tactical, Navigation und Strategic-Military vornahm. Aufgrund der Zunahme von kostengünstigeren, weniger akkuraten IMUs ist eine weitere Einteilung jedoch sinnvoll. Beim Bestimmen der Kategorie ist es nicht notwendig, dass alle Parameterwerte in einer Klasse liegen. Als Beispiel dient die Inertialplattform HG1700, dessen Bias-Stabilität bei den gyroskopischen Sensoren in die Navigation I-Kategorie und bei der Bias-Stabilität der Beschleunigungssensoren in die Automotive-Kategorie fällt. Da jedoch alle anderen Werte der Tactical-Klasse zuzuordnen sind, klassifiziert der Hersteller Honeywell die IMU als Tactical [Hon12].

	Fehler-parameter	Consumer	Automotive	Tactical	Navigation I	Navigation II
Gyros. Sensoren	Bias Stabilität (°/h)	>200	10-200	1-10	0,1-1	<0,01
	Skalierungs-faktor (PPM)	-	>500	200-500	100-200	2-50
	Winkel Random Walk (°/h/ $\sqrt{Hz}$ )	180	180	12-30	3-12	0,12-0,3
Beschl. Sensoren	Bias Stabilität ( $\mu g$ )	2400	1200	200-500	50-100	5-10
	Skalierungs-faktor (PPM)	-	>1.000	400-1.000	100-200	10-20
	Geschwindigkeit Random Walk ( $\mu g/\sqrt{Hz}$ )	1.000	1.000	200-400	50	5-10

Tabelle 2.2: IMU Klassifizierung anhand der Fehlerparameter [Edw13]

<sup>1</sup>Random Walk ist definiert als ein stochastischer Prozess, dessen Schritte zufällig und unkoreliert erfolgen. [Edw13]

## 2.2 Datenstrukturen

Datenstrukturen dienen der Speicherung und Organisation von Daten. Je nach Aufgabengebiet können Operationen wie das Suchen und Einfügen von Datenelementen effizient gelöst werden. Im folgenden werden Datenstrukturen vorgestellt, die in dieser Arbeit Anwendung finden.

### 2.2.1 Punktfolge

Die Punktfolge ist eine geometrische Datenstruktur, welche als eine Menge von Punkten  $P_i, i = 1, \dots, n$  definiert ist. Diese Punkte liegen gewöhnlicherweise im dreidimensionalen, kartesischen Raum. Der Begriff „Wolke“ beschreibt hierbei die unorganisierte Natur der Punktmenge und deren räumlichen Zusammenhang, welcher jedoch aus unscharfen Rändern besteht. [OGW<sup>+</sup>13]

Punktfolgen beschreiben 3D Objekte und Umgebungen, welche unter anderem durch die im vorhergehenden Kapitel beschriebenen Laserscanner abgetastet wurden. Somit finden Punktfolgen vor allem in dem Bereich der Computergraphik bei der Visualisierung und Modellierung von Objekten Anwendung.

### 2.2.2 Baumstrukturen

Die Graphentheorie ist ein wichtiger Teilbereich der Informatik. Graphen finden vor allem Verwendung bei der Modellierung von Objekten und Verbindungen zwischen diesen Objekten. Dabei werden die Objekte als Knoten und die Verbindungen als Kanten bezeichnet. Typische Anwendungsgebiete sind hierbei die Netzwerkanalyse, die Routenplanung und das Projektmanagement. [MS08]

In der Graphentheorie werden Bäume als eine spezielle Form von Graphen definiert. Bäume sind ungerichtete Graphen, die exakt einen Pfad zwischen zwei Knoten besitzen. Diese Eigenschaft impliziert, dass Bäume zusammenhängend sind und keine Zyklen beinhalten. Gewurzelte Bäume beginnen mit einem Startknoten, der als Wurzel bezeichnet wird, und haben demnach  $n-1$  Kanten, wobei  $n$  die Anzahl der Knoten definiert. Für die Benennung verschiedener Knoten im Baum ist es gebräuchlich Terme aus der Verwandschaftsbeziehung zu verwenden. Besitzt ein Knoten Nachfolger wird er als Elternknoten bezeichnet und die nachfolgenden Knoten entsprechend als Kinder. Knoten mit demselben Elternknoten sind Geschwister und Knoten ohne Kinder sind Blätter. Innere Knoten sind alle Knoten außer der Wurzel und den Blättern. Ein Teil dieser Beziehungen ist beispielhaft an einem binären Suchbaum der Höhe 3 in der Abbildung 2.5 illustriert. [MS08]

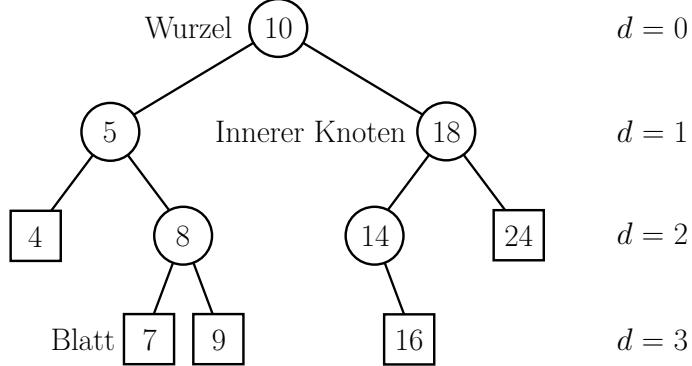
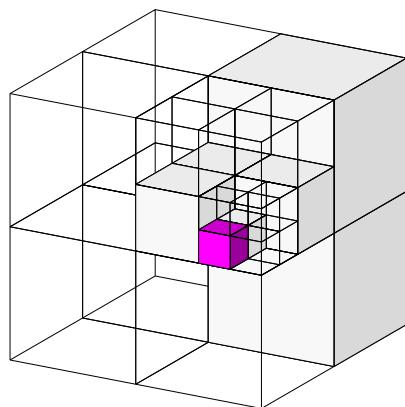


Abbildung 2.5: Binärer Suchbaum der Höhe 3

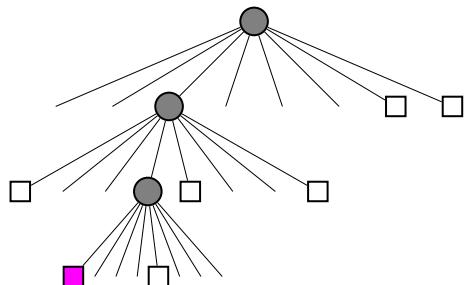
Die Höhe eines Baums wird durch die maximale Tiefe angegeben. Die Tiefe ist wiederum durch die Anzahl der Kanten bestimmt, die von der Wurzel bis zum Knoten benötigt werden. Bei einem binären Baum hat jeder Knoten höchstens zwei Kinder und bei einem Suchbaum werden die Knoten so angeordnet, dass die linken Kinder jeweils niedrigere und die rechten Kinder höhere Werte als der Elternknoten aufweisen. [MS08]

### Octree

Wie der Binärbaum gehört der Octree, eingeführt 1982 durch Donald Meagher [Mea82], zu der Klasse der  $N$ -Bäume, wobei mit  $N$  die Höchstanzahl der Kindknoten angegeben wird. Octrees werden in der Computergraphik typischerweise zur Ordnung geometrischer 3D-Daten verwendet. Die Wurzel eines Octrees repräsentiert einen Voxel mit einer bestimmten Kantenlänge im 3D-Raum. Wie in Abbildung 2.6 illustriert, unterteilen die acht Kindknoten den Raum in acht gleichgroße Voxels mit halbiertener Kantenlänge.



(a) Geometrische Darstellung



(b) Baumstruktur

Abbildung 2.6: Beispiel eines Octree, bei dem weiße Zellen als frei, Magentafarbene als belegt, und freie Zellen als unbekannt definiert sind. Nach [WHB<sup>+</sup>10]

Beim Überführen einer Punktwolke in einen Octree, wird jeder Knoten als belegt markiert, in dessen Voxel mindestens ein Punkt der Punktwolke liegt. Somit findet eine Diskretisierung der eingelesenen Geometrie statt. Durch die hierarchische Struktur des Octrees kann diese Geometrie nun in verschiedenen Auflösungen dargestellt werden, je nach dem auf welcher Ebene des Baumes die Knoten betrachtet werden (siehe Abbildung 2.7). Die Gesamtauflösung des Octrees wird dabei als die Kantenlänge der Voxel auf der untersten Ebene angegeben.

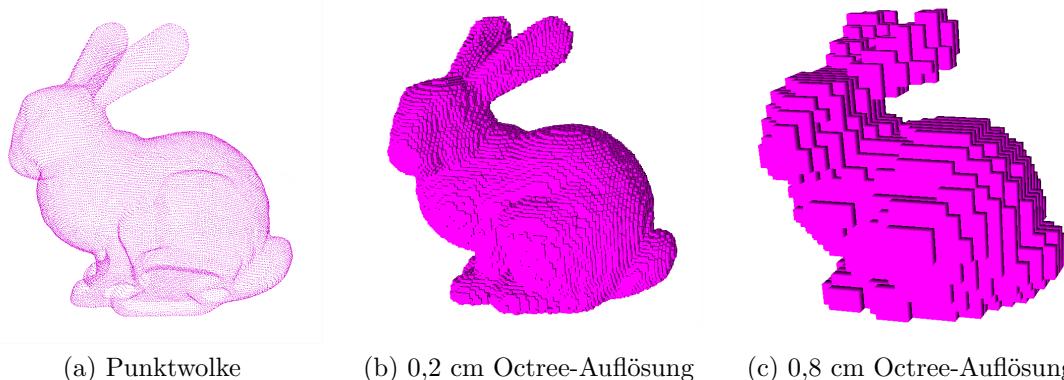


Abbildung 2.7: Darstellung verschiedener Octree Auflösungen anhand des ca. 19 cm hohen Stanford-Hasen

Beim Modellieren von Hinderniskarten ist es jedoch nicht ausreichend nur belegte Knoten zu markieren, da es bei den restlichen Knoten notwendig ist zwischen frei und unbekannt zu unterscheiden [W<sup>H</sup>B<sup>+</sup>10]. In freien Bereichen können sich die Vehikel sicher bewegen, wohingegen sie unbekanntes Terrain meiden sollten. Daher werden Strahlen von der Sensorposition zu jedem Punkt der Punktwolke gezogen, um die Knoten, die von dem Strahl getroffen werden, als frei zu kennzeichnen. Bei Octrees mit probabilistischer Belegungsfunktion [W<sup>H</sup>B<sup>+</sup>10] wird ein Wahrscheinlichkeitswert festgelegt, welcher den Zustand der Belegtheit beinhaltet. Wird ein Voxel durch aufeinanderfolgende Scandatensätzen als belegt markiert, wird die Wahrscheinlichkeit der Belegung erhöht. Dies bietet wiederum die Möglichkeit fehlerhaft belegte Voxel durch spätere Datensätze wieder freizuziehen.

### **kD-Baum**

Der *kD*-Baum ist eine weitere Datenstruktur zum Organisieren von Punktmengen in einem *k*-dimensionalen Raum. Beim *kD*-Baum wurde das Konzept des binären Suchbaums von Jon Louis Bentley im Jahr 1975 [Ben75] auf beliebig viele Dimensionen erweitert. Der Raum wird sukzessiv durch eine Hyperebene in etwa zwei gleichgroße Halbräume aufgeteilt. Dabei schneidet die Hyperebene bei jeder Teilung jeweils eine andere Dimension des Raumes, wobei die Ebene jeweils orthogonal zu der zu schnei-

denden Dimension ausgerichtet ist. Wenn die erste Teilung durch die  $x$ -Dimension durchgeführt werden soll, wird zunächst der Median der  $x$ -Koordinaten aller Punkte berechnet, um an dieser Stelle den Raum zu trennen. Anschließend werden die Punkte entsprechend ihrer  $x$ -Koordinate in die beiden Teilbäume aufgeteilt. Besitzt ein Punkt einen niedrigeren oder gleichwertigen  $x$ -Wert wie die Hyperebene, wird der Punkt in den linken Teilbaum, andernfalls in den rechten Teilbaum eingeordnet. Der Vorteil bei der Nutzung des Medians ist die Tatsache, dass der Baum sehr ausbalanciert ist, da die Anzahl der Punkte in jedem Teilbaum höchstens um einen Punkt unterschiedlich ist. Nach Abschluss der Aufteilung befinden sich die Punkte in den Blättern des Baumes, während die Wurzel und die inneren Knoten die Teilung des Suchraums repräsentieren. Abbildung 2.8 veranschaulicht das Verfahren im  $\mathbb{R}^2$  mit zehn Punkten und der alternierenden Aufteilung des Raums in der  $x$ - und  $y$ -Dimension.

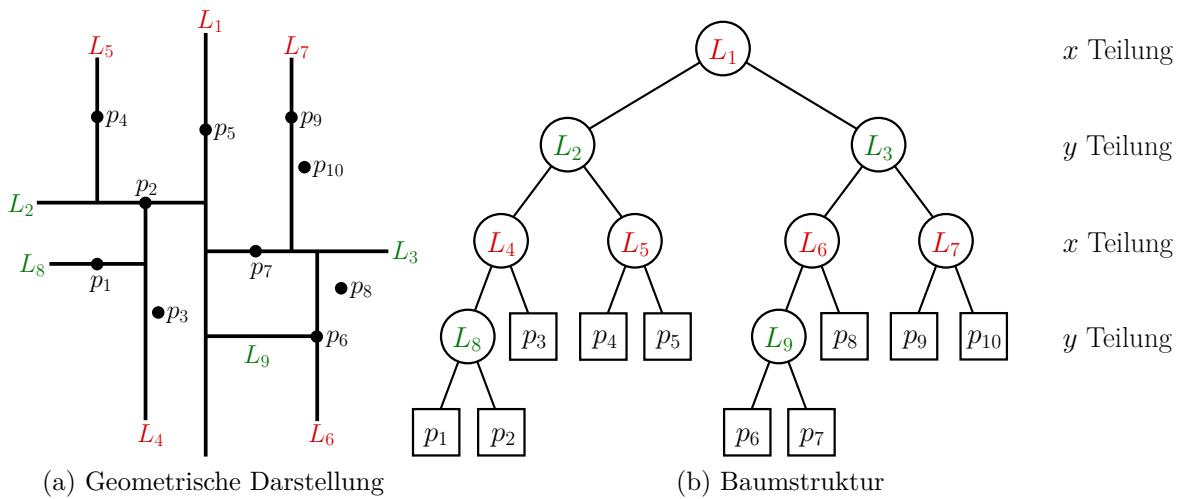


Abbildung 2.8: Beispiel eines Kd-Baums. Nach [BCKO08]

## 2.3 Mathematische Grundlagen

In diesem Abschnitt werden mathematische Grundlagen beschrieben, die vor allem für die Hauptkomponentenanalyse (PCA) benötigt werden. Dieses Verfahren zur statistischen Analyse von Daten wird anschließend beschrieben und an mehreren Stellen in dieser Arbeit verwendet. Die nachfolgenden Erläuterungen basieren auf der Arbeit von Smith [Smi02]. Vektoren  $\vec{v}$  bestehen in diesem Abschnitt sowie in der kompletten Arbeit aus Kleinbuchstaben, die mit einem Pfeil markiert sind. Matrizen  $M$  sind als fettgedruckte Großbuchstaben gekennzeichnet.

### 2.3.1 Statistik

Der statistische Bereich der Mathematik beschäftigt sich hauptsächlich mit der Analyse von großen Datensätzen. Primär sollen durch bestimmte Methoden Beziehungen in den Daten ausfindig gemacht werden. Im Folgenden wird mit der Varianz und Kovarianz ein kleines Feld der Statistik angeschnitten.

Bei einer Reihe von eindimensionalen Messwerten  $x_1, \dots, x_n$  eines Datensatz  $X$  ist es möglich die empirische Varianz zu berechnen, welche die durchschnittliche quadratische Abweichung vom Mittelwert darstellt. Positive und negative Abweichungen tragen durch das Quadrieren zu gleichen Teilen zur Varianz bei. Das arithmetische Mittel  $\bar{x}$  ist dabei durch folgende Formel gegeben:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.1)$$

Anschließend kann die Varianz  $s^2$  bestimmt werden:

$$s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (2.2)$$

Ein weiteres Maß für die Streuung ist die Standardabweichung  $s$ , die sich aus der Wurzel der Varianz ergibt.

$$s = \sqrt{s^2} \quad (2.3)$$

Handelt es sich nun um bivariate Messwerte  $(x_i, y_i)$  kann durch die statistische Analyse geprüft werden in welcher Beziehung die beiden Dimensionen zueinander stehen. Da die Varianz nur die Abweichung der beiden Dimensionen unabhängig voneinander bestimmen kann, wird zu diesem Zweck die sogenannte empirische Kovarianz berechnet:

$$\text{cov}(X, Y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad (2.4)$$

Zur einfacheren Berechnung kann der Verschiebungssatz von Steiner (Formel 2.5) verwendet werden. Diese Methode ist theoretisch exakt. Die Nutzung von float-Werten für die Umsetzung steigert jedoch die Effizienz auf Kosten der Genauigkeit.

$$\text{cov}(X, Y) = \frac{1}{n} \sum_{i=1}^n x_i y_i - \bar{x} \bar{y} \quad (2.5)$$

Erhält man beim Bestimmen der Kovarianz einen positiven Wert, steigen die Dimensionen gemeinsam an. Bei negativen Werten steigt eine Dimension an während die andere abnimmt. Bei einer Kovarianz von 0 sind die beiden Dimensionen unabhängig voneinander.

Bei Datensätzen mit mehr als zwei Dimensionen, beispielsweise die  $x$ -,  $y$ - und  $z$ -Koordinaten in  $\mathbb{R}^3$ , kann die Kovarianz jeweils für alle Kombinationen paarweise bestimmt. Alle resultierenden Kovarianzen können nun in einer  $m \times m$  großen Kovarianzmatrix  $\mathbf{C}$  gespeichert werden, wobei  $m$  für die Anzahl der Dimensionen steht.

$$\mathbf{C}^{m \times m} = (c_{ij}, c_{ij} = \text{cov}(\text{Dimension}_i, \text{Dimension}_j)) \quad (2.6)$$

Demnach würde eine Kovarianz mit den drei Koordinaten in  $\mathbb{R}^3$  folgendermaßen konstruiert werden:

$$\mathbf{C} = \begin{pmatrix} \text{cov}(X, X) & \text{cov}(X, Y) & \text{cov}(X, Z) \\ \text{cov}(Y, X) & \text{cov}(Y, Y) & \text{cov}(Y, Z) \\ \text{cov}(Z, X) & \text{cov}(Z, Y) & \text{cov}(Z, Z) \end{pmatrix} \quad (2.7)$$

Hieran lässt sich verdeutlichen, dass eine Kovarianzmatrix immer symmetrisch entlang der Hauptdiagonalen ist. Dies lässt sich auf die Symmetrie der Kovarianz zurückführen, da aufgrund des Kommutativgesetzes  $\text{cov}(X, Y) = \text{cov}(Y, X)$  gilt. Außerdem befindet sich auf der Diagonalen der Kovarianzmatrix jeweils die Kovarianz einer Dimension mit sich selbst. In diesem Fall entspricht die Kovarianz der Varianz, da die Kovarianz nur eine Verallgemeinerung der Varianz darstellt.

### 2.3.2 Matrix Algebra

Grundsätzlich kann ein Vektor  $(x, y)^T$  im zweidimensionalen Raum als ein Pfeil angesehen werden, der vom Ursprung  $(0, 0)$  zu dem Punkt  $(x, y)$  zeigt. Dieser Vektor kann nun durch eine Multiplikation mit einer Transformationsmatrix zu einem anderen Punkt zeigen. Ist der resultierende Vektor unverändert oder ein Vielfaches des

ursprünglichen Vektors, ist dieser Vektor ein Eigenvektor der Transformationsmatrix und der Multiplikationsfaktor der dazugehörige Eigenwert.

Im folgenden wird anhand eines Beispiels die Berechnung der Eigenwerte und Eigenvektoren gezeigt. Die Matrix  $\mathbf{S} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$  ist eine Spiegelmatrix im zweidimensionalen Raum entlang der  $y=x$  Diagonalen (siehe Abbildung 2.9). Nun werden zunächst die Eigenwerte  $\lambda_i$  berechnet. Dafür wird die Determinante einer Matrix benötigt, die sich wie folgt ergibt:

$$\mathbf{M} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \longrightarrow \det(M) = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc \quad (2.8)$$

Nun wird das charakteristische Polynom zu der Matrix  $S$  gebildet, wofür die Einheitsmatrix  $I$  benötigt wird.

$$\det(\mathbf{S} - \lambda \mathbf{I}) = \begin{vmatrix} 0 - \lambda & 1 \\ 1 & 0 - \lambda \end{vmatrix} = -\lambda \cdot -\lambda - 1 \cdot 1 = \lambda^2 - 1 \quad (2.9)$$

Die Eigenwerte ergeben sich nun durch das Nullstellen des berechneten Polynoms. Im Falle einer quadratischen Gleichung können diese Stellen mit Hilfe der  $pq$ -Formel berechnet werden.

$$\lambda_{1/2} = \frac{0}{2} \pm \sqrt{\left(\frac{0}{2}\right)^2 + 1} \Rightarrow \lambda_1 = 1 \text{ und } \lambda_2 = -1 \quad (2.10)$$

Anschließend lassen sich die Eigenvektoren  $\vec{x}_i$  zu diesen Eigenwerten bestimmen indem die folgende Gleichung gelöst wird.

$$\mathbf{S} \cdot \vec{x}_i = \lambda_i \cdot \vec{x}_i \Rightarrow (\mathbf{S} - \lambda_i \cdot \mathbf{I}) \vec{x}_i = 0 \quad (2.11)$$

Also bei diesem Beispiel für  $\lambda_1$ .

$$\begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (2.12)$$

Daraus ergibt sich ein Gleichungssystem mit zwei äquivalenten Gleichungen aus denen dann der Eigenvektor bestimmt werden kann.

$$\begin{aligned} -x_1 + y_1 &= 0 \\ x_1 - y_1 &= 0 \end{aligned} \Rightarrow x_1 = y_1 \quad (2.13)$$

Somit ist der erste Eigenvektor gefunden, wenn die  $x$ - der  $y$ -Koordinate entspricht. Da unendlich viele dieser Vektoren vorhanden sind, wird standardmäßig in diesem

Fall der Vektor  $\vec{x}_1 = (1/\sqrt{2}, 1/\sqrt{2})^T$  als Eigenvektor festgelegt, da er die normierte Einheitslänge besitzt. Alle Vektoren, die ein Vielfaches von  $\vec{x}_1$  sind, werden nicht als eigenständige Eigenvektoren bezeichnet, da sie linear abhängig sind.

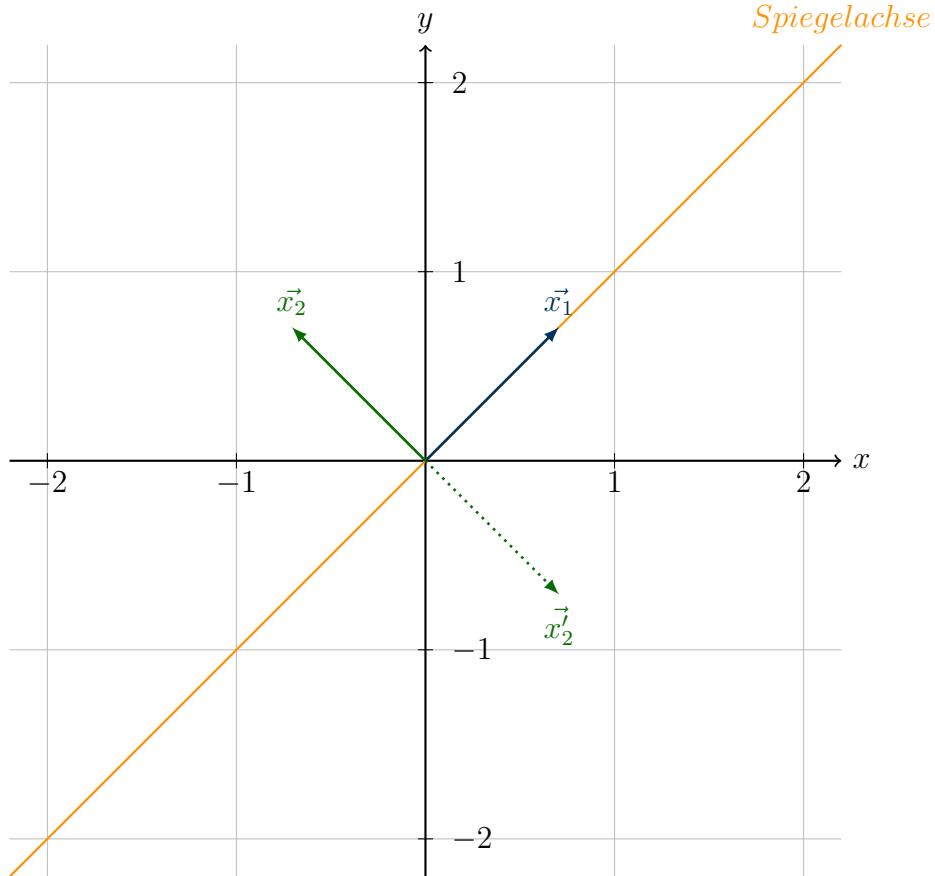


Abbildung 2.9: Darstellung der  $y=x$  Spiegelachse und den Eigenvektoren der Spiegelmatrix

Dieselben Berechnungen ergeben für  $\lambda_2$  den Eigenvektor  $\vec{x}_2 = (1/\sqrt{2}, -1/\sqrt{2})^T$ . Nun lassen sich diese Vektoren gemeinsam mit der Spiegelachse von  $\mathbf{S}$  in Abbildung 2.9 darstellen. Hierbei fällt auf, dass der erste Eigenvektor  $\vec{x}_1$  auf der Spiegelachse liegt. Demnach ist er von der Spiegelung nicht betroffen. Dies bestätigt die Aussagen über Eigenvektoren, die am Anfang dieses Abschnitts beschrieben wurde, dass der Eigenvektor nach der Transformation immernoch in dieselbe Richtung zeigt. Nach der Spiegelung von  $\vec{x}_2$  gilt dasselbe, da  $\vec{x}_2^l$  ein negatives Vielfaches von  $\vec{x}_2$  darstellt. Außerdem fällt auf, dass die beiden Eigenvektoren orthogonal zueinander stehen, was durch das Skalarprodukt der beiden Vektoren geprüft werden kann. Diese orthogonale Basis ist eine der Eigenschaften von Eigenvektoren. Weiterhin können Eigenvektoren nur für  $n \times n$  Matrizen berechnet werden, wobei die maximale Anzahl an linear unabhängigen Eigenvektoren  $n$  beträgt. Jedoch hat nicht jede Matrix Eigenvektoren. Vektoren, die beispielsweise durch eine Rotationsmatrix um  $30^\circ$  in

$\mathbb{R}^2$  gedreht werden, zeigen in keinem Fall in dieselbe Richtung wie vor der Transformation. Eine Ausnahme hierbei ist der Nullvektor, welcher nicht als Eigenvektor gewertet wird.

### 2.3.3 Hauptkomponentenanalyse

Bei zwei- und dreidimensionalen Daten ist es möglich statistische Muster in den Datenmengen visuell zu ermitteln. Da die Visualisierung von höherdimensionalen Daten schwer zu realisieren ist, werden andere Verfahren benötigt um Gemeinsamkeiten und Unterschiede identifizieren zu können. Eines dieser Verfahren ist die Hauptkomponentenanalyse. Da diese Analyse den Einfluss der einzelnen Dimensionen bestimmen kann, wird eine Dimensionsreduzierung mit geringem Informationsverlust ermöglicht. Folgend werden die Schritte der Hauptkomponentenanalyse anhand eines zweidimensionalen Beispieldatensatzes (siehe Tabelle 2.3) erläutert. Dies bietet den Vorteil, dass die Ergebnisse der PCA veranschaulicht werden können.

X	Y
-2	-3
1	2
2	1
0	0
-2	-1
2	3
-1	-2

Tabelle 2.3: Zweidimensionaler Beispieldatensatz für die PCA

Werden die  $x$ - und  $y$ -Koordinaten einzeln betrachtet, so fällt auf, dass der Mittelwert jeweils null ist. Dies ist notwendig, damit alle Variablen gleich gewichtet werden und keine Verzerrung vorhanden ist. Auf die meisten Datensätze trifft dies jedoch nicht zu. Daher wird zunächst mit Formel 2.1 der Mittelwert für beide Dimensionen berechnet. Anschließend wird von jedem Wert der Mittelwert subtrahiert, sodass in beiden Dimensionen der Mittelwert null wird. Nach der Normalisierung des Datensatzes wird die  $2 \times 2$  Kovarianzmatrix konstruiert (siehe Formel 2.6).

$$\mathbf{C} = \begin{pmatrix} \text{cov}(X, X) & \text{cov}(X, Y) \\ \text{cov}(Y, X) & \text{cov}(Y, Y) \end{pmatrix} = \begin{pmatrix} \frac{18}{7} & \frac{20}{7} \\ \frac{20}{7} & 4 \end{pmatrix} \quad (2.14)$$

Da die nicht-diagonalen Einträge der Kovarianzmatrix positiv sind, steigen die  $x$ - und  $y$ -Koordinaten, wie in Abschnitt 2.3.1 beschrieben, gemeinsam an. Nun werden die Eigenwerte und Eigenvektoren der Kovarianzmatrix berechnet.

$$\lambda_1 = 6,23 \quad \lambda_2 = 0,34 \quad (2.15)$$

$$\vec{x}_1 = \begin{pmatrix} 0,615 \\ 0,788 \end{pmatrix} \quad \vec{x}_2 = \begin{pmatrix} -0,788 \\ 0,615 \end{pmatrix}$$

Aus den Eigenvektoren wird eine neue orthogonale Basis für den Datensatz gebildet, wobei die Eigenvektoren die Hauptkomponenten widerspiegeln. Die Eigenwerte der jeweiligen Eigenvektoren geben dabei an wie groß der Einfluss des Eigenvektors bzw. der Komponente ist. Demnach werden die Eigenwerte der Größe nach sortiert.

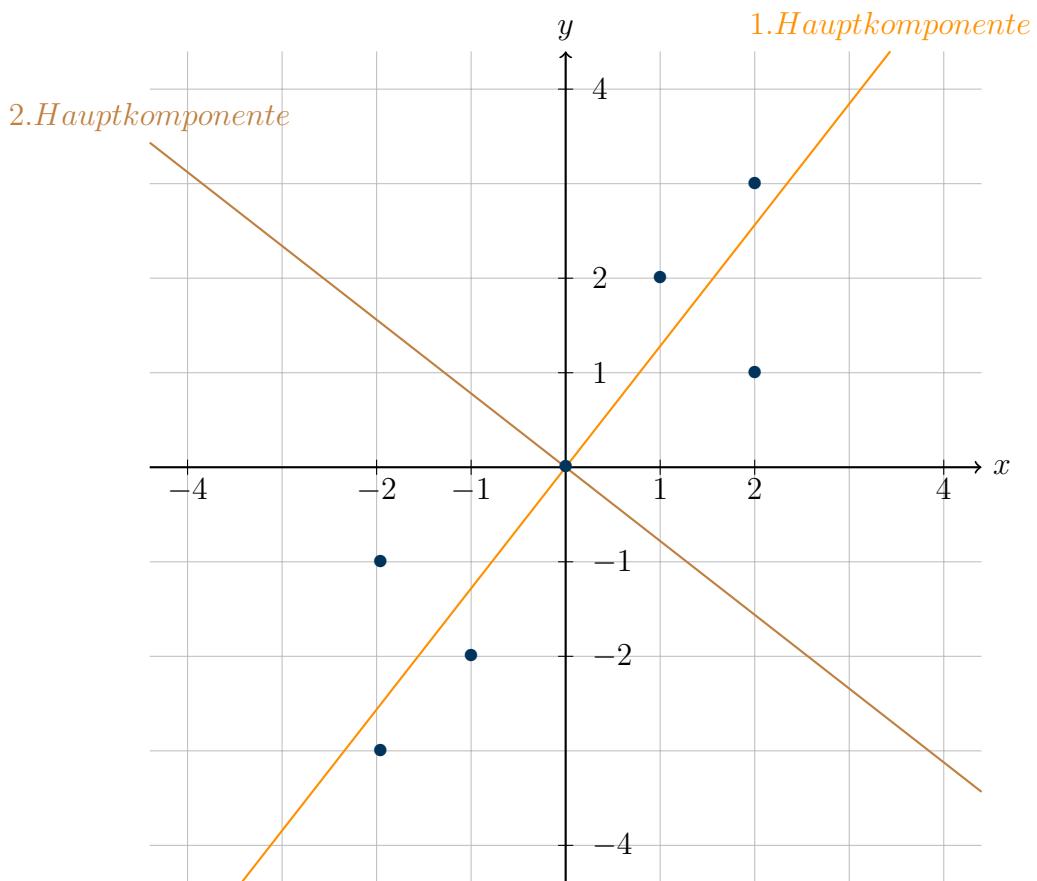


Abbildung 2.10: Darstellung der Beispieldaten und dessen Hauptkomponenten

In Abbildung 2.10 wird der Datensatz visualisiert und die Beziehung zwischen den Daten ist deutlich erkennbar, da wie erwartet, beide Koordinaten gemeinsam ansteigen. Die erste Hauptkomponente, respektive der erste Eigenvektor, besitzt die stärkste Varianz und gibt damit auch das signifikanteste Muster in den Daten an. Außerdem ist der Vektor identisch mit der Ausgleichsgeraden. Die zweite Hauptkomponente stellt ein weit weniger wichtiges Muster dar, was sich auch in der großen Differenz von  $\lambda_2$  zu  $\lambda_1$  zeigt. Das Muster zeigt die Varianz der Punkte gegenüber der ersten Hauptkomponente.

## 2.4 Geometrische Grundlagen

Die erfassten Entfernungswerte des Laserscanners sollen zur Erstellung einer globalen Hinderniskarte dienen, damit sich der Sensorträger kollisionsfrei bewegen kann. Für die richtige Verortung der Daten muss die Position und Orientierung des Sensors im globalen Raum bekannt sein. Hierfür ist eine Transformation zwischen drei rechtshändigen, kartesischen Koordinatensystemen notwendig, die in Abbildung 2.11 dargestellt sind.

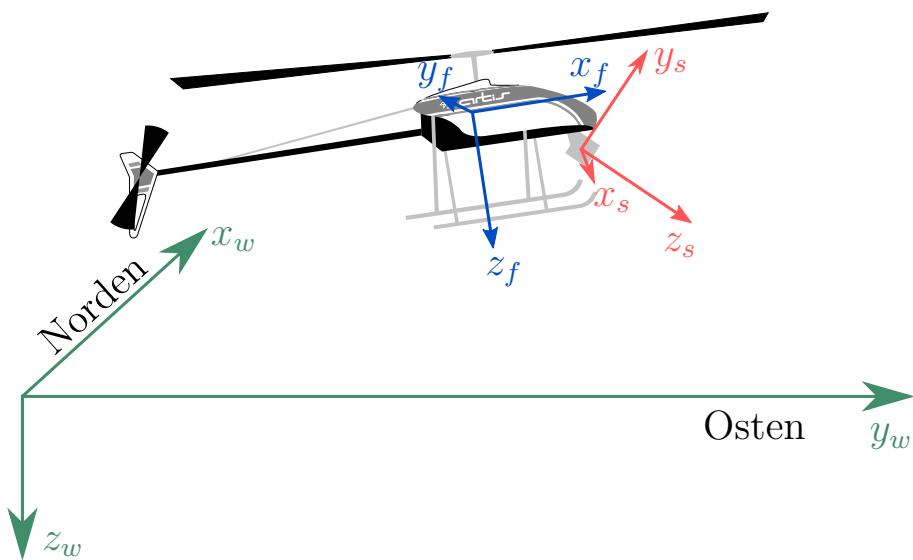


Abbildung 2.11: Darstellung der verwendeten Koordinatensysteme. Nach [And11]

Das geodätische Weltkoordinatensystem (Index  $g$ ) ist nach Andert [And11] als NED-System definiert und hat einen Referenzpunkt auf der Erdoberfläche. Dementsprechend ist die  $x$ -Achse nach Norden und die  $y$ -Achse nach Osten ausgerichtet, wodurch das Koordinatensystem tangential zur Erdoberfläche liegt. Außerdem zeigt die  $z$ -Achse in Richtung der Schwerkraft.

Der verwendete Kleinhubschrauber verwendet ein trägerfestes Koordinatensystem (Index  $f$ ), dessen Bezugspunkt als Schwerpunkt des Hubschraubers definiert ist. Die  $x$ -Achse dieses Systems ist nach vorne, die  $y$ -Achse nach rechts und die  $z$ -Achse nach unten ausgerichtet. Dabei wird die aktuelle Position und Lage den Luftfahrtkonventionen<sup>2</sup> entsprechend angegeben. Die Position wird demnach im geodätischen Koordinatensystem durch den Positionsvektor  $\vec{t}_{fg} = (x_g, y_g, z_g)^T$  und die Lage durch die Eulerwinkel  $\Phi$  (Rolllage),  $\Theta$  (Nicklage) und  $\Psi$  (Gierwinkel) angegeben. Dabei werden die Rotationsachsen mitgedreht und folgende Reihenfolge eingehalten:  $\Psi \rightarrow \Theta \rightarrow \Phi$ . Das Sensorkoordinatensystem (Index  $s$ ) beschreibt die Position und Ausrichtung des Laserscanners. Hierbei zeigt, wie in der Computergrafik üblich, die  $z$ -Achse in

<sup>2</sup>siehe Luftfahrtregel 9300, Teil 1, S. 13 (Definitionen) und S. 29 (Koordinatentransformationen)

Blickrichtung des Sensors. Die  $x$ -Achse ist nach links und die  $y$ -Achse nach oben ausgerichtet. Der Laserscanner erfasst die Daten in Kugelkoordinaten, die dann in das Sensorkoordinatensystem überführt werden müssen. Der folgende Abschnitt beschäftigt sich mit dieser Transformation.

### 2.4.1 Herleitung der Sensorkoordinaten

Da der Laserscanner schrittweise radial Laserstrahlen aussendet und somit die Distanz misst (siehe Abschnitt 2.1.1), wird jeder Messpunkt in Kugelkoordinaten durch die Distanz  $d$  und den horizontalen  $\phi$  sowie vertikalen Austrittswinkel  $\theta$  beschrieben.

$$p_p : \begin{pmatrix} d \\ \phi \\ \theta \end{pmatrix}_{\text{kugel}} \quad \text{mit } d \in \mathbb{R}_0^+, \phi \in \mathbb{R}_{-180^\circ}^{180^\circ}, \theta \in \mathbb{R}_{-90^\circ}^{90^\circ} \quad (2.16)$$

Wie in Abbildung 2.12 illustriert wird, ist  $\theta$  definiert als der Winkel zwischen dem Punkt  $p$  und der  $x_s z_s$ -Ebene des kartesischen Sensorkoordinatensystems.  $\phi$  hingegen bezeichnet den Winkel zwischen der positiven  $z_s$ -Achse und der Projektion von  $p$  auf die  $x_s z_s$ -Ebene.

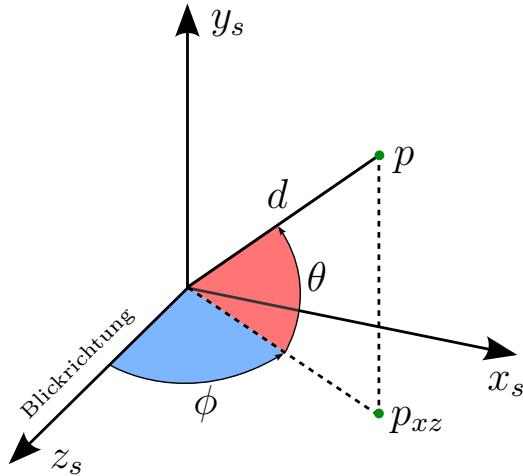


Abbildung 2.12: Umrechnung zwischen Kugelkoordinaten und kartesischen Koordinaten

Die Transformation in das Sensorkoordinatensystem kann nun wie folgt bestimmt werden:

$$p_s : \begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix} = \begin{pmatrix} d \cdot \cos(\theta) \cdot \sin(\phi) \\ d \cdot \sin(\theta) \\ d \cdot \cos(\theta) \cdot \cos(\phi) \end{pmatrix} \in \mathbb{R}^3 \quad (2.17)$$

### 2.4.2 Koordinatentransformation

Mittels der  $3 \times 3$  Rotationsmatrix  $\mathbf{R}_{ba}$  und des Translationsvektors  $\vec{t}_{ba} : (x_{ba}, y_{ba}, z_{ba})^T$  lässt sich, nach Andert [And11], die Transformation eines Punktes  $p_a$  aus dem Koordinatensystem  $a$  in das Koordinatensystem  $b$  im dreidimensionalem Raum beschreiben. Zuerst wird der Ursprung von  $a$  durch  $\vec{t}_{ba}$  auf den Ursprung von  $b$  verschoben. Anschließend findet eine passive Rotation durch  $\mathbf{R}_{ba}$  statt, so dass die Achsen von  $a$  denen von  $b$  angeglichen werden. Somit gilt für die Transformation

$$p_b = \mathbf{R}_{ba} \cdot (p_a - \vec{t}_{ba}) \quad (2.18)$$

Da Rotationsmatrizen orthogonal sind und daher  $\mathbf{R}^{-1} = \mathbf{R}^T$  gilt, ist die Rücktransformation durch

$$p_a = \mathbf{R}_{ba}^T \cdot p_b + \vec{t}_{ba} \quad (2.19)$$

gegeben.

Die Rotationsmatrix ergibt sich aus den vorher definierten Eulerwinkeln. Die Reihenfolge ist dabei strikt einzuhalten, da die Multiplikation von Matrizen nicht kommutativ ist.

$$\mathbf{R}_{ba} = \mathbf{R}_x(\Phi) \cdot \mathbf{R}_y(\Theta) \cdot \mathbf{R}_z(\Psi) \quad (2.20)$$

$$\mathbf{R}_x(\Phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\Phi) & \sin(\Phi) \\ 0 & -\sin(\Phi) & \cos(\Phi) \end{pmatrix}, \mathbf{R}_y(\Theta) = \begin{pmatrix} \cos(\Theta) & 0 & -\sin(\Theta) \\ 0 & 1 & 0 \\ \sin(\Theta) & 0 & \cos(\Theta) \end{pmatrix},$$

$$\mathbf{R}_z(\Psi) = \begin{pmatrix} \cos(\Psi) & \sin(\Psi) & 0 \\ -\sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Alternativ können die Rotation und Translation in einer homogenen  $4 \times 4$  Matrix kombiniert werden. Auf der Basis der homogenen Matrizen können aufeinanderfolgende Transformationen mit der Denavit-Hartenberg-Transformation realisiert werden.

### 3. Analyse

Die Registrierung von 3D Punktmengen ist ein weitverbreitetes Problem im Lokalisierungs- und Mapbuilding-Bereich der mobilen Robotik. Ziel der Registrierung ist die Überführung der vom Laserscanner gelieferten Scandaten in ein gemeinsames Koordinatensystem. Um dies zu erreichen, müssen Überlappungen zwischen den einzelnen Scans identifiziert werden, was als Matching bezeichnet wird. Die Registrierung bietet mehrere Ansätze zur Erreichung dieses Ziels. Zum einen wird bei der paarweisen Registrierung jeweils der aktuelle Scan dem Vorangegangenen angepasst. Der darauffolgende Scan wird wiederum dem transformierten aktuellen Scan angeglichen. Bei dieser Variante besteht die Gefahr, dass sich Ausrichtungsfehler akkumulieren und somit die Registrierung stark verschlechtern.

Bei der inkrementellen Registrierung werden zunächst die ersten beiden Scans registriert und zu einem Metascan zusammengefügt. Alle weiteren Scans werden nun dem Metascan angeglichen, wodurch dieser erweitert wird. Da bei beiden Ansätzen die Reihenfolge der Scans strikt eingehalten wird, werden zukünftige Scandaten nicht miteinbezogen, welche die bisherige Registrierung verbessern würden.

Die simultane Registrierung (engl. Multi View Registration) hingegen macht dies, indem die Scans in einen Masterscan und Nachbarscans unterteilt werden. Anschließend wird iterativ für jeden Punkt der nächstgelegene Punkt in allen Nachbarscans gesucht und somit eine Transformation bestimmt. Diese Methode besitzt den Vorteil, dass der Registrierungsfehler gleichmäßig auf alle Scans verteilt wird. Aufgrund der finiten Anzahl von Scans ist dieser Ansatz jedoch nur in der Offline-Registrierung effektiv nutzbar. [Nü06] In der vorgegebenen Aufgabenstellung hält ein Octree die Daten der globalen Hinderniskarte und die Scandatensätze werden sukzessiv in diese Karte überführt. Dies ähnelt der inkrementellen Registrierung mit dem Octree als Metascan.

Um ein angemessenes Verfahren auszuwählen, wurde eine eingehende Literaturrecherche zum Thema echtzeitnahe Registrierung von sukzessiven Laserscans zur Erstellung von globalen Hinderniskarten durchgeführt. Dabei lassen sich die Verfahren in feine und grobe Registrierung kategorisieren. Die grobe Registrierung beschäftigt sich mit der Problematik der Angleichung von Scans, deren Ausgangspositionen sich stark unterscheidet. Dieser Schritt ist notwendig, da die meisten Verfahren der feinen Registrierung, welche für die Bestimmung der optimalen Ausrichtung zuständig sind, zu einem lokalen Minimum konvergieren. Demnach soll durch die grobe Registrierung sichergestellt werden, dass das lokale Minimum auch dem Globalen entspricht. In dieser Arbeit findet die grobe Registrierung anhand der Navigationsdaten statt. Droht jedoch ein Ausfall dieser Daten, ist es notwendig ein Verfahren zu entwickeln, welches sich mit der groben Registrierung auf Scandatenebene beschäftigt. Da die Inklusion dieser Thematik den Rahmen dieser Arbeit übersteigt und die vorhandenen Lösungen in der groben Registrierung lange Berechnungszeiten aufweisen, wird diese Problematik nicht weiter berücksichtigt.

Demnach liegt das Hauptaugenmerk dieser Arbeit auf der feinen Registrierung. Der vorherrschende Algorithmus in diesem Bereich ist der Iterative Closest Point Ansatz (**ICP**), welcher iterativ den summierten quadratischen Abstand zwischen zwei Punktwolken minimiert. Im folgenden Abschnitt werden sowohl weitere Verfahren zur feinen Registrierung als auch der **ICP** und dessen Varianten analysiert und deren Anwendbarkeit auf die Problematik dieser Arbeit überprüft. Anschließend wird in **Abschnitt 3.2** untersucht in welcher Form der Octree in der Registrierung verwendet werden kann. Des Weiteren wird eine Übersicht der themenverwandten Arbeiten gegeben und abschließend ein Verfahrensablauf für die gegebene Aufgabenstellung dargestellt.

## 3.1 Feine Registrierung

Salvi et al. [SMFF07] haben verschiedene Methoden der feinen Registrierung anhand ihrer Registrierungsstrategie, Fehlermetrik, Genauigkeit und Robustheit untersucht. Diese getesteten Methoden sind der **ICP**, die Methode von Chen und Medioni, Matching Signed Distance Fields und evolutionäre Algorithmen. Zusätzlich wurden in dieser Arbeit mit der Methode von Okatani und Deguchi und dem 3D Normal Distribution Transform Algorithmus zwei weitere Verfahren analysiert.

### Matching Signed Distance Fields

Die Matching Signed Distance Fields Methode von Masuda [Mas02] arbeitet mit Tiefenbildern. Im Gegensatz zu einer Punktwolke ist ein Tiefenbild eine 2D Matrix,

welche die Tiefeninformation in den  $x$ - und  $y$ -Koordinaten speichert. Mit Hilfe der Winkelauflösung, der Sensorposition sowie des maximalen vertikalen und horizontalen Winkels kann zwischen beiden Repräsentationsarten konvertiert werden.

Bei Masudas Methoden werden die Tiefenbilder zunächst in ein gemeinsames Koordinatensystem gebracht. In diesem Koordinatensystem werden Kontrollpunkte  $k_i$  durch uniformes Abtasten des Raumes definiert. Für jedes Tiefenbild wird nun ein Signed Distance Field bestimmt, welches aus den nächsten Punkten  $p_i$  zu  $k_i$ , der Oberflächennormalen an  $p_i$  und der vorzeichenbehafteten Distanzen von  $p_i$  zu  $k_i$  besteht. Nun werden in einem iterativen Verfahren auf diese Weise in jedem Tiefenbild Korrespondenzen zu  $k_i$  gefunden. Anhand dieser korrespondierenden Punkte werden nun die Transformationsparameter bestimmt. Nachdem Konvergenz erreicht wurde, stimmen die Signed Distance Fields überein, so denn die Registrierung erfolgreich war. Diese Methode arbeitet in dem Bereich der simultanen Registrierung, wodurch es nicht in Echtzeit arbeiten kann, da am Anfang nicht alle Scans zur Verfügung stehen. Damit erfüllt diese Methode weder die definierte Echtzeitanforderung (Anforderung 3) noch die Angleichung zwischen Octree und Punktfolke (Anforderung 2).

### Evolutionäre Algorithmen

Ein Problem vieler Algorithmen zur feinen Registrierung ist das Konvergieren zu einem lokalen Minimum. Bei der Verwendung von evolutionären Algorithmen ist dies jedoch nicht der Fall. Diese Algorithmen erstellen zufällige Lösungskandidaten, die anhand einer Fitnessfunktion bewertet werden. Anhand dieser Bewertungen werden neue Lösungskandidaten gebildet. Chow et al. [CTL04] haben einen Algorithmus entwickelt, der die sechs zu bestimmenden Bewegungsparameter (die drei Komponenten des Translationsvektors und die drei Rotationswinkel) in einem Chromosom kodiert. Um den Registrierungsfehler zu minimieren, wurde der Median (siehe [Formel 3.1](#)) als Fitnessfunktion gewählt.  $\mathbf{T}$  repräsentiert hierbei die Transformationsmatrix mit den Bewegungsparametern und  $p_i$  die Punkte in der Eingangspunktfolke  $P$  sowie  $q_j$  die Punkte in der Zielpunktfolke  $Q$ . Die Menge an Punkten in  $P$  und  $Q$  sind als  $N_p$  und  $N_q$  beschrieben. Diese Definition wird in den folgenden Abschnitten weiterhin verwendet.

$$\text{Fitness}(\mathbf{T}) = \text{Median}(E_i(\mathbf{T})) \quad \text{für } 1 \leq i \leq N_p \quad (3.1)$$

$$E_i(\mathbf{T}) = \min_j (\mathbf{T}(p_i) - q_j) \quad \text{für } 1 \leq j \leq N_q \quad (3.2)$$

Ein weiterer Vorteil der evolutionären Algorithmen ist die Robustheit gegenüber Rauschen und Ausreißern aus nicht-überlappenden Bereichen. Aufgrund des hohen Rechenaufwands haben die Autoren zwei verschiedene Beschleunigungsstrategien entwickelt. Zum einen verwendet der Algorithmus nur eine Untermenge der Eingangspunktewolke, was laut den Autoren die Genauigkeit der Registrierung bei Punktewolken mit zehntausenden von Punkten nicht signifikant verschlechtert. Zum anderen wird zur Suche von korrespondierenden Punkten ein  $k$ D-Baum verwendet (siehe [Abschnitt 2.2.2](#)). Trotz dieser Verbesserungen zeigen die Experimente mit realen Daten von Salvi et al. [[SMFF07](#)] im Jahre 2007, dass die Berechnungszeiten mit über vier Minuten für die Registrierung von zwei Punktewolken mit 1.000 Punkten sehr lang sind. Obwohl sich die technischen Möglichkeiten seit diesen Versuchen deutlich verbessert haben, ist anzunehmen, dass dieser Ansatz die zeitlichen Anforderungen (Anforderung 3) nicht erfüllen kann. Ein alternativer Ansatz wäre eine Terminierung des evolutionären Algorithmus nach einer festgesetzten Zeit. Im Rahmen dieser Arbeit wurde aufgrund der ungewissen Leistungsfähigkeit von dieser Variante Abstand genommen.

### Iterative Closest Point Algorithmus

Die weitverbreitetste und erfolgversprechendste Methode zur feinen Registrierung ist der Iterative Closest Point Algorithmus, der im Jahre 1992 von Besl und McKay [[BM92](#)] entwickelt wurde. Das Verfahren ist unabhängig von der Repräsentation des 3D-Modells. Dies kann beispielsweise ein Liniensegment, eine parametrisierte Kurve, eine implizite Oberfläche oder auch eine Punktewolke sein.

Das Hauptziel des **ICP** ist das Bestimmen einer optimalen Transformation durch das iterative Minimieren einer Distanzmetrik zwischen korrespondierenden Punkten. Dabei unterteilt sich der Algorithmus in vier Teilschritte, welche solange iterativ angewendet werden bis eine Abbruchbedingung erreicht wird (siehe Schritt 4). [Abbildung 3.1](#) illustriert die Anwendung eines **ICPs** auf eine simple Punktmenge.

1. Da die tatsächlich korrespondierenden Punkte unbekannt sind, wird im ersten Schritt für jeden Punkt  $p_i$  anhand der euklidischen Distanz<sup>1</sup> der nächste Nachbar  $q_i$  in der Zielpunktewolke gesucht und für diese Iteration als korrespondierender Punkt bestimmt. Diese Schätzung schließt nicht aus, dass tatsächlich korrespondierende Paare gefunden werden, wie in [Abbildung 3.1](#) exemplarisch verdeutlicht wird.

---

<sup>1</sup>Formel der Euklidischen Distanz:  $d(x, y) = \|x - y\|^2 = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$ .

2. Mit Hilfe der Fehlermetrik wird die Transformation berechnet, welche die in Schritt 1 gefundenen Punktpaare am besten zusammenführt. Die meistverwendete Fehlermetrik ist die Summe der quadrierten Distanzen der korrespondierenden Punktpaare, welche auch als Point-to-Point Metrik bezeichnet wird.
3. Anwenden der Transformation auf die anzugeleichende Punktfolge
4. Beenden des Algorithmus falls mindestens ein Terminierungskriterium erfüllt ist. Dabei sind drei gebräuchliche Terminierungskriterien vorhanden. Das Erste tritt ein, falls der Wert der Fehlermetrik unter einen spezifizierten Schwellenwert fällt, der die gewünschte Präzision der Registrierung festlegt. Außerdem wird der **ICP** beendet, wenn zwischen der momentanen und vorangegangenen Transformation keine große Verbesserung erzielt werden konnte. Dies wird ebenfalls über einen Schwellenwert bestimmt. Zuletzt dient eine Höchstzahl an Iterationen als ultimatives Terminierungskriterium.

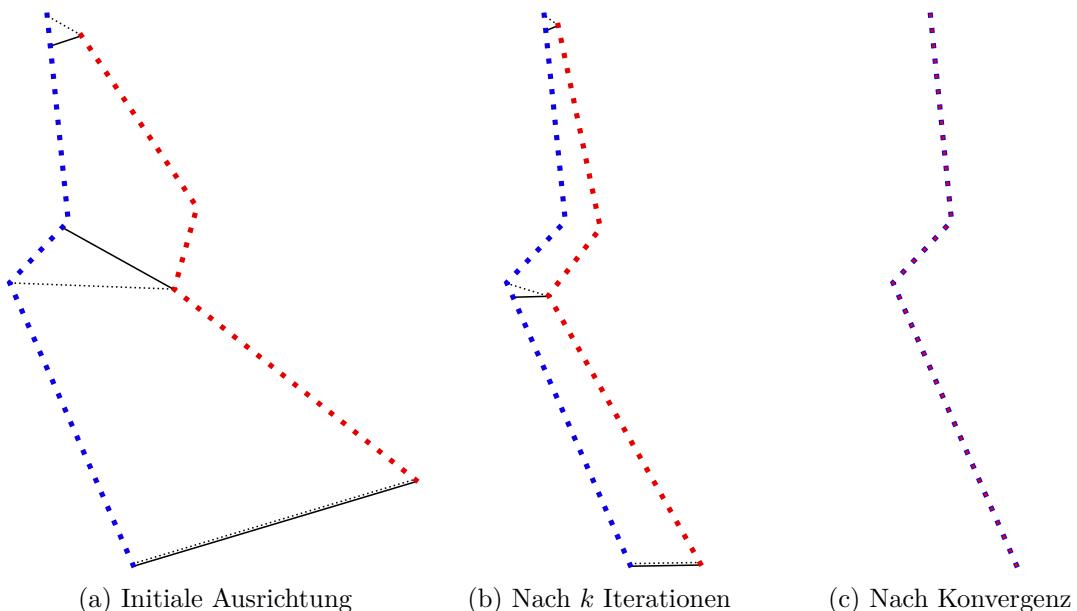


Abbildung 3.1: Angleichung einer Eingangspunktwolke (rot) an eine Zielpunktwolke (blau) durch den **ICP**. Bei ausgewählten Punkten der Eingangspunktwolke sind die nächsten Nachbarn mit einer Linie und die korrekten Korrespondenzen mit einer gepunkteten Linie hervorgehoben.

Die Registrierung mit dem **ICP** verschlechtert sich durch die Anzahl der Ausreißer sowie nicht-überlappenden Bereiche, wodurch teilweise keine zufriedenstellende Lösung bestimmt werden kann. Um diesen Umständen entgegenzuwirken, wurden verschiedene Verfahren entwickelt, die in [Kapitel 5](#) analysiert werden. Die Vorteile

des Iterative Closest Point Algorithmus sind die Unabhängigkeit von der Repräsentation des Modells (Anforderung 2) und die stabil guten Ergebnisse bei der Registrierung. Ein Nachteil des ICP ist die Berechnungszeit (Anforderung 3), vor allem bei der Bestimmung des nächsten Nachbarn. Für diese Problematik werden ebenfalls verschiedene Verbesserungen in Kapitel 5 beschrieben und auf ihre Anwendbarkeit geprüft.

### Methode von Chen und Medioni

Chen und Medioni [CM91] entwickelten ungefähr zeitgleich zu Besl eine dem ICP sehr ähnliche Methode. Im Gegensatz zu der Berechnung des nächsten Nachbarn in der Zielpunktwolke wird hierbei die Point-to-Plane Metrik verwendet. Diese berechnet die quadrierte Distanz zu einer Tangentialebene, die durch den nächstgelegenen Punkt und dessen Normalvektor bestimmt ist.

Dafür wird jedoch ebenfalls vorerst für jeden Punkt der Eingangspunktwolke  $p_i$  der nächste Nachbar  $q_i$  in der Zielpunktwolke gesucht. Nun werden für alle  $q_i$  die Tangentialebene und der Normalenvektor  $\vec{n}_i$  bestimmt. Eine nähere Erläuterung für diesen Berechnungsschritt für Punktwolken befindet sich in Abschnitt 4.1.1. Anschließend wird  $p_i$  entlang der Normalen  $\vec{n}_i$  auf die Tangentialebene von  $q_i$  projiziert. Die Summe der quadrierten Längen  $d_i^2$  aller Projektionen bildet die Fehlermetrik. In Abbildung 3.2 ist dieses Verfahren an zwei Punkten veranschaulicht. [CM91]

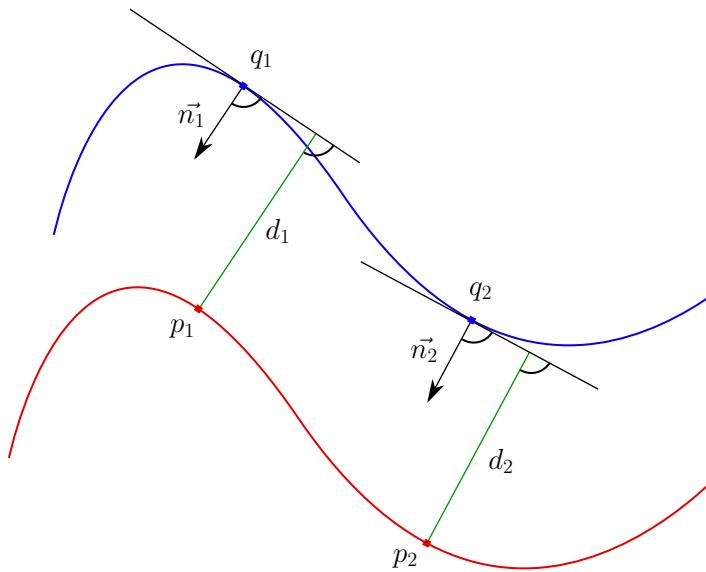


Abbildung 3.2: Beispiel der Point-to-plane Fehlermetrik

Für eine Metrik dieser Form ist jedoch zu Transformationsberechnung keine analytisch geschlossene Lösung vorhanden, so dass auf standardmäßige nichtlineare Least

Squares Methoden, wie die Levenberg-Marquardt Methode, zurückgegriffen werden muss.

Im folgenden wird diese Methode als Variante des [ICP](#) gehandelt, dessen Unterschied die verwendete Fehlermetrik darstellt.

### Methode von Okatani und Deguchi

Okatani und Deguchi [OD00] nehmen im Gegensatz zu den anderen Verfahren nicht an, dass die Scandaten fehlerfrei sind. Ihre Methode verwendet Tiefenbilder und nimmt eine Korrektur der Punkte anhand der Wahrscheinlichkeit der Messfehler bezogen auf den Standpunkt des Scanners vor. Generell werden Messfehler größer für Punkte, die weiter entfernt liegen, und für Punkte, die auf steilen Oberflächen in der Blickrichtung liegen. Außerdem zeigen sie, dass die Messfehler in Blickrichtung viel größer sind als die Messfehler in der Bildebene des Tiefenbildes. Daher wird eine gaußsche Wahrscheinlichkeitsverteilung entlang der  $z$ -Achse (Tiefeninformation) erwartet. Die Zielpunktwolke wird bei ihrem Ansatz als eine Menge von Dreiecken repräsentiert. Demnach ist es notwendig eine Triangulierung durchzuführen. Anschließend wird in einem iterativen Verfahren die wahrscheinlichste Transformation der Eingangspunktwolke anhand der vorher definierten Messfehlerverteilungen gefunden. Hierbei wird die Position von jedem gemessenen Punkt durch die Varianz und Ausdehnung der Wahrscheinlichkeitsverteilung korrigiert. Die optimale Transformation wird durch die Bewertung der Effektivität der Korrektur von allen Messpunkten ausgewählt. Demnach ähnelt diese Registrierungsmethode dem [ICP](#). Die Methode von Okatani und Deguchi findet in dieser Arbeit keine Anwendung, da die Messfehler der Scandatensätze im Vergleich zur Diskretisierung durch den Octree klein sind und damit keine großen Vorteile zum [ICP](#) vorhanden sind. Außerdem ist die Mesh-Repräsentation der Zielpunktwolke nicht mit dem Octree kompatibel. Die Information über größere Ungenauigkeiten bei steigender Distanz zum Sensor wird jedoch in [Abschnitt 4.1](#) zur Reduzierung des Scandatensatzes verwendet.

### 3D Normal Distribution Transform

Die 3D Normal Distribution Transform (3D-NDT) Methode von Magnusson et al. [MLD07] arbeitet ebenfalls mit einer anderen Repräsentation der Zielpunktwolke. Anstatt einer Octree-Repräsentation oder einer Punktewolke wird eine Kombination von Normalverteilungen genutzt, um die Wahrscheinlichkeit zu berechnen, mit der ein Punkt auf der Oberflächen der Zielpunktwolke liegt. Die Normalverteilungen bilden dabei eine stückweise-glatte Repräsentation der Oberfläche der Zielpunktwolke mit kontinuierlichen Ableitungen ersten und zweiten Grades. Durch das Nutzen dieser

Repräsentation ist es möglich numerische Optimierungsmethoden für die Registrierung zu verwenden.

Im ersten Schritt wird der Raum der Zielpunktwolke in Voxel aufgeteilt, wofür die Autoren mehrere Verfahren vorschlagen. Für jeden Voxel, der eine bestimmte Anzahl an Punkten aufweist, wird ein Durchschnittspunkt und eine Kovarianzmatrix berechnet (siehe [Abschnitt 2.3.1](#)). Anschließend wird anhand dieser beiden Werte eine Normalverteilung bestimmt, welche die Wahrscheinlichkeit eines Punkts an jeder Position in diesem Voxel ermitteln kann.

Außerdem werden die zu optimierenden Parameter, die Translation und Rotation des aktuellen Scandatensatzes, in einem Vektor gespeichert, der somit die Pose definiert. Der Algorithmus bewertet die Pose nun, indem die Wahrscheinlichkeitsdichtefunktionen an allen Punkten der Eingangspunktwolke ausgewertet werden. Da Optimierungsprobleme im Regelfall als Minimierungsproblem definiert sind, wird das Bewertungsmaß als negierte Summe der Wahrscheinlichkeiten, mit der die transformierten Punkte des Eingangsscans auf der Oberfläche der Zielpunktwolke liegen, bestimmt. Anhand der Hesse und Jacobi Matrizen berechnet der Newton Algorithmus iterativ die verbesserte Pose bis ein Schwellenwert unterschritten wird.

Ein großer Vorteil dieser Methode gegenüber dem Iterative Closest Point Ansatz ist, dass die rechnerisch komplexe Suche des nächsten Nachbarn wegfällt. Die Autoren benennen eine optimale Voxelgröße von 1 - 2 m für die Umgebung innerhalb von Bergwerken. Da bei der Modellierung der Hinderniskarte zumeist auf eine Voxelgröße von 0,5 m beim Octree zurückgegriffen wird, würden maximal acht Voxel innerhalb eines [3D-NDT](#)-Voxel für die Berechnung der Normalverteilung zur Verfügung stehen. Da die Voxel des speichereffizienten Octree jedoch nur die Information über die Belegtheit enthalten, wäre die einzige Möglichkeit entweder die Ecken oder das Zentrum der Voxel als Punkte zu definieren. Die geometrische Regelmäßigkeit dieser Punkte hat zur Folge, dass die Normalverteilungen an vielen Stellen sehr ähnlich ausfällt, was zu einer nicht-optimalen Registrierung führen könnte.

Das [3D-NDT](#) kann eine sinnvolle Alternative zu der Verwendung des Octrees als globale Hinderniskarte darstellen. Die Repräsentation ist durch die Speicherung der Normalverteilungen ebenfalls speichereffizient und verbessert möglicherweise die Registrierung eingehender Scandaten. Der Vorteil des Octrees ist jedoch die direkte visuelle Darstellung. Inwiefern die [3D-NDT](#) Methode zur Pfadplanung verwendet werden kann, muss geprüft werden. In dieser Arbeit wird daher auf diesen Ansatz verzichtet, da die Verwendung einer Kombination von Octree und [3D-NDT](#) wenig erfolgsversprechend ist. (Anforderung 2)

## Fazit

Salvi et al. [SMFF07] sind zu dem Ergebnis gekommen, dass der ICP mit der Point-to-Plane Fehlermetrik die besten Registrierungsergebnisse liefere, da sie im Gegensatz zu den anderen Methoden im Stande ist das Problem des lokalen Minimums weitestgehend zu lösen, indem die Punkte auf eine Ebene projiziert werden. Jedoch ist die Berechnungszeit deutlich höher als bei den Point-to-Point Varianten, die ebenfalls gute Ergebnisse liefern, wenn eine robuste Auswahl einer Untermenge an Punkten verwendet wird. Daher werden im Kapitel 4 verschiedene Verfahren zur Selektion von Punkten vorgestellt.

Im weiteren Verlauf dieser Arbeit werden der ICP und dessen Varianten genauer betrachtet. Einerseits aufgrund der guten Ergebnisse dieses häufig verwendeten Ansatzes und andererseits weil die anderen Methoden entweder nicht auf das vorliegende Problem anwendbar oder die Ergebnisse unvorhersehbar sind. Eine weitere Bekräftigung besteht durch die vermehrte Nutzung des ICP im Bereich der simultanen Lokalisierung und Kartenerstellung (SLAM). Daher wird in Abschnitt 3.3 ein kurzer Überblick über diesen Bereich gegeben, wobei verschiedene Ansätze näher betrachtet und die Unterschiede zum vorliegenden Problem erläutert werden. Im folgenden Abschnitt werden die Möglichkeiten genauer betrachtet, wie der Octree im ICP am effektivsten verwendet werden kann.

## 3.2 Verwendung des Octrees zur Registrierung

Der ICP wurde unter anderem aus dem Grund ausgewählt, da er unabhängig gegenüber der Repräsentation des Zielmodells ist, an das die Punktfolge angeglichen werden soll. Da bei der vorliegenden Aufgabenstellung eine Registrierung zwischen einer Punktfolge und einem Octree stattfinden soll, werden nun die Möglichkeiten beschrieben in denen der Octree in den ICP mit eingebracht werden kann.

Vorerst wird durch die Abbildung 3.3 noch einmal die Problemstellung mit einem Quadtree abstrahiert. Die belegten Knoten sind blau markiert. Der aktuelle Scan-datensatz, der in rot dargestellt ist, soll nun so transformiert werden, dass die Messpunkte korrekt in den Quadtree überführt werden können.

Aufgrund des effizienten Aufbaus und der Speicherstruktur des Octrees ist nur bekannt, welche Voxel als belegt, frei oder unbekannt markiert sind. Demnach bieten sich drei verschiedene Repräsentationen an. Diese sind die Ebenen, die durch die Flächen der Voxel definiert sind, die Kanten der Voxel und charakteristische Punkte innerhalb oder am Rand des Voxels. Diese Punkte können beispielsweise die geometrischen Zentren oder die Eckpunkte der Voxel darstellen. Das Scanverfahren tastet

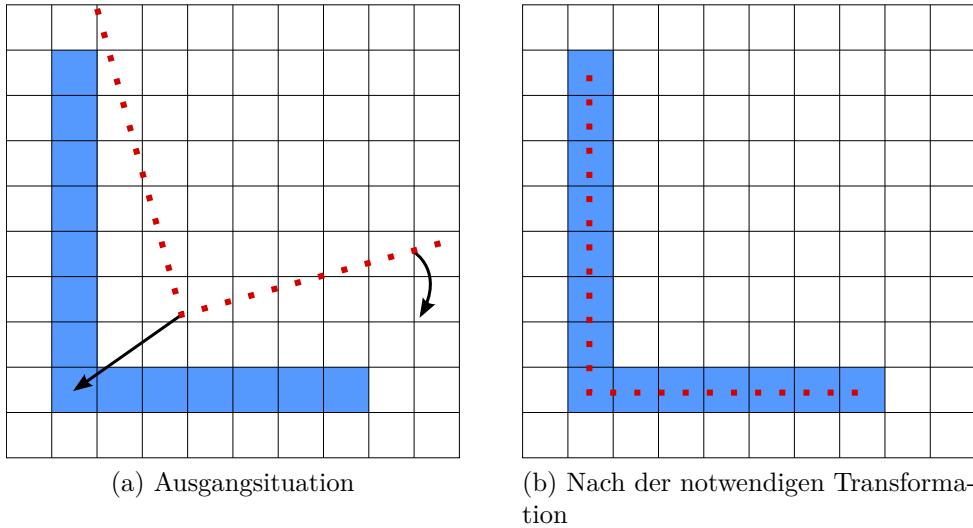


Abbildung 3.3: Beispielregistrierung eines Scandatensatz als Punktwolke (rot) mit der globalen Hinderniskarte als Octree in 2D. Belegte Knoten sind blau markiert.

die Umgebung ab, indem die Distanz zu den Rändern von Objekten gemessen wird. Daher scheint eine Verwendung einer Repräsentation, die den Rand des Octrees abbildet, sinnvoll. Der Octree führt jedoch eine Diskretisierung ein, was die Wahrscheinlichkeit erhöht, dass der Rand der gemessenen Objekte innerhalb der Voxel liegt. Da die hier vorgestellten Repräsentationen bis auf das Zentrum des Voxels, den Rand des Octrees darstellen, wird in dieser Arbeit eine Punktwolke mit allen Voxelzentren verwendet. Diese Repräsentation hat außerdem den Vorteil, dass Fehler, die unterhalb der halben Octreeauflösung liegen, zu keiner schlechteren Registrierung führen.

### 3.3 Themenverwandte Arbeiten

Themenverwandte Arbeiten fallen zu einem großen Teil in die Thematik **ICP-SLAM**, wobei hier vorrangig die Odometriedaten von bodengebundenen Sensorträgern zu korrigieren sind.

Bei der Odometrie werden Signalgeber an den Rädern verwendet, um die zurückgelegte Strecke zu messen. Fehler entstehen wenn Räder unrund sind, rutschen (Schlupf) oder durch den Reifenluftdruck verschiedene Durchmesser haben. Während die Odometrie auf kurzen Distanzen noch gute Ergebnisse liefert, wachsen jedoch die Fehler mit zunehmender Entfernung. [DP00]

Außerdem besteht ein Unterschied in der Anzahl der verwendeten Scandatensätze. Im Gegensatz zum kontinuierlichen Scannen wird beim **SLAM** in den meisten Fällen der Sensorträger über mehrere Meter bewegt, bevor ein weiterer Scan akquiriert

wird. Aufgrund dieser Differenzen greifen diese Arbeiten oft auf Ansätze der groben Registrierung zurück, ehe der ICP zur Feinjustierung verwendet wird. Nach der Registrierung wird überprüft ob ein sogenanntes Loop Closing vorliegt. Das Loop Closing beschreibt die Situation, wenn der Roboter eine Position erreicht an der er bereits vorher einen Scandatensatz aufgenommen hat. Mit Hilfe dieser Information kann der Registrierungsfehler zwischen diesen beiden Scans auf alle dazwischenliegenden Datensätze verteilt werden.

Beim 6D-SLAM-Ansatz von Nuechter et al. [NLHS05] wird, neben den Odometriedaten, eine Heuristik verwendet, die auf einem Vergleich von zwei Octrees basiert. Dafür werden der aktuelle und der folgende Scan in jeweils einen Octree überführt. Anschließend wird die beste Transformation auf jeder Suchtiefe des Octrees bestimmt. Die Transformation, bei der die meisten belegten Voxel überlappen, wird für die Registrierung ausgewählt. Für diesen Schritt werden sechs verschachtelte Schleifen benötigt, was zu einem der Hauptprobleme führt. Nach dem Anwenden des ICP und des Loop Closing benötigt dieser Ansatz für 77 Scans, mit ca. 100.000 Punkten, etwa 10 Minuten, was die geforderten nahen Echtzeitanforderungen dieser Arbeit nicht erfüllt. Bei der vorliegenden Aufgabenstellung besteht ebenfalls die Möglichkeit ein Loop Closing zu erkennen. Der Registrierungsfehler kann jedoch nicht auf die vorhergehende Scandatensätze verteilt werden, da diese nach der Überführung in den Octree nicht mehr zur Verfügung stehen.

Triebel et al. [TPB06] nutzen für die globale Repräsentation eine erweiterte Höhenkarte, die es ermöglichen soll Unterführungen von Brücken darzustellen. Bei dieser Karte kann jede 2D Zelle eine Liste von Oberflächen beinhalten. Diese Oberflächen werden durch den Mittelwert und die Varianz der gemessenen Höhen in dieser Zelle definiert. Zusätzlich werden die Zellen als passierbar, nicht passierbar oder vertikal klassifiziert. Bei den ersten beiden Klassen werden nun für jeden Meter vier Punkte ausgewählt. Bei vertikalen Zellen hängt die Anzahl der selektierten Punkte von ihrer Höhe ab. Diese Punkte werden nun für die Registrierung mit dem ICP verwendet. Bei einer Versuchsdurchführung wurde mittels 77 Scandatensätzen, mit insgesamt 20.207.000 Datenpunkten, eine Region von  $195 \times 146$  Metern abgedeckt. Die Autoren haben jedoch keine Berechnungszeiten für ihr Registrierungsverfahren angegeben.

Moosmann und Stiller [MS11] weichen von der Strategie ab, bei der nach einigen Metern ein neuer Scandatensatz generiert wird. Dabei findet ein kontinuierliches Scannen (10 Hz) mit einem Laserscanner statt, der dem in dieser Arbeit verwendeten Scanner (siehe Abschnitt 6.1) sehr ähnlich ist. Als globale Karte wird ein 3D Gitter verwendet, welches in jeder Zelle eine Oberfläche speichert, die durch einen Punkt, einen Normalvektor und einen Konfidenzwert, der die Flachheit der lokalen

Region angibt, definiert ist. Bei der Verwendung des **ICP** wird eine Untermenge an Oberflächen aus dem aktuellen Scandatensatz verwendet, um eine schnelle Registrierung zu ermöglichen. Der wichtigste Parameter dieses Ansatzes ist die Wahl der Gittergröße. Während ein kleiner Wert eine präzise Kartierung ermöglicht, kann durch einen hohen Wert eine schnellere Berechnung erreicht werden. Die Autoren geben ebenfalls keine Berechnungszeiten an. Da die festgelegte Gittergröße bei den durchgeföhrten Experimenten jedoch bei nur 5 cm lag, ist anzunehmen, dass auf diese Weise keine echtzeitnahe Registrierung erlangt werden kann. Dennoch ist dieser Ansatz theoretisch interessant, erfordert jedoch eine Anpassung der globalen Repräsentation.

Im Bereich der unbemannten Luftfahrzeuge werden zumeist Multicopter verwendet, die eine sehr geringe Traglast haben. Daher werden vor allem 2D Zeilenscanner an die **UAs** montiert. Winkvist [Win13] hat einen Ansatz für echtzeitfähiges **ICP-SLAM** bei einem mit einer **IMU** und Zeilenscanner ausgestatteten Hexacopter entwickelt. Eine 3D Kartierung durch einen **ICP** mit Daten von einem Zeilenscanner ist nicht praktikabel, da die Punkte der Scandatensätze übereinander geschoben und somit falsch angeglichen werden. Aus diesem Grund wurden große Veränderungen eingeführt. Die Rotation wird nicht korrigiert, da sie zu fehlerhaften Registrierungen führt und die Kalkulation für den Computer eines **UAs** aufwändig ist. Stattdessen wird die Rotation allein durch die **IMU** bestimmt. Somit berechnet diese echtzeitfähige Variante nur eine laterale Translation. Die Translation wird durch den Mittelpunkt der Scanpunkte und den Mittelpunkt der korrespondierenden Punkte in der globalen Repräsentation bestimmt. Als globale Karte dient eine Punktfolge, die durch die Scandatensätze erweitert wird, wenn das **UA** sich mehr als einen halben Meter bewegt hat. Zudem wird die Fehlerfunktion nicht bei jeder Iteration berechnet, sondern vor und nach der iterativen Phase. Suggeriert der Fehler eine inkorrekte Angleichung wird der Datensatz verworfen.

Droeschel et al. [DSB13] generieren mit einem ähnlichen Versuchsaufbau 3D Scandatensätze, indem der Zeilenscanner zusätzlich rotiert wird. Somit werden 3D Punktfolgen mit bis zu 21.600 Punkten bei einer Reichweite von 30 m erstellt. Dieser Prozess dauert 0,5 Sekunden. Als Umgebungskarte wird ein dreidimensionales Zellengitter erstellt, welches einem Octree ähnelt. Die Zellen speichern die Belegtheit und eine Liste der 3D Punkte der beiden vorhergehenden Scandatensatzes. Wird nun ein neuer Scan akquiriert, werden bei der Verwendung des **ICP** die nächsten Nachbarn anhand der gespeicherten Listen in den Zellen gesucht. Ansonsten sind bei der Registrierung keine Veränderungen vorgenommen worden. Die Experimente, welche innerhalb von Gebäuden durchgeföhrten wurden, zeigen, dass dieser Ansatz bei einer Zellgröße von 2 m eine durchschnittliche Laufzeit von 0,42 s erreicht. Dies

entspricht den Anforderungen dieser Arbeit an eine echtzeitfähige Registrierung. Es ist allerdings darauf hinzuweisen, dass eine Zellengröße von 2 m bereits eine recht grobe Auflösung darstellt. Außerdem wurden noch keine Versuche außerhalb von Gebäuden durchgeführt.

### 3.4 Ablauf des entwickelten Verfahrens

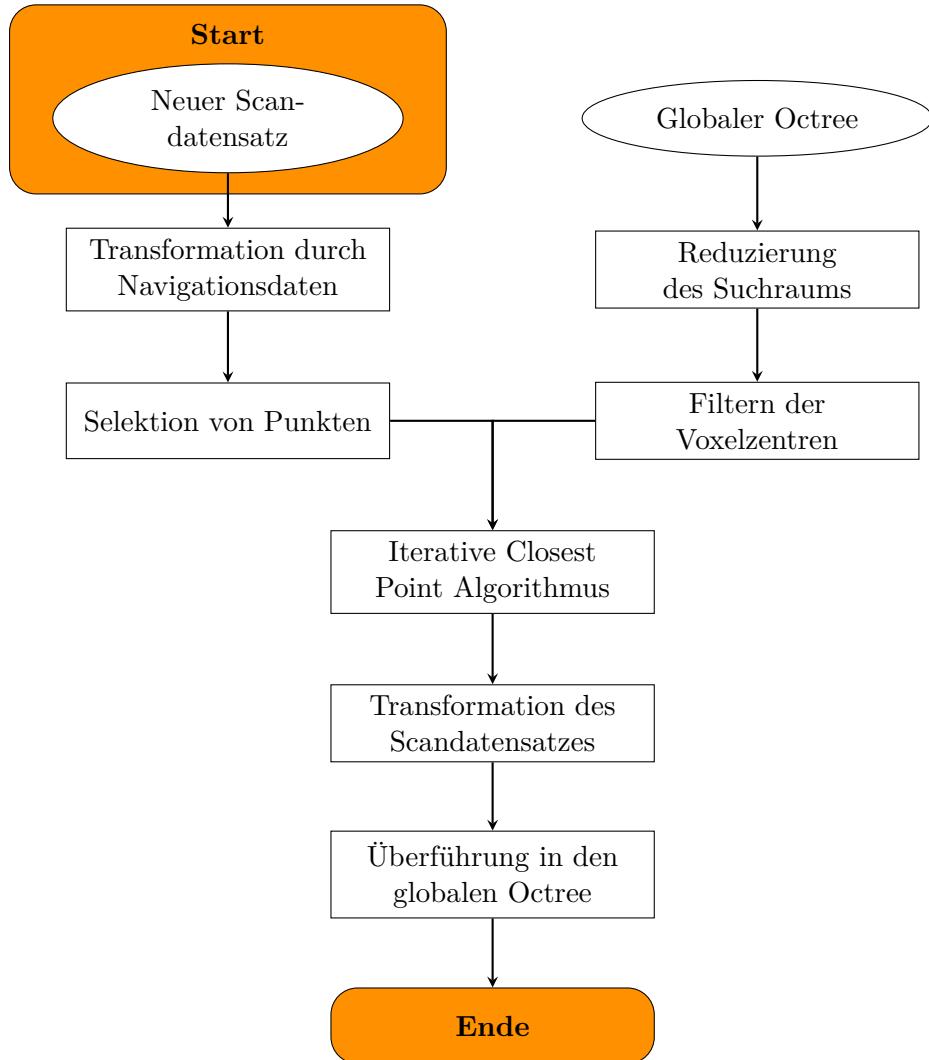


Abbildung 3.4: Ablaufdiagramm des entwickelten Ansatzes

In diesem Abschnitt wird der Ablauf des entwickelten Verfahrens für die Korrektur der Navigationsdaten durch die Registrierung beschrieben. Hierfür wurde zunächst ein Ablaufdiagramm definiert, welches die notwendigen Schritte zum Erreichen dieses Ziels beinhalten. Dieser Ablauf wird durch einen neu akquirierten Scandatensatz angestoßen. Diese Scandaten werden anhand synchronisierter Navigationsdaten in das Weltkoordinatensystem überführt (siehe [Abschnitt 2.4](#)). Um eine robuste Regis-

trierung mit dem ICP zu gewährleisten und die Rechenkomplexität zu verringern, wird eine Untermenge der Punkte aus dem Scandatensatz ausgewählt.

Zur Definierung der Zielpunktwolke werden die für die Registrierung relevanten Voxel ermittelt, so dass der Suchraum reduziert wird und nicht der gesamt Octree verwendet wird. Anschließend wird die Zielpunktwolke aus den Zentren dieser Voxel gebildet. Nun findet die Registrierung des reduzierten Scandatensatzes mit der Zielpunktwolke mittels des ICPs statt. Anhand der dabei ermittelten Translation und Rotation wird der vollständige Datensatz transformiert und in den globalen Octree überführt.

Die folgenden zwei Kapitel beschreiben nun einzelne Teile dieses Ablaufs. Kapitel 4 geht auf die Wahl der Punkte ein, die für den ICP verwendet werden und deckt somit die Reduzierung des Suchraums und die Selektion der Untermenge des Scandatensatzes ab. Kapitel 5 hingegen analysiert die verschiedenen Varianten in den einzelnen Phasen des ICP

# 4. Selektion von Punkten

Der originale ICP Algorithmus bestimmt für alle verfügbaren Punkte Korrespondenzen und nutzt diese für die Registrierung. In dieser Methodik sind mehrere Nachteile bei der Anwendung auf das vorliegende Problem vorhanden. Zum einen verhindert die Komplexität und der damit verbundene Zeitaufwand bei einer hohen Anzahl aktueller Scanpunkten sowie besetzter Voxel im Octree eine echtzeitnahe Umsetzung (Anforderung 3). Zum anderen soll eine höhere Stabilität erlangt werden, indem Punkte, die nicht positiv zur Registrierung beitragen, gefiltert werden. Im folgenden Abschnitt wird die Auswahl der Punkte des Scandatensatzes erläutert und [Abschnitt 4.2](#) beschreibt die Eingrenzung des Suchraums beim Octree.

## 4.1 Reduzierung der Punkte im Scandatensatz

Eine erste Reduzierung der Punkte findet anhand der Distanz zur Sensorposition statt, da Messungenauigkeiten größer werden umso weiter die Messpunkte entfernt sind. Es wird ermittelt wie viele Punkte eine geringere Entfernung als 20 m zum Sensor aufweisen. Liegt die Anzahl an Punkten bei über 90% aller Punkte des Scandatensatzes, wird mit diesen Punkten weiter verfahren. Andernfalls wird der Maximalabstand sukzessive um 5 m erweitert bis mehr als 90% des Datensatzes beibehalten werden. Alternativ könnten direkt die 10% weitesten Punkte entfernt werden. Dies könnte jedoch dazu führen, dass Punkte verworfen werden, die keine große Distanz zum Sensor aufweisen.

Dieser grob gefilterte Datensatz kann durch verschiedene Verfahren weiter ausgedünnt werden. Im Verlauf dieses Abschnitts wird untersucht welches Verfahren sich am besten zur Reduzierung der Scanpunkte eignet. Um diese sogenannten Sampling Verfahren bewerten zu können wurde von Gelfand et al. [GIRL03] ein Maß für die

geometrische Stabilität entwickelt. Dieses Maß beschreibt, wie stabil zwei Punkt wolken sind, wenn sie mit Hilfe der Point-to-Plane Metrik (siehe [Abschnitt 3.1](#)) registriert werden. Da diese Metrik unter anderem auf den Normalvektoren der Punkt wolle ke basiert, wird in dem nächsten Abschnitt zunächst erläutert wie die Normalen bei Punkt wolken bestimmt werden können. Anschließend wird das geometrische Stabilitätsmaß erläutert, welches in [Abschnitt 4.1.3](#) zur Bewertung verschiedener Sampling Verfahren verwendet wird. Die anhand dieser Bewertung gewählte Methode wird in der Folge genauer betrachtet.

#### 4.1.1 Berechnung von Normalvektoren

Bei oberflächenlosen Punkt wolken existieren zwei Möglichkeiten um die Normalvektoren zu bestimmen. Zum einen kann die Punkt wolle durch Triangulierung zu einem Oberflächenobjekt konvertiert werden, um anschließend konventionell die Normale an jedem Punkt bestimmen zu können. Zum anderen können Approximationsverfahren genutzt werden, welche die Normalen an jedem Punkt der Punkt wolle schätzen. Eines dieser Approximationsverfahren stammt von Berkmann und Caelli [[BC94](#)]. Zunächst wird eine lokale Nachbarschaft festgelegt, welche entweder aus den  $k$ -nächsten Nachbarn oder den Punkten innerhalb eines Radius bestehen kann. Nun wird eine Hauptkomponentenanalyse mit diesen Punkten durchgeführt (siehe [Abschnitt 2.3.3](#)). Die Eigenvektoren, die sich daraus ergeben, definieren eine orthogonale Basis. Die zwei Eigenvektoren mit den höchsten dazugehörigen Eigenwerten formen dabei eine Ebene, die im Least-Squares Sinne eine Ebene in die Punktmenge einpasst. Diese Ebene kann als Tangentialebene zur Oberfläche betrachtet werden. Aufgrund der Orthogonalität steht der Eigenvektor mit dem kleinsten Eigenwert senkrecht auf dieser Ebene und wird somit als Normalvektor definiert.

Nach Rusu [[Rus09](#)] existiert keine mathematische Möglichkeit das Vorzeichen des Normalvektors zu bestimmen, wenn dieser durch eine Hauptkomponentenanalyse berechnet wurde. Dies führt in vielen Fällen zu Problemen. Beispielsweise wird in [Abschnitt 5.3.1](#) eine Methode zum Verwerfen von Punktpaaren beschrieben, dass auf dem Winkel zwischen den beiden zugehörigen Normalen basiert. Haben zwei Normalvektoren exakt dieselbe Ausrichtung, könnte es bei einer zufälligen Bestimmung des Vorzeichens zu einer fehlerhaften Verwerfung des Punktes kommen, da der Winkelabstand  $180^\circ$  anstatt  $0^\circ$  betragen könnte. Da aber bei jedem eingehenden Scandatensatz die Sensorposition  $s_p$  bekannt ist, kann mit Hilfe dieser Position das Vorzeichen bestimmt werden, indem man die Normalen immer in Richtung des Sensors orientiert. Erfüllt ein Punkt  $p_i$  dabei die [Formel 4.1](#) nicht, wird die Normale mit -1 multipliziert.

$$\vec{n_i} \cdot (s_p - p_i) > 0 \quad (4.1)$$

Wie in Abbildung 4.1 deutlich wird, funktioniert dieses Prinzip bei Normalvektoren in fast allen Fällen sehr gut. Probleme entstehen bei Vektoren, die fast parallel zur  $z$ -Achse ausgerichtet sind. Eine Möglichkeit diesen Umstand zu beheben wäre eine zusätzliche Abfrage für Punkte, deren Normalvektoren dieser Ausrichtung entsprechen. Dann könnte eine erneute Suche der  $k$ -nächsten Nachbarn stattfinden, um einen Abgleich der Ausrichtungen zu vollziehen. Eine zweite Möglichkeit wäre ein Vergleich der  $z$ -Koordinaten des Punktes mit der des Sensors. Liegt der Punkt unterhalb des Sensors wird der Normalvektor nach oben ausgerichtet. In dieser Arbeit wurde auf diese Erweiterungen verzichtet, da sehr viele Punkte auf dem Boden gescannt werden und dementsprechend viele Normalen Parallelität zur  $z$ -Achse aufweisen. Demnach ist das Verhältnis zwischen Nutzen und Rechenaufwand nicht angemessen.

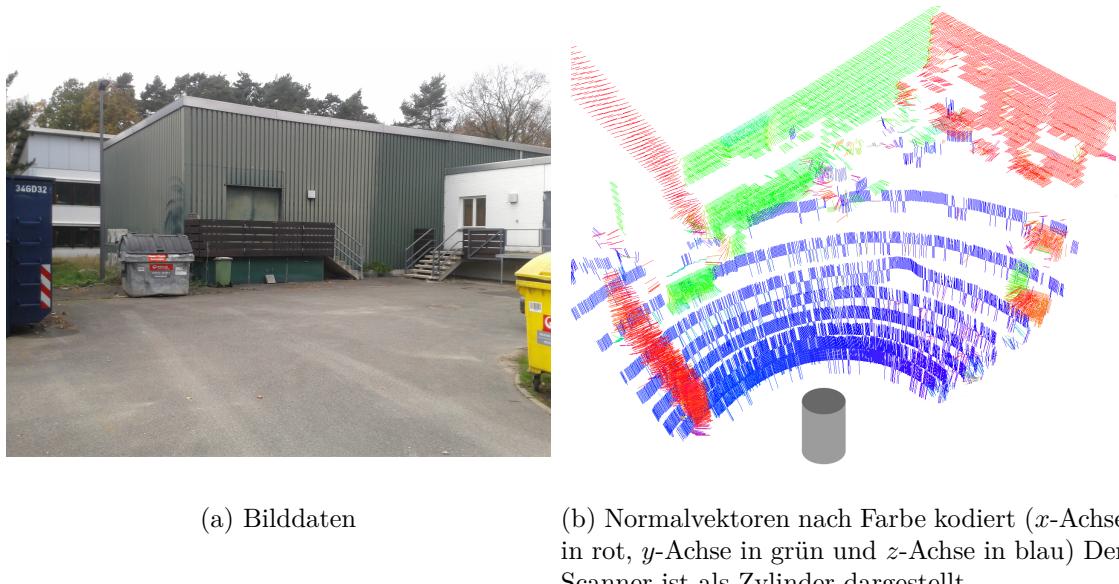


Abbildung 4.1: Kamera- und Laserscannerdaten auf einem Teil des DLR Geländes

Die Verwendung der  $k$ -nächsten Nachbarn und die Bestimmung der Nachbarn durch einen Radius bieten jeweils Vor- und Nachteile. Der Vorteil der Radius-Suche besteht darin, dass keine Punkte verwendet werden, die weit von der abzubildenden Oberfläche entfernt sind und somit einen schlechten Einfluss auf die Bestimmung der Ebene haben. Es besteht jedoch die Gefahr, dass die Anzahl an Punkten im Radius zu klein oder im schlechtesten Fall 0 ist und daher eine schlechte Repräsentation der Oberfläche entsteht. Bei den  $k$ -nächsten Nachbarn ist diese Gefahr wiederum nicht gegeben, es könnten jedoch weit entfernte Punkte mit einbezogen werden. In dieser

Arbeit wird die Nachbarschaft über die 10 nächsten Nachbarn gebildet, da somit eine konstante Berechnungszeit garantiert wird. Wie in Abbildung 4.1 zu erkennen ist, führt dieser Wert bei dem vorliegenden Datensatz zu brauchbaren Ergebnissen. Die Suche findet hierbei ebenfalls durch einen  $k$ D-Baum statt. Im Gegensatz zur Bestimmung des nächsten Nachbarn sind minimale Veränderungen notwendig. Zunächst müssen die  $k$  niedrigsten bisher gefundenen Distanzen gespeichert werden anstatt der minimalen Distanz. Außerdem wird ein Zweig des Baumes nicht weiter durchsucht, wenn kein Punkt näher an dem Anfragepunkt liegen kann als die  $k$  momentan nahegelegenen Punkte.

#### 4.1.2 Geometrische Stabilität

Das entwickelte Maß von Gelfand et al. [GIRL03] beschreibt die geometrische Stabilität zweier Punkt wolken bei einer Registrierung mit der Point-to-Plane Metrik. Dabei ist der Registrierungsfehler  $E$  durch folgende Formel gegeben.

$$E = \sum_{i=1}^k ((\mathbf{R}p_i + \vec{t} - q_i) \cdot \vec{n}_i)^2 \quad (4.2)$$

Hierbei sind die Punkte der Eingangspunkt wolke mit  $p$ , die Punkte der Zielpunkt wolke mit  $q$  und die Normalvektoren der Zielpunkt wolke mit  $\vec{n}$  definiert. Die Transformation wird durch die Rotationsmatrix  $\mathbf{R}$  und den Translationsvektor  $\vec{t}$  bestimmt.

$$\vec{t} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \mathbf{R} = \begin{pmatrix} c(\beta)c(\gamma) & s(\alpha)s(\beta)c(\gamma) - c(\alpha)s(\gamma) & c(\alpha)s(\beta)c(\gamma) + s(\alpha)s(\gamma) \\ c(\beta)s(\gamma) & s(\alpha)s(\beta)s(\gamma) + c(\alpha)c(\gamma) & c(\alpha)s(\beta)s(\gamma) - s(\alpha)c(\gamma) \\ -s(\beta) & s(\alpha)c(\beta) & c(\alpha)c(\beta) \end{pmatrix} \quad (4.3)$$

Da  $\mathbf{R}$  nichtlineare Funktionen, wie den Sinus und den Cosinus<sup>1</sup> der Rotationswinkel, enthält, nehmen Gelfand et al. an, dass die Rotationsmatrix linearisiert werden kann, da die inkrementelle Rotation gering ist. Diese Annahme wird durch die im Evaluationskapitel (siehe Kapitel 6) verwendeten Datensätze gestützt. Kleine Winkel<sup>2</sup>  $\theta$  können geschätzt werden indem  $\cos(\theta)$  mit 1,  $\sin(\theta)$  mit  $\theta$  und  $\sin^n(\theta)$  mit 0 ersetzt werden. Die approximierte Rotationsmatrix für die drei Winkel lautet:

$$\mathbf{R}_{app} = \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix} \quad (4.4)$$

---

<sup>1</sup>In der Formel 4.3 mit c für Cosinus und s für Sinus dargestellt

<sup>2</sup>In dieser Arbeit sind kleine Winkel als  $\theta \leq \frac{1}{2}$  Bogenmaß definiert [Boa06]

Dabei stehen  $\alpha$ ,  $\beta$  und  $\gamma$  jeweils für die Rotation um die  $x$ -,  $y$ - und  $z$ -Achse. Ersetzt man nun  $\mathbf{R}$  mit  $\mathbf{R}_{app}$  in der Formel 4.2 ergibt dies nach Umformungen und Substitution folgende Gleichung (Herleitung ist in Abschnitt A.1 gegeben).

$$E = \sum_{i=1}^k ((p_i - q_i) \cdot \vec{n}_i + \vec{r} \cdot (p_i \times \vec{n}_i) + \vec{t} \cdot \vec{n}_i)^2 \quad (4.5)$$

Die komplette Transformation kann nun als ein 6-Vektor  $(\vec{r}^T \quad \vec{t}^T)^T$  dargestellt werden. Die Terme der Formel 4.5 können dahingehend verstanden werden, dass jedes Punktpaar  $(p_i, q_i)$  eine translativ Kraft in die Richtung von  $\vec{n}_i$  und eine Drehkraft um die Achse  $(p_i \times \vec{n}_i)$  zu  $P$  beisteuert. Daher verändert sich der Fehler eines einzelnen Punktpaares durch die Transformation  $(\Delta \vec{r}^T \quad \Delta \vec{t}^T)^T$  wie folgt:

$$\Delta d_i = (\Delta \vec{r}^T \quad \Delta \vec{t}^T) \begin{pmatrix} (p_i \times \vec{n}_i) \\ \vec{n}_i \end{pmatrix} \quad (4.6)$$

Dies führt zu einer  $6 \times 6$  Kovarianzmatrix  $\mathbf{C}$  für die ganze Punktwolke (Herleitung ist in Abschnitt A.1 gegeben)

$$\mathbf{C} = \mathbf{F} \mathbf{F}^T = \begin{pmatrix} (p_1 \times \vec{n}_1) & \dots & (p_k \times \vec{n}_k) \\ \vec{n}_1 & \dots & \vec{n}_k \end{pmatrix} \begin{pmatrix} (p_1 \times \vec{n}_1)^T & \vec{n}_1^T \\ \dots & \dots \\ (p_k \times \vec{n}_k)^T & \vec{n}_k^T \end{pmatrix} \quad (4.7)$$

Anhand dieser Matrix können nun instabile Transformationen, auch Sliding-Transformationen genannt, ausfindig gemacht werden. Sliding beschreibt generell den Prozess, bei dem Punktwolken aneinander vorbeigeschoben werden. Die Abbildung 4.2 verdeutlicht diese Art von Transformationen.

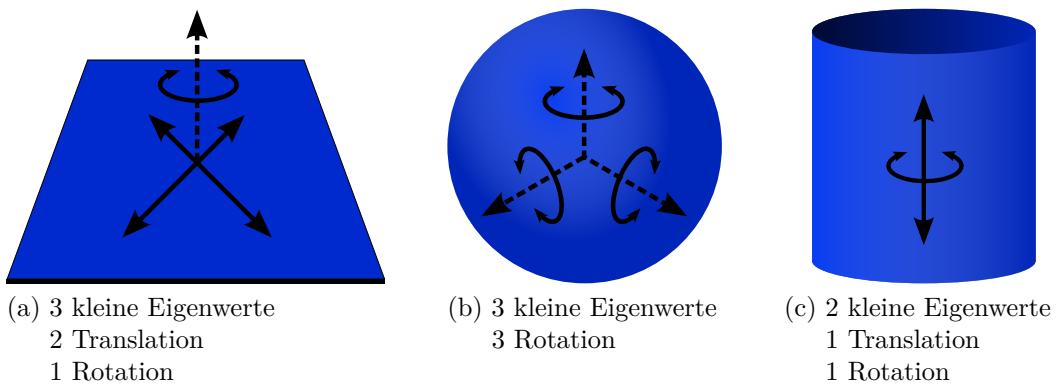


Abbildung 4.2: Geometrisch instabile 3D Formen. Unter jeder Form steht die Anzahl und der Typ der instabilen Transformationen. Nach [GIRL03]

Die Sliding-Richtungen sind dabei durch diejenigen Eigenvektoren von  $\mathbf{C}$  gegeben, deren Eigenwerte im Verhältnis zum größten Eigenwert klein sind. Demnach ent-

spricht die Anzahl an Eigenvektoren dem der Freiheitsgrade, die durch die Geometrie nicht begrenzt werden. Im Umkehrschluss sind Eigenvektoren mit großen Eigenwerten geometrisch stabil. Um dies quantifizieren zu können, wird die Konditionszahl der Kovarianzmatrix verwendet, die sich durch die Division des kleinsten mit dem größten Eigenwert ergibt. Demnach wäre eine Konditionszahl von 1 optimal.

### 4.1.3 Analyse verschiedener Sampling Verfahren

Im Folgenden werden verschiedene Sampling Verfahren beschrieben und in [Abbildung 4.3](#) und [Abbildung 4.4](#) dargestellt. Die Bewertung wird mittels des Maß der geometrischen Stabilität durchgeführt.

#### Octree Sampling

Das naheliegenste Verfahren zum Abtasten des Scandatensatzes ist die Verwendung eines zweiten Octrees, welcher dieselbe Parametrisierung aufweist wie der globale Octree. Dabei wird die Punktwolke zunächst in einen leeren Octree überführt und anschließend werden die Zentren der belegten Voxel gefiltert. Somit wird die Anzahl der Punkte basierend auf der festgelegten Größe der Voxel und der Dichte der Punktwolke minimiert.

#### Uniform Sampling

Eine beliebte Methode stellt das Uniform Sampling [TL94] dar, bei dem die Punktwolke ebenfalls von einem dreidimensionalen Voxelgitter umgeben wird. Anschließend wird aus den Punkten in jedem Voxel ein Durchschnittspunkt berechnet, der somit alle Punkte des Voxels repräsentiert. Dieses Verfahren ähnelt dem Octree Sampling. Die uniforme Methode hat dabei den Vorteil, dass die zugrundeliegende Oberfläche besser durch den Durchschnittspunkt als durch die Voxelzentren approximiert wird. Dies wird in [4.3a](#) und [4.3b](#) verdeutlicht, da die Octreezentren im Gegensatz zu den Durchschnittspunkten viel seltener auf der Oberfläche der Objekte liegen.

#### Random Sampling

Beim Random Sampling werden Punkte zufällig ausgewählt. Dies geschieht anhand eines Zufallszahlengenerators, der in jedem Iterationsschritt des ICP eine andere Untermenge auswählt. In Umgebungen, in denen nur kleine charakteristische Stellen vorkommen, besteht die Gefahr mit diesen Verfahren nur sehr wenige Punkte dieser Stellen zu erhalten. Da es beim Uniform Sampling hingegen zum zu starken Glätten

dieser Regionen kommen kann, wurden weitere Sampling Methoden wie das Normal Space Sampling und das Covariance Sampling entwickelt.

### Normal Space Sampling

Das Normal Space Sampling von Rusinkiewicz und Levoy [RL01] wählt Punkte unter dem Kriterium aus, dass die Verteilung der Normalvektoren so groß wie möglich ist. Hierfür wird ein Histogramm erstellt, in dessen Klassen die Punkte anhand der Position ihres Normalvektors im Winkelraum eingeteilt werden. Schließlich wird eine abgetastete Punktwolke generiert, indem aus jeder Klasse des Histogramms ein zufälliger Punkt ausgewählt wird.

### Covariance Sampling

Das Ziel des Covariance Sampling ist die Punkte so zu selektieren, dass eine möglichst stabile Registrierung mit der Originalpunktfolge erreicht wird. Dieses Verfahren von Gelfand et al. [GIRL03] nutzt die Konditionszahl der Kovarianzmatrix des geometrischen Maßes, um Punkte so auszuwählen, dass die Konditionszahl der resultierenden Punktwolke möglichst nah an 1 liegt.

### Fazit

Rusinkiewicz und Levoy [RL01] vergleichen Uniform Sampling, Random Sampling und Normal Space Sampling in Bezug auf ihren Einfluss auf die Konvergenz bei der Nutzung eines ICP. Sie kommen zu dem Ergebnis, dass bei Umgebungen mit einer hohen Verteilung an Normalen die Sampling Methoden vergleichbare Ergebnisse liefern. Ist dies jedoch nicht der Fall erreicht nur das Normal Space Sampling eine zufriedenstellende Registrierung. Obwohl Uniform Sampling und Random Sampling fast identische Resultate produzieren, hat das Random Sampling einen theoretischen Vorteil, da durch die Auswahl zufälliger Punkte in jeder Iteration die Wahrscheinlichkeit reduziert wird, inkorrekte Korrespondenzen zu nutzen. Ichim [Ich13] hingegen zieht das Covariance Sampling hinzu und testet die Verfahren an fünf verschiedenen RGB-Punktwolken, die in einem Bürokomplex aufgezeichnet wurden. Dies hat zur Folge, dass vor allem rechtwinklige Objekte wie Wände und Tische festgehalten wurden. Das Bewertungskriterium ist das vorher beschriebene geometrische Maß nach der Abtastung. Das Normal Space Sampling erzielt hierbei viermal das beste und einmal das zweitbeste Ergebnis. Das Covariance Sampling erzielt einmal die niedrigste, zweimal die zweitniedrigste und zweimal die höchste Konditionszahl. Am schlechtesten schneidet dieses Verfahren bei Umgebungen mit vielen kleinen Gegenständen ab.

Im Rahmen dieser Arbeit wurden die Samplingverfahren an zwei verschiedenen Punktwolken getestet. Zum Überprüfen der Ergebnisse der Literatur wurden Objekte mit verschiedenen Formen ausgewählt. Diese sind zum einen der Stanford-Hase, da dessen Form viele verschiedene Normalvektoren beinhaltet. Zum anderen wurde eine Punktwolke gewählt, die auf dem Gelände des DLR erfasst wurde und sowohl Gebäudeteile als auch Kraftfahrzeuge und Vegetation beinhaltet. Aufgrund der Dichte der Punkte wurde der Datensatz des Stanford-Hasen von ca. 36.000 und der DLR-Gelände Datensatz von ca. 33.000 auf 2.500 Punkte reduziert. Zur Bewertung des Abtastens wurde abermals das geometrische Maß, also die Konditionszahl der Kovarianzmatrix, verwendet. Die Ergebnisse sind in Tabelle 4.1 dargestellt und Abbildung 4.3 respektive Abbildung 4.4 zeigen die resultierenden Punktwolken der beiden Scandatensätze.

Datensatz	Konditionszahl				
	Octree	Uniform	Random	Normal	Covariance
Standford-Hase	5,44288	5,6673	5,83894	5,3432	4,00592
DLR-Gelände	5,89850	5,24859	9,03078	4,88218	2,89937

Tabelle 4.1: Konditionszahlen der Scandatensätze Stanford-Hase und DLR-Gelände

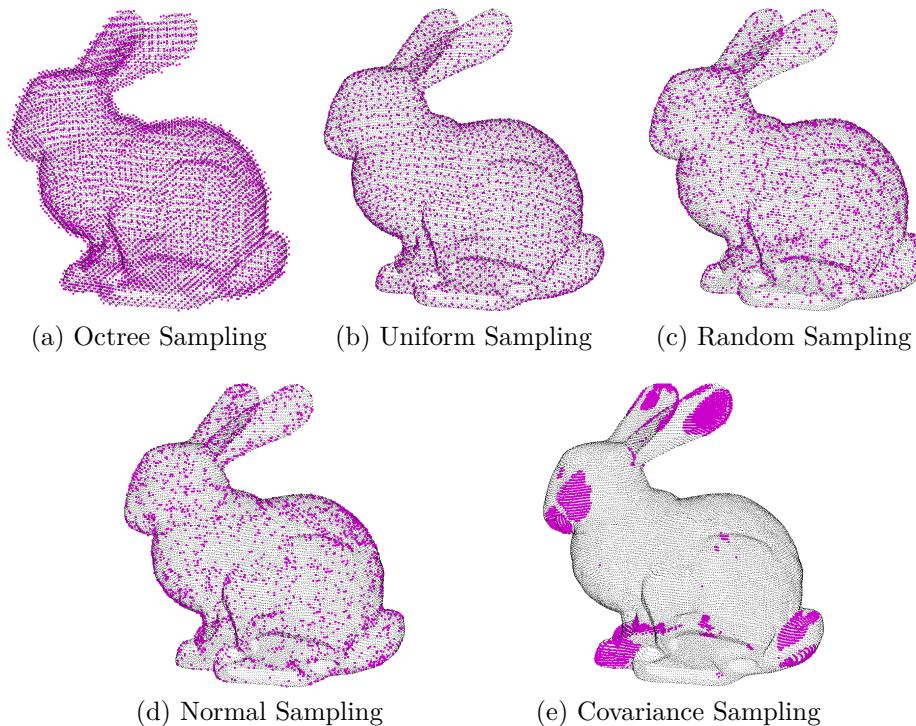


Abbildung 4.3: Vergleich verschiedener Sampling Methoden anhand des Stanford-Hasen. Die gesamte Punktwolke ist in schwarzen Punkten dargestellt und die abgetasteten Punkte sind größer und magentafarben hervorgehoben.

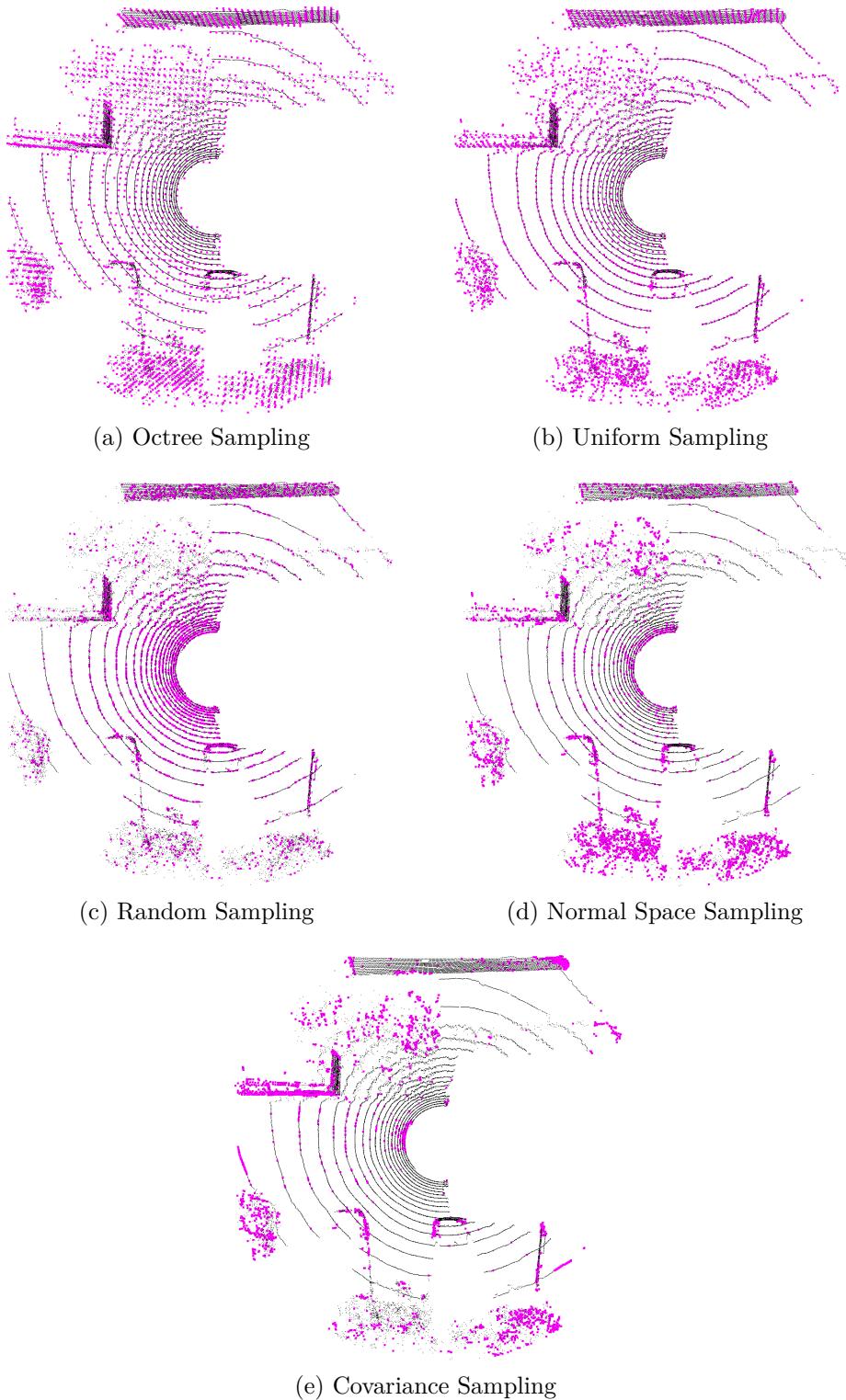


Abbildung 4.4: Vergleich verschiedener Sampling Methoden anhand eines Scandatensatzes, der auf dem DLR-Gelände aufgenommen wurde. Es handelt sich um eine Vogelperspektive, die am unteren Rand Vegetation und Teile von drei PKWs zeigt, am oberen Rand eine Mauer und links davon einen Transporter. Die gesamte Punkt-  
wolke ist in schwarzen Punkten dargestellt und die abgetasteten Punkte sind größer  
und magentafarben hervorgehoben.

In beiden Fällen schneidet das Covariance Sampling am besten und das Normal Space Sampling am zweitbesten ab. Auf den Abbildungen wird deutlich, dass das Covariance Sampling besonders markante Stellen abtastet. Beim Stanford-Hasen sind diese Stellen vor allem die Ohren, die Pfoten, die Schnauze und der Schwanz. Beim Gelände-Datensatz werden zumeist Teile der Wände, Autos und Büsche ausgewählt. Das Normal Space Sampling verteilt beim Stanford-Hasen die gefilterten Punkte mehr über die ganze Punktwolke. Es befinden sich viele Punkte an den vorderen Pfoten, eine Stelle wo die Normalvektoren sich mehr vom restlichen Datensatz unterscheiden. Beim Gelände-Datensatz verhält es sich ähnlich dem Covariance Sampling, wählt aber mehr Punkte am Boden und in der Vegetation. Vorallem die Punkte in der Vegetation sind eher unzuverlässige Punkte, da sich die Blätter der Büsche durch den Wind bewegen und somit zu verschiedenen Zeitpunkten die Laserstrahlen unterschiedlich reflektieren. Außerdem lässt sich der Normalvektor an diesen Stellen nicht zuverlässig bestimmen. Anhand dieses Tests wurde in dieser Arbeit das Covariance Sampling ausgewählt, um die Anzahl der Punkte zu reduzieren und besonders charakteristische Stellen zu filtern. Daher wird das Covariance Sampling im folgenden Abschnitt noch einmal gesondert betrachtet.

Zusätzlich kann die Zielpunktwolke ebenso durch diese Verfahren weiter gefiltert werden. Das Anwenden eines Sampling Verfahrens würde die Anzahl an Iterationen des **ICP** verringern. Da bereits durch den Octree ein ausreichender Abstraktionsgrad gegeben ist, wird in dieser Arbeit auf weiteres Abtasten verzichtet.

Es gibt noch weitere komplexere Verfahren um die Untermengen einer Punktwolke zu filtern. Diese finden Anwendung in der groben Registrierung um zwei Punktwolken so zu transformieren, dass die feinen Registrierungsmethoden nicht zu fehlerhaften lokalen Minima konvergieren. Hierbei wird die Punktwolke auf wenige Punkte, sogenannte Keypoints, reduziert, zu denen viel komplexere Charakteristiken bestimmt werden.

#### 4.1.4 Covariance Sampling

Das Ziel des Covariance Samplings ist es nun die Punkte so zu selektieren, dass die Konditionszahl der resultierenden Kovarianzmatrix möglichst nahe an 1 liegt. Dafür wird ein ähnlicher Vorverarbeitungsschritt wie bei der **PCA** (Abschnitt 2.3.3) verwendet. Da die Rotation von dem Term  $p_i \times \vec{n}_i$  abhängig ist, sind die Rotationen von der Distanz des Punktes  $p_i$  zum Ursprung abhängig, da dieser das Zentrum der Rotation von  $\vec{r}$  darstellt. Daher wird aus allen Punkten in der Punktwolke der Mittelpunkt gebildet, der anschließend von allen Punkten subtrahiert wird. Somit ist der Mittelpunkt gleich dem Zentrum der Rotation. Die Größenordnung der Ro-

tation und Translation können auch inkompatibel sein, da ein Punkt beliebig weit vom Ursprung entfernt sein kann. Um das Einfluss von Rotation und Translation anzugleichen wird die Punktwolke skaliert, so dass die durchschnittliche Entfernung vom Ursprung 1 beträgt. Somit wird der maximale Anteil der translativen Kraft eines Punktes dem maximalen Anteil der Drehkraft angeglichen.

Anschließend wird die Kovarianzmatrix  $\mathbf{C}$  erstellt, dessen sechs Eigenvektoren  $\vec{x}_1, \dots, \vec{x}_6$ , wie in [Abschnitt 2.3.2](#) beschrieben, berechnet werden. Für jeden Punkt  $p_i$  wird nun ein Vektor mit sechs Einträgen gebildet.

$$\vec{v}_i = (p_i \times \vec{n}_i, \vec{n}_i) \quad (4.8)$$

Außerdem werden sechs Listen  $L_1, \dots, L_6$  aufgebaut, die jeweils alle  $\vec{v}_i$  enthalten. Jede Liste  $L_k$  wird nun absteigend sortiert anhand des Skalarprodukts von  $\vec{v}_i$  und  $\vec{x}_k$ , also dem  $k$ -ten Eigenvektor. Jede Liste speichert also die Punkte in absteigender Reihenfolge ihres Beitrags zur geometrischen Stabilität zu einer der sechs Freiheitsgrade.

Nun werden die Punkte in einem iterativen Prozess so ausgewählt, dass die Eigenwerte möglichst gleich bleiben und somit eine niedrige Konditionszahl erzielt wird. Dafür werden zunächst sechs Werte  $t_1, \dots, t_6$  definiert, die als momentane Schätzung der sechs Eigenwerte dienen. Diese Werte werden durch die folgende Formel bestimmt.

$$t_k = \sum_{i=0}^m (\vec{v}_i \cdot \vec{x}_k)^2 \quad (4.9)$$

$(\vec{v}_i \cdot \vec{x}_k)^2$  steht für die Größe des Fehlers, der auftritt wenn der Punkt  $p_i$  von seiner optimalen Position durch  $\vec{x}_k$  transformiert wird.  $m$  repräsentiert die Anzahl der bisher selektierten Punkten, demnach werden alle  $t_k$  nach jedem neu ausgewählten Punkt aktualisiert.

Das iterative Auswahlverfahren wird beendet, wenn eine festgelegte Anzahl an Punkten selektiert wurde. Da Gelfand et al. für echte Scandaten 10% der Punktwolken nutzen, wird in dieser Arbeit ebenso verfahren. Der iterative Prozess gliedert sich dabei in drei Schritte:

1. Bestimmung des momentan kleinsten geschätzten Eigenwertes  $t_k$
2. Auswahl des Punktes mit der größten geometrischen Stabilität aus  $L_k$
3. Aktualisierung der geschätzten Eigenwerte

## 4.2 Eingrenzung des Suchraums

Für die Registrierung des aktuellen Scandatensatzes ist es notwendig die Zielpunkt-wolke zu definieren. Die Zielpunktwolke wird aus den Voxelzentren des Octrees gebil-det, der die globale Hinderniskarte darstellt. Dieser kann eine sehr große Umgebung repräsentieren, welche die Sensorreichweite übersteigen kann. Damit sind Teile des Octrees für die aktuelle Registrierung irrelevant, da sie zu weit von der Sensorpo-sition entfernt liegen, um positiv zur Registrierung beitragen zu können. Deshalb wurde eine Lösung entwickelt, die abhängig von den Eigenschaften des verwende-ten Laserscanners, die belegten Voxel des Octrees filtert. Laserscanner inkludieren verschiedene Eigenschaften wie die Sensorreichweite in jeden Scandatensatz, so dass der hier präsentierte Ansatz sich automatisch an den aktuellen Scanner anpasst.

Das grundlegende Prinzip ist das Bilden einer Kugel um die Sensorposition, des-sen Radius durch die Sensorreichweite gegeben ist. Die Zentren der Voxel, die sich wenigstens zum Teil innerhalb dieser Kugel befinden, werden der Zielpunktwolke hinzugefügt. Bei diesem Ansatz kann der Radius noch weiter eingegrenzt werden, indem der Radius auf die Distanz zum Punkt mit der größten Entfernung gesetzt wird. Auf den Radius wird ein Versatz von mehreren Metern addiert, da die Mög-lichkeit besteht, dass die Sensorposition durch die Navigationsdaten nicht fehlerfrei bestimmt wird.

Die Überprüfung der Voxel erfolgt hierarchisch und mit steigender Rechenkomple-xität. Als erstes wird geprüft ob die Position des Sensors innerhalb des Voxels liegt. Dies ist der Fall, wenn die Differenz zwischen der Sensorposition und des Voxelzen-trums in allen Dimensionen jeweils kleiner ist als die halbe Kantenlänge des Voxels. Ist dieser simple Test erfolgreich, werden die Kindknoten des dazugehörigen Knoten weiter untersucht. Ist dieser Knoten ein Blatt, wird das Zentrum des Voxels in der Punktwolke gespeichert.

Ein negativer Ausgang hat einen weiteren Test zur Folge, der feststellen soll ob sich die Kugel und der betreffende Voxel überschneiden. Hierfür wurde der Algorithmus von Arvo [Arv90] verwendet. Dieser Algorithmus findet den Punkt  $p$  in oder auf dem Voxel, der am nächsten an dem Zentrum der Kugel  $c$ , welche als Sensorposition definiert ist, liegt. Ist die resultierende Distanz  $d(p, c)$  kleiner als der Radius  $r$ , überschneiden sich Kugel und Voxel. Zur vereinfachten Berechnung wird auch in diesem Fall die quadrierte Distanz  $d^2$  und dementsprechend der quadrierte Radius  $r^2$  verwendet.

Die Distanz wird in jeder Dimension unabhängig berechnet, was die Anwendbarkeit dieses Algorithmus auf beliebig viele Dimensionen ermöglicht. Zunächst werden zwei Punkte des Voxels definiert. Zum einen der Punkt  $b^{min}$ , der die minimalen Werte

des Voxels in allen Dimensionen vereint. Dieser Punkt kann als Eckpunkt angesehen werden, welcher die linke, untere und vordere Seite des Voxels vereint. Zum anderen wird der gegenteilige Punkt  $b^{max}$  mit den maximalen Werten in allen Dimensionen definiert. Nun wird zunächst die quadrierte Distanz  $d^2$  mit null initialisiert. Für jede Dimension  $i$  wird getestet ob  $b_i^{min} \leq c_i \leq b_i^{max}$ . Verläuft dieser Test erfolgreich, liegt das Zentrum in dieser Dimension innerhalb des Voxels. Ansonsten wird die quadrierte Distanz zwischen  $c_i$  und  $b_i^{min}$  oder  $b_i^{max}$  berechnet, je nachdem welcher Wert näher an  $c_i$  liegt. Diese Distanz wird nun auf  $d^2$  addiert. Nachdem alle Dimensionen durchlaufen wurden, liegt eine Überschneidung vor wenn  $d^2 < r^2$  erfüllt ist.

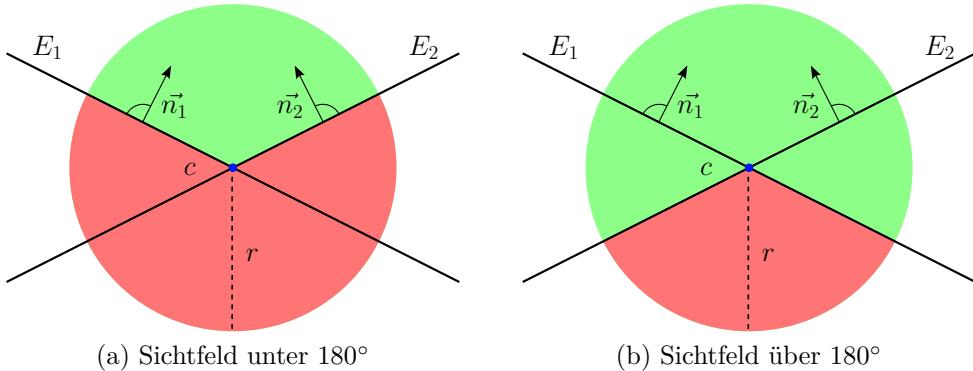


Abbildung 4.5: Vogelperspektive des horizontalen Sichtfelds des Laserscanners (grün). Irrelevante Bereiche sind in rot dargestellt

Außerdem ist es möglich die Anzahl an Punkten weiter einzuschränken, indem das horizontale Sichtfeld des Laserscanners mit einbezogen wird. Das horizontale Sichtfeld ist durch einen Anfangs- und einen Endwinkel bestimmt und wird ebenfalls in jedem Datenpaket des Scanners mitgesendet. Die Abbildung 4.5 stellt dieses Verfahren aus der Vogelperspektive dar und beinhaltet die beiden Fälle, die bei der Umsetzung beachtet werden müssen. In 4.5a ist der Öffnungswinkel des horizontalen Sichtfelds kleiner als  $180^\circ$  und in 4.5b größer als  $180^\circ$ . Um zu testen welcher Voxel im Sichtfeld liegt, werden zwei Ebenen  $E_1$  und  $E_2$  definiert. Für jede Ebene werden drei Punkte benötigt. Der erste Punkt entspricht der Sensorposition. Die  $x$ - und  $y$ -Koordinaten für den zweiten und dritten Punkt werden durch einfache Trigonometrie berechnet, wobei  $\theta$  bei der ersten Ebene dem Anfangswinkel und bei der zweiten Ebene dem Endwinkel entspricht.

$$\begin{aligned} x &= c.x + (r \cdot \cos(\theta)); \\ y &= c.y + (r \cdot \sin(\theta)); \end{aligned} \tag{4.10}$$

Die  $z$ -Koordinate wird nun für den zweiten Punkt mit der  $z$ -Koordinaten von  $c$  gleichgesetzt und beim dritten Punkt mit dem Radius, um die Ebene aufzuspannen. Aus den drei resultierenden Punkten werden nun zwei Vektoren berechnet, dessen Kreuzprodukt die Oberflächennormale ergibt. Dabei werden die Vektoren so ausgewählt, dass der Normalvektor nach der Rechte-Hand-Regel in Richtung des Sichtfeldes zeigt (siehe Abbildung 4.5). Außerdem wird der Normalvektor anhand seiner Länge normalisiert.

Nach der Definition der beiden Ebenen kann nun getestet werden ob sich ein Voxel innerhalb des horizontalen Sichtfeldes befindet. Hierfür wird für jeden Eckpunkt des Voxels ein Vektor von einem beliebigen Punkt auf der Ebene, beispielsweise die Sensorposition, bestimmt. Anschließend wird das Skalarprodukts des resultierenden Vektors mit der Oberflächennormale berechnet. Liegt die Ecke auf der Ebene ist das Skalarprodukt null, liegt die Ecke vor der Ebene ist das Skalarprodukt positiv und hinter der Ebene negativ. Ein Voxel liegt nun vor der Ebene, wenn sich mindestens eine Ecke vor der Ebene befindet.

Nun findet, wie eingangs beschrieben, eine Fallunterscheidung statt, ob das horizontale Sichtfeld kleiner oder größer als  $180^\circ$  ist. Ist der Winkel kleiner muss der zu überprüfende Voxel vor beiden Ebenen liegen, ist er größer muss er nur vor mindestens einer der beiden Ebenen liegen. Außerdem gibt es noch zwei Spezialfälle. Bei einem Sichtfeld von genau  $180^\circ$  ist nur eine Ebene notwendig und bei  $360^\circ$  ist der Test überflüssig. Außerdem ist zu beachten, dass die korrekte Lage im Raum betrachtet wird. Dies wird garantiert indem das horizontale Sichtfeld durch den von den Navigationsdaten gelieferten Gier-Winkel rotiert wird. Auch bei diesem Test ist es notwendig einen Versatz bei der Bestimmung der Ebenen einzubauen, um durch fehlerhafte Navigationsdaten keine relevanten Voxel auszuschließen.

Bei diesem Ansatz wird der Einfluss des Roll- und Nick-Winkels auf die Bestimmung der Ebenen nicht berücksichtigt. Ebenso könnte das vertikale Sichtfeld auf dieselbe Weise eingegrenzt werden, was schließlich zu einer Art Frustum Culling<sup>3</sup> führen würde.

---

<sup>3</sup>Beim Frustum Culling wird in der 3D Computergrafik ein Kegelstumpf als Sichtbereich definiert

# 5. Iterative Closest Point

Nachdem die Eingangs- und Zielpunktfolge durch die im vorangegangenen Kapitel beschriebenen Methoden bestimmt wurden, ist es möglich den Iterative Closest Point Algorithmus anzuwenden. Das Grundgerüst dieses Verfahrens wurde bereits in [Abschnitt 3.1](#) dargestellt. Im weiteren Verlauf dieses Kapitels werden die Varianten des ICP, die diesen in verschiedenen Phasen verbessern bzw. beschleunigen, vorgestellt. Um auf diese Varianten gezielt eingehen zu können, werden nun die Phasen definiert. Hierfür wird auf eine ähnliche Definition wie in [\[RL01\]](#) zurückgegriffen.

1. Bestimmung von korrespondierenden Punkten innerhalb der gewählten Punktmenge
2. Wichtung von Punktpaaren
3. Verwerfung von Punktpaaren
4. Auswahl der zu minimierenden Fehlermetrik

## 5.1 Bestimmung von korrespondierenden Punkten

Das Bestimmen des nächsten Nachbarn ist der Berechnungsschritt mit dem höchsten Zeitaufwand innerhalb des ICP. Daher ermöglicht eine schnelle Suche eine große Leistungssteigerung. Mathematisch beschrieben wird dieses Problem durch [Formel 5.1](#), die aussagt, dass in einer Punktfolge  $P$  mit  $n$  Punkten  $p$  der Punkt  $p_c$  mit der kürzesten euklidischen Distanz zum Anfragepunkt  $q$  der nächste Nachbar ist.

$$d(q, p_c) \leq d(q, p_i) \quad \forall p_i \in P \quad (5.1)$$

Um diesen Nachbarn zu bestimmen wurden verschiedene Verfahren entwickelt, wobei ein Teil dieser Verfahren im Folgenden analysiert wird.

### 5.1.1 Analyse verschiedener Methoden

#### Brute Force Verfahren

Die naheliegendste und gleichermaßen langsamste Methode ist das Brute Force Verfahren, bei dem der Abstand zu jedem Punkt berechnet wird und somit Berechnungszeit von  $O(n)$  entstehen. Da zumeist für die Suche des nächsten Nachbarn der euklidische Abstand verwendet wird, kann zur Optimierung diese Distanz quadriert werden, so dass nicht bei jeder Berechnung die Wurzel gezogen werden muss. Außerdem kann dadurch bereits bei Teilsummen eine Überprüfung stattfinden, ob die momentan niedrigste Distanz überschritten wird. Gilt also im dreidimensionalen Raum  $(q_x - p_{ix})^2 > d^2(q, p_c)$  oder  $(q_x - p_{ix})^2 + (q_y - p_{iy})^2 > d^2(q, p_c)$ , kann frühzeitig abgebrochen und Rechenzeit gespart werden.

Um den Vorgang weiter zu beschleunigen, ist es notwendig, die Anzahl der zu überprüfenden Punkte zu reduzieren. Daher wurden verschiedene Verfahren entwickelt, die in einem Initialisierungsschritt die Punktmenge sortieren. Zinßer et al. [[ZSN03a](#)] haben für diese Verfahren eine spezielle Analyse für das Anwendungsgebiet ICP Registrierung von 3D Punktfolgen durchgeführt. Im Folgenden wird eine Auswahl der überprüften Verfahren erläutert.

#### Triangle Induced Nearest Neighbor

Das Triangle Induced Nearest Neighbor (TINN) Verfahren von Greenspan [[GGT00](#)] ordnet die Punkte  $p$  in einem Vorverarbeitungsschritt nach ihrer Distanz zu einem beliebigen Referenzpunkt  $r$  in eine sortierte Liste ein. Bei einem Anfragepunkt  $q$

wird der Abstand zu  $r$  berechnet. Nun wird durch eine binäre Suche zunächst der Punkt  $p_i$  mit dem ähnlichsten Abstand zu  $r$  als Kandidat  $p_c$  festgelegt. Anschließend wird auf beiden Seiten der Liste nach einem besseren Kandidaten gesucht. Besitzt dabei ein  $p_i$  eine kürzere Distanz zu  $q$  als  $p_c$ , wird dieser Punkt als neuer Kandidat festgelegt. Außerdem wird jedes  $p_i$  auf die folgende Dreiecksungleichung geprüft.

$$|d(q, r) - d(p_i, r)| \leq d(p_i, q) \leq d(q, r) + d(p_i, r) \quad (5.2)$$

Weist  $d(p_i, q)$  beim Durchlaufen der Liste ein Minimum auf, kann keiner der absteigenden Einträge näher am Anfragepunkt liegen als der momentane Kandidat. Dasselbe Verfahren wird durchgeführt, indem die Liste aufsteigend nach nähergelegenen Punkten durchsucht wird. Dieses Verfahren ist speziell für Punktwolken mit einer geringen Anzahl von Punkten ausgelegt, da bei einer hohen Anzahl weite Teile der Liste durchlaufen werden müssten. Eine experimentelle Evaluierung von Zinßer et al. [ZSN03a] zeigt, dass die Berechnungskosten der Suche bei dreidimensionalen Daten etwa bei  $O(n^{\frac{2}{3}})$  liegen. Außerdem sollte das TINN Verfahren nur bei Punktmenigen mit unter 500 Punkten verwendet werden.

Eine spezielle Form [FBS75] dieses Ansatzes schlägt vor, den Referenzpunkt durch die Koordinatenachse auszutauschen, in der die Punkte die größte Varianz aufweisen. Auf diese Art ist es nicht notwendig für jeden Punkt eine zusätzliche Distanz zu speichern, da hierfür der Koordinatenwert des Punkts verwendet werden kann.

### kD-Baum

Ein weiterer Ansatz besteht im geometrischen Unterteilen des Raumes, sodass nur eine kleine Anzahl an Punkten untersucht werden muss. Der kD-Baum, wie bereits in Abschnitt 2.2.2 beschrieben, ist ein  $k$ -dimensionaler binärer Suchbaum, der den Raum hierarchisch in Hyperrechtecke aufteilt. Zum Teilen des Raums werden Hyperebenen verwendet, die orthogonal zu den Koordinatenachsen ausgerichtet sind. Bei der Suche nach dem nächsten Nachbarn, wird zunächst der kD-Baum so durchlaufen als würde der Anfragepunkt in den Baum eingefügt werden. Der Punkt, in dem so erreichten Blatt wird als Kandidat festgelegt. Nun kann anhand der Distanz vom Anfragepunkt zum Kandidaten geprüft werden, welche benachbarten Hyperrechtecke einen näheren Punkt beinhalten könnten. Dies geschieht durch Backtracking im Baum. Die Berechnungskosten für die Suche des nächsten Nachbarn betragen  $O(\log(n))$  [FBF77].

### Box-Dekompositions-Baum

Eine Variante des kD-Baums ist der Box-Dekompositions-Baum [AMN<sup>+</sup>94], der eine größere Robustheit gegenüber Datensätzen aufweisen soll, in denen sich Gruppen

mit nah beieinander liegenden Punkten bilden. Hierbei wird neben der normalen Teilung auch eine Operation zum Schrumpfen (engl. shrinking) des Raumes eingeführt. Hierbei wird ein Hyperrechteck bestimmt, welches aus dem zu teilenden Raum herausgezogen wird. Diese Operation wird verwendet um schnell in Regionen vorzustoßen in denen viele Punkte liegen. Abbildung 5.1 veranschaulicht den Einsatz dieser Operation. Arya et al. geben Berechnungskosten für die Nachbarsuche von  $O(d \log(n))$ , wobei  $d$  die Anzahl der Dimension repräsentiert.

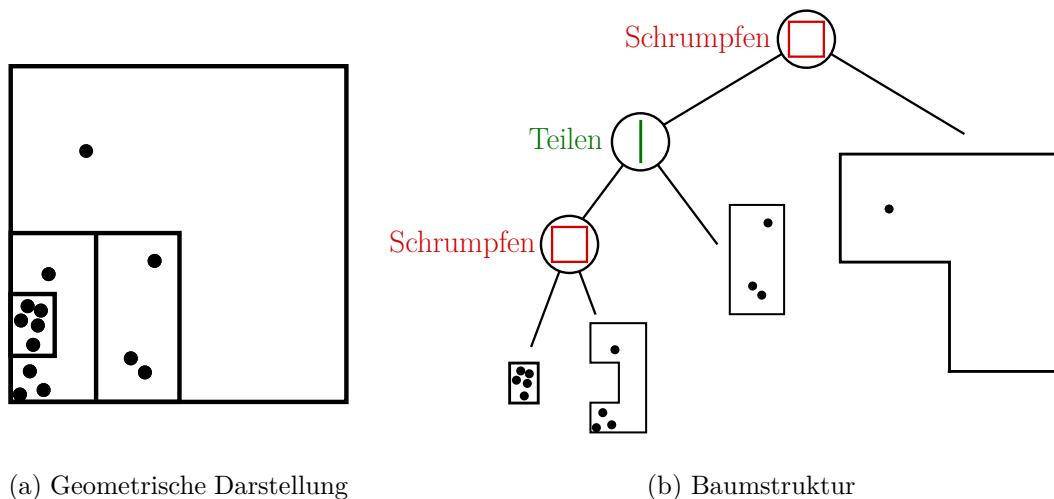


Abbildung 5.1: Beispiel der Operationen Teilen und Schrumpfen des Box-Dekomposition-Baums [AMN<sup>+</sup>94]

## Fazit

In [ZSN03a] wurden die Brute-Force Methode, **TINN**, die Koordinaten-Variante von **TINN** und der  $k$ D-Baum auf die jeweilige Geschwindigkeit beim Einsatz in einem ICP getestet. Dabei wurde festgestellt, dass der  $k$ D-Baum die beste Leistung bietet. Zusätzlich werden Erweiterungen vorgestellt, durch die die Suche weiter beschleunigt wird. Diese Optimierungen wurden aufgrund des begrenzten Umfangs dieser Arbeit nicht weiter untersucht. Nüchter [Nü06] vergleicht die Zeiten von  $k$ D-Baum und Box-Dekomposition-Baum beim Scanmatching und kommt zu dem Ergebnis, dass der  $k$ D-Baum den nächsten Nachbarn schneller ermitteln kann. Demzufolge wurde in dieser Arbeit der  $k$ D-Baum für die Suche des nächstgelegenen Punktes ausgewählt. Im folgenden Abschnitt wird daher genauer auf die Umsetzung eingegangen.

### 5.1.2 kD-Baum

Zur Suche des nächsten Nachbarn  $q$  des Punktes  $p$  wurde in dieser Arbeit auf den kD-Baum zurückgegriffen und verschiedene Änderungen gegenüber dem Standardverfahren (siehe [Abschnitt 2.2.2](#)) eingeführt.

Generell wird bei der Suche ein rekursives Verfahren angewandt. Zunächst wird die kürzeste Distanz  $d = \infty$  initialisiert. Nun werden, beginnend bei der Wurzel, die Knoten rekursiv durchlaufen. Da die Wurzel bzw. die inneren Knoten den Raum jeweils in zwei Halbräume teilen, wird geprüft in welchem Halbraum  $p$  liegt. Dementsprechend findet anschließend eine Rekursion mit dem rechten oder linken Kindknoten statt. Erreicht der Algorithmus ein Blatt wird die Distanz zwischen dem Punkt  $p$  und dem im Blatt gespeicherten Punkt  $q_i$  berechnet. Ist diese Distanz nun kleiner als  $d$  wird  $d$  auf die neue Distanz gesetzt und  $q_i$  als nächstgelegener Punkt zwischengespeichert. Darauffolgend wird eine Sphäre, also in 2D ein Kreis und in 3D eine Kugel, um Punkt  $p$  mit dem Radius  $d$  aufgestellt. Kehrt der Algorithmus nun von dem Blatt zurück in den vorangegangenen inneren Knoten wird getestet ob sich die andere Seite des Halbraumes mit der Sphäre überschneidet. Ist dies der Fall, findet ein rekursiver Aufruf mit dem entsprechenden Kindknoten statt. Andernfalls kann kein Punkt, der im kD-Baum unterhalb dieses Kindknotens liegt, näher an  $p$  liegen als der momentan nächstgelegene Punkt. Nachdem die Traversierung aller in Frage kommenden Knoten abgeschlossen wurde, ist der nächste Nachbar von  $p$  gefunden. Zur weiteren Beschleunigung des Algorithmus wird die quadrierte Distanz bei der Abstandsberechnung verwendet. Ein Beispiel des Suchverfahrens im zweidimensionalen Raum befindet sich in [Kapitel C](#) dieser Arbeit.

Bei der Implementierung des kD-Baums sind vor allem drei Entscheidungen zu treffen. Diese sind die Wahl der maximalen Anzahl an Punkten in jedem Blatt, die Bestimmung der Dimension durch die geteilt werden soll und die Determinierung der Position der Hyperebene.

Der kD-Baum, der in [Abschnitt 2.2.2](#) vorgestellt wurde, speichert nur einen Punkt pro Blatt. Bei einer großen Anzahl an Punkten entsteht ein sehr tiefer Baum, bei dem das rekursive Durchlaufen sehr viel Zeit in Anspruch nimmt. Sind in einem Blatt mindestens zwei Punkte vorhanden, muss die Distanz vom Anfragepunkt zu allen Punkten in dem Blatt berechnet werden. Daher erfordert die Wahl der maximalen Anzahl an Punkten einen Kompromiss zwischen der Anzahl der überprüften Knoten und der Tiefe des Baumes und die damit verbundene Anzahl rekursiver Aufrufe. In dieser Arbeit wurde die maximale Anzahl auf 10 Punkte festgelegt, da die Tests bei unterschiedlichen Datensätzen in [\[ZSN03a\]](#) und [\[Nü06\]](#) zeigen, dass dieser Wert einen guten Kompromiss darstellt.

Auch bei der Wahl der zu schneidenden Dimension wurde von der Standardvariante abgewichen, bei der die Dimension in jeder Baumstufe der Reihe nach alterniert. Stattdessen wird die Dimension anhand der Ausbreitung der Punkte in dem zu teilenden Bereich bestimmt. Es wird die Dimension ausgewählt, in dem die Punkte die größte Spanne aufweisen. Obwohl bei dieser Methodik die gewählte Dimension mitgespeichert werden muss, bietet sie die Chance auf eine bessere Verteilung der Punkte in der gewählten Dimension.

Die Position der Hyperebene wird in dieser Variante auch nicht über den Median der Koordinaten der gewählten Dimension bestimmt. Die Hyperebene wird im Mittelpunkt der ermittelten Spanne angelegt. Durch die Verwendung dieses Mittelpunkts wird verhindert, dass die Kanten der entstehenden Hyperräume ein schlechtes Längenverhältnis aufweisen. Dies kann vor allem bei Datensätzen auftreten, deren Punkte zum Teil eng gruppiert sind. [Moo90]

## 5.2 Wichtung von Punktpaaren

Zu den selektierten Punkten der Eingangspunktwolke werden korrespondierende Punkte in der Zielpunktwolke gesucht, welche dann ein Punktpaar oder auch eine Korrespondenz bilden. Dabei hat jedes Punktpaar einen Einfluss auf die Registrierung. Um diesen Einfluss kontrollieren zu können, ist es möglich die Korrespondenzen anhand spezieller Kriterien zu gewichten. Dies soll die Registrierung verbessern, indem Punktpaare, die wahrscheinlich wenig zur Transformationsfindung geeignet sind, eine geringe Wichtung bekommen. Der originale ICP Algorithmus kann als Beispiel für eine konstante Wichtung betrachtet werden, da alle Punkte gleichermaßen zur Registrierung beitragen.

### 5.2.1 Analyse verschiedener Wichtungsmethoden

#### Wichtung durch die Distanz

Eine simple Möglichkeit besteht in der Wichtung der Korrespondenzen nach dem euklidischen Abstand zwischen den beiden Punkten. So erhalten Punkte nach Formel 5.3 mit größerem Abstand ein kleineres Gewicht  $w$ , wobei  $p$  und  $q$  die korrespondierenden Punkte sind und  $d_{max}$  der größte Abstand zwischen allen Korrespondenzen ist.

$$w = 1 - \frac{d(p, q)}{d_{max}} \quad (5.3)$$

Dieses Verfahren ist für die Verwendung bei Punktwolken, die von Laserscannern produziert worden sind, nicht optimal. Aufgrund des radialen Scanverfahrens bei zwei aufeinanderfolgenden Scandatensätzen haben valide Punktpaare, die weiter von der Sensorposition entfernt sind eine größere Distanz als Punktpaare, die nah am Sensor liegen. Eine Alternative stellt die Verwendung der Winkeldistanz dar. Diese ist sowohl für die naheliegenden Korrespondenzen, als auch für die weiter entfernten identisch. In dieser Arbeit findet diese Methode jedoch keine Anwendung, da durch die Verwendung der Zentren des Octree eine Ungenauigkeit eingebracht wird, würde die Wichtung zum größten Teil zufällig erfolgen.

### Wichtung durch die Normalvektoren

Eine weitere Methode wichtet die Korrespondenzen anhand der Ähnlichkeit der Ausrichtung der dazugehörigen Normalvektoren. Diese Ähnlichkeit kann hierbei mit Hilfe des Skalarprodukts der beiden Normalvektoren bestimmt werden. Je gleichgerichtet die Normalen sind, desto höher werden sie bei der Registrierung gewichtet. Demnach ist die Wichtung durch [Formel 5.4](#) gegeben, wobei  $\vec{n}_p$  und  $\vec{n}_q$  jeweils die Normalvektoren von  $p$  und  $q$  darstellen.

$$w = \vec{n}_p \cdot \vec{n}_q \quad (5.4)$$

Die Verwendung von Normalvektoren besitzt mehrere Nachteile. Zum einen ist ein hoher Rechenaufwand nötig, da in jeder Iteration die korrespondierenden Punkte in der globalen Repräsentation neu zugewiesen werden. Somit müssen auch die Normalen für diese Punkte berechnet werden. Es gibt die Möglichkeit die Normalvektoren jedem Octreeknoten zuzuweisen. Dafür müsste jedoch ein zusätzlicher Update-Vorgang eingeführt werden, da sich die Normalvektoren bei Punkten an den Rändern oder durch Freizeichnen einzelner Knoten ändern könnten. Zusätzlich kann es zum Teil Probleme bei der Berechnung der Normalvektoren aus Punktwolken vorhanden, da bei der Verwendung der [PCA](#) die Orientierung nicht eindeutig ist. Dies kann zur Folge haben, dass einer von zwei identischen Normalvektoren in die entgegengesetzte Richtung zeigt. (siehe [Abschnitt 4.1.1](#))

### Fazit

Die Wichtung kann weder anhand der Distanz noch anhand der Normalvektoren die Registrierung positiv beeinflussen. Zudem wird sowohl in [\[RL01\]](#) als auch in [\[Nü06\]](#) und [\[Ich13\]](#) angegeben, dass bei den durchgeföhrten Tests und Anwendungsfällen die Wichtung keine oder nur minimale Unterschiede liefert. Daher wurde in dieser Arbeit auf die Wichtung von Punktpaaren verzichtet.

## 5.3 Verwerfung von Punktpaaren

Das Verwerfen von Punktpaaren kann als binäre Wichtung betrachtet werden. Das Ziel ist falsch bestimmte Korrespondenzen nicht in die Transformationsbestimmung einfließen zu lassen. Dies ist von besonderer Wichtigkeit bei Randpunkten, die keine korrekte Korrespondenzen in der Zielpunktwolke aufweisen können. Methoden in diesem Bereich nutzen, wie die Wichtungsmethoden, geometrische Eigenschaften, wie die Distanz und die Normalvektoren der Punktpaare. Im Folgenden wird eine Auswahl dieser Verfahren beschrieben und in Abbildung 5.2 illustriert.

### 5.3.1 Analyse verschiedener Verwerfungsmethoden

#### Verwerfung durch die Distanz

Die simpelste Methode zum Verwerfen von falschen Korrespondenzen basiert auf der Festlegung eines Schwellenwertes für den maximalen euklidischen Abstand, den zwei korrespondierende Punkte aufweisen dürfen (siehe 5.2a). Turk et al. [TL94] geben an, dass eine gute Registrierung von zwei Punktwolken eines Objektes, beispielsweise einer Vase, erfolgen kann, wenn der Schwellenwert auf das Doppelte des Abstands zweier benachbarter Punkte gesetzt wird. In [MSY96] wird der Schwellenwert durch ein Vielfaches der Standardabweichung der Distanzen bestimmt. Pulli et al. [Pul99] nutzen eine ähnliche Methode, welche die Korrespondenzen mit den  $n\%$  weitesten Distanzen herausfiltern (siehe 5.2b). Schütz et al. [SJH97] schlagen eine weitere Heuristik vor. Zunächst wird von den zu registrierenden Punktwolken der Schwerpunkt berechnet. Anschließend werden Punktpaare, deren Distanz viel größer als der Abstand der beiden Schwerpunkte ist, als Ausreißer deklariert. Dafür wird zur Klassifizierung auf den Abstand der beiden Schwerpunkte ein empirisch ermittelter Wert addiert. Bei diesen Methoden tritt ebenso wie bei der Wichtung durch den euklidischen Abstand die Problematik auf, dass korrekte korrespondierende Punkte einen größeren Abstand aufweisen, umso weiter sie vom Laserscanner entfernt sind. In dieser Arbeit findet jedoch das Schwellenwert-Verfahren Anwendung, um sehr weit auseinanderliegende Punktpaare zu verwerfen.

#### Verwerfung von mehrfachen Korrespondenzen

Wie bereits beschrieben, wird zum Finden von Korrespondenzen jedem Punkt in der Eingangspunktwolke ein Punkt in der Zielpunktwolke zugewiesen. Dabei besteht die Möglichkeit, dass einem Punkt der Zielpunktwolke mehrere Punkte der Eingangspunktwolke zugewiesen werden. Zinßer et al. [ZSN03b] verhindern dies, indem nur das Punktpaar erhalten bleibt, welches die kürzeste Distanz zum Zielpunkt aufweist

(siehe 5.2c). Die restlichen Korrespondenzen werden verworfen. Diese Methode soll vor allem bei der Registrierung von Punktwolken, die sich nur teilweise überlappen, die Robustheit des ICP verstärken. Die Autoren geben aber als Nachteile an, dass sowohl das Erreichen der Konvergenz deutlich verlangsamt wird, als auch das der Beweis auf Konvergenz von Besl und McKay [BM92] nicht mehr gilt. Demnach kann es vorkommen, dass der Algorithmus nicht konvergiert. Aufgrund dieser Nachteile wurde dieser Ansatz in dieser Arbeit verworfen.

### Verwerfung durch die Normalvektoren

Pulli et al. [Pul99] nutzen eine weitere Technik, die ebenfalls auf ähnliche Weise bei den Wichtungsmethoden vorhanden ist. Hierbei werden Korrespondenzen verworfen, deren Normalvektoren einen größeren Winkel als  $45^\circ$  aufweisen (siehe 5.2d). Dieses Verfahren findet aufgrund derselben Nachteile, die die Wichtung durch Normalvektoren hat, keine Anwendung.

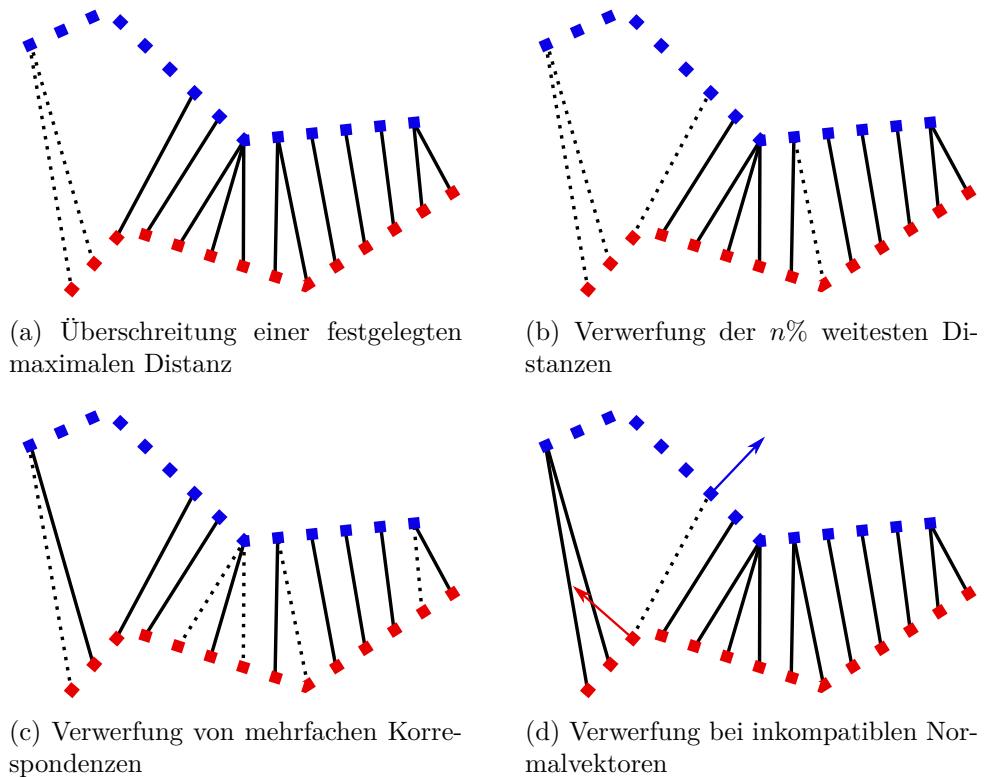


Abbildung 5.2: Darstellung mehrerer Methoden zum Verwerfen von Korrespondenzen. Verworfene Punktpaare sind als gepunktete Linie gekennzeichnet.

### Verwerfung durch RANSAC

Die Random Sample Consensus (RANSAC) Methode ist ein iteratives Verfahren zum Angleichen eines Modells an experimentelle Daten, die einen großen Anteil

an Fehlern aufweisen [FB81]. Dieses nicht-deterministische Verfahren kann jedoch auch zum Ermitteln von schlechten Korrespondenzen verwendet werden. In jeder Iteration werden zufällig drei nicht-kollineare Punkte der Eingangspunktwolke ausgewählt. Nun wird eine Transformation dieser drei Punkte zu ihren korrespondierenden Punkten in der Zielpunktwolke berechnet. Anschließend wird die ermittelte Transformation auf jeden Punkt der Eingangspunktwolke angewendet und die neue Distanz zu dem jeweiligen korrespondierenden Punkt bestimmt. Dieser Punkt wird nun als Ausreißer klassifiziert, wenn die Distanz einen festgelegten Schwellenwert überschreitet. Nachdem alle Iterationen abgeschlossen sind, wird die Transformation mit den wenigsten Ausreißern als beste Repräsentation ausgewählt. Demnach werden alle mit dieser Transformation ermittelten Ausreißer verworfen.

Dieses Verfahren nimmt ebenfalls eine Verwerfung mittels der Distanz vor und eine Steigerung der Rechenzeit ist zu erwarten, da zwar die Korrespondenzen erhalten bleiben aber alle Distanzen in jeder Iteration neu berechnet werden müssen. Erste Tests mit dem in Abschnitt 4.1.3 verwendeten Datensatz vom DLR-Gelände ergaben eine Reduktion der Laufzeit des ICP von 300 - 500 ms, was eine sehr gute Verbesserung zum Erreichen der Anforderung zur nahen Echtzeit darstellt. Dies liegt vor allem an der Reduktion der Anzahl der Punkte für die Berechnung der Fehlermetrik. Durch das Verwenden des RANSAC benötigt der ICP durchschnittlich 1 - 2 Iterationen weniger. Eine Möglichkeit der Verbesserung der Zuverlässigkeit des RANSAC wäre die Distanzen nach der Entfernung zum Sensor zu gewichten. Ebenfalls könnte die Verwendung der Winkeldistanz zwischen Korrespondenzen zur verbesserten Ausreißerermittlung verwendet werden. Jedoch ist hierbei eine Kombination mit der euklidischen Distanz notwendig, um weit voneinander entfernte Korrespondenzen nicht mit einzubeziehen.

## Fazit

In dieser Arbeit finden zwei Methoden zur Verwerfung von Ausreißern Anwendung. Der RANSAC Algorithmus führt zu einer Beschleunigung des ICP, indem die Anzahl an Iterationen verringert werden. Zudem werden Punktpaare verworfen, deren Abstand zueinander größer als ein festgelegter Schwellenwert ist. Dieser Wert wurde in dieser Arbeit auf 5 m festgelegt, damit grobe Ausreißer entfernt werden.

### 5.3.2 RANSAC

Beim Verwerfen von falschen Korrespondenzen findet der **RANSAC** in jeder Iteration des **ICP** Anwendung. Hierfür ist es notwendig bei der Initialisierung zwei Parameter zu setzen. Diese sind die Anzahl an Iterationen  $k$  und der Schwellenwert  $s$ , der die Distanz bestimmt, ab der ein Punkt als Ausreißer gewertet wird. Anhand probabilistischer Berechnungen kann durch [Formel 5.5](#) berechnet werden wie viele Iterationen benötigt werden, um mit einer Wahrscheinlichkeit  $P$  wenigstens eine fehlerfreie Kombination von Punkten zu erhalten [FB81].

$$k = \frac{\log(1 - P)}{\log(1 - (1 - \epsilon)^m)} \quad (5.5)$$

Hierbei ist  $m$  die Anzahl an Punkten die benötigt werden um die Transformation zu bestimmen. Bei der vorliegenden Transformation werden für die 6 Freiheitsgrade (3 Translation und 3 Rotation) mindestens 3 nicht-kollineare Punkte verwendet. Das  $\epsilon$  steht für die Wahrscheinlichkeit beim zufälligen Bestimmen der Punkte einen Ausreißer zu wählen, was gleichzeitig bedeutet, dass  $(1 - \epsilon)^m$  die Wahrscheinlichkeit drei valide Punkte zu erhalten widerspiegelt. Da das genaue Verhältnis von Ausreißern zu validen Punkten jedoch schwer zu bestimmen ist, wird dieser Wert in dieser Arbeit provisorisch auf 0,5 gesetzt. Dabei ist anzumerken, dass die Veränderung dieses Wertes weiteres Potential zur zeitlichen Optimierung bietet. Für  $P$  wurde 0,999 festgelegt was eine Anzahl von 52 Iterationen ergibt (siehe [Formel 5.6](#)). Es findet demnach ein Kompromiss zwischen der Anzahl der Iterationen bzw. der Berechnungszeit und der Wahrscheinlichkeit einer korrekten Lösung statt.

$$k = \frac{\log(1 - 0,999)}{\log(1 - (1 - 0,5)^3)} \approx 51.7313 \quad (5.6)$$

Der Schwellenwert  $s$ , der zur Bestimmung von Ausreißer verwendet wird, muss problemspezifisch gewählt werden. Bei der vorliegenden Aufgabenstellung wird dieser Wert anhand des durch den Octree eingeführten Diskretisierungsgrads bestimmt. Bei einer Voxelgröße von 0,5 cm ist ein Punkt nach einer korrekten Transformation theoretisch nicht weiter als 0,25 cm von seinem korrespondierenden Punkt entfernt. Zusätzlich muss ein Laserscannerspezifisches Rauschen angenommen werden, daher wird diesem Wert eine leicht höhere Zahl zugewiesen. Nun werden anhand des Schwellenwerts Punkte aussortiert, die entweder grobe Ausreißer durch das Scannen (siehe [Abschnitt 2.1.1](#)) darstellen oder keine korrespondieren Punkte in der globalen Karte besitzen, da sie zu neuen unbekannten Bereichen gehören.

Nach der Bestimmung der Parameter, iteriert der **RANSAC** Algorithmus über folgende Schritte  $k$  mal:

- Zufälliges Auswählen von drei verschiedenen, nicht-kollinearen Punkten aus der Eingangspunktwolke. Um eine akzeptable Transformation zu erhalten, sollten die gewählten Punkte einen gewissen Abstand zueinander aufweisen. Dafür wird zunächst durch [Formel 5.7](#) ein weiterer Schwellenwert  $w$  bestimmt. Anhand einer Hauptkomponentenanalyse (siehe [Abschnitt 2.3.3](#)) werden die Hauptrichtungen der Eingangspunktwolke bestimmt. Die Varianzen der Hauptrichtungen sind dabei durch den jeweils korrespondierenden Eigenwert  $\lambda$  gegeben. Demnach ist die Standardabweichung als Wurzel des Eigenwerts definiert. Als Schwellenwert wird nun die mittlere Standardabweichung der Hauptrichtungen berechnet. Dieser Wert wird quadriert, da die Vergleiche mit quadrierten Distanzen stattfinden. Bei der Verwendung von Punkten im 3D Raum ist  $n = 3$  bestimmt.

$$w = \left( \frac{1}{n} \sum_{i=0}^n \sqrt{\lambda_i} \right)^2 \quad (5.7)$$

Unterschreitet mindestens eine der quadrierten Distanzen zwischen drei zufällig ausgewählten Punkten diesen Schwellenwert, wird eine neue Kombination bestimmt.

- Mit Hilfe der Singulärwertzerlegung (siehe folgenden Abschnitt) werden die Transformationsparameter bestimmt, welche die drei Punkte ihren korrespondierenden Punkten angleichen.
- Bestimmen der Anzahl  $v$  an validen Punkten (siehe [Formel 5.8](#)), die eine geringere Distanz zu ihrem korrespondierenden Punkt aufweisen als  $s$ . Um diesen Vorgang zu optimieren, wird hier die quadrierte Distanz zwischen den Punkten zu  $s^2$  verglichen. Wird demnach  $d^2(p, q) < s^2$  erfüllt, handelt es sich um einen validen Punkt.

$$v = \sum_{i=0}^n \begin{cases} 1 & \text{falls } d^2(p_i, q_i) < s^2 \\ 0 & \text{sonst} \end{cases} \quad (5.8)$$

- Nun werden die validen Punkte gespeichert, falls  $v > v_{max}$  erfüllt wird, wobei  $v_{max}$  die größte bisher gefundene Anzahl an validen Punkten repräsentiert.

Nach dem Abschluss des [RANSAC](#) Algorithmus, wird im [ICP](#) nur mit den zu  $v_{max}$  gehörigen validen Punkten weiter verfahren, so dass die restlichen Punkte verworfen werden.

## 5.4 Auswahl der zu minimierenden Fehlermetrik

Das Hauptelement des ICP ist die Minimierung einer Fehlermetrik, da durch diesen Vorgang die geforderte Transformation schrittweise angenähert wird. Wie in Abschnitt 3.1 erwähnt, ist die Point-to-Plane Fehlermetrik eine Alternative zu der klassischen Verwendung der Distanz zum nächsten Nachbarn. Obwohl das in dieser Arbeit verwendete Covariance Sampling (siehe Abschnitt 4.1.4) auf dieser Fehlermetrik basiert, sind erste Registrierungen weniger erfolgreich verlaufen als mit der Point-to-Point Metrik.<sup>1</sup>

Daher findet in dieser Arbeit die klassische Point-to-Point Fehlermetrik Anwendung, für die analytisch geschlossene Lösungen vorhanden. Diese basieren auf der Singulärwertzerlegung (SWZ) [AHB87], Quaternions [Hor87], orthonormalen Matrizen [HHN88] und Dual Quaternions [WSV91]. Eggert et al. [ELF97] haben einen Vergleich dieser vier Methoden aufgestellt und sind zu dem Ergebnis gekommen, dass die Unterschiede in der numerischen Genauigkeit sehr gering ausfallen. Bei der Stabilität schneidet die Singulärwertzerlegung knapp vor der Quaternion-Methode am besten ab. Daher wird die SWZ-Methode in dieser Arbeit verwendet und im folgenden Abschnitt genauer betrachtet.

### 5.4.1 Singulärwertzerlegung

Wie in Abschnitt 5.4 erwähnt, sind vier verschiedene Verfahren zur Minimierung der Point-to-Point Fehlermetrik (siehe Formel 5.9) in geschlossener Form vorhanden. Im Folgenden wird die Herleitung der Fehlerfunktion und anschließend die Verwendung der Singulärwertzerlegung von Arun et al. [AHB87] genauer erläutert.

$$E(\mathbf{R}, \vec{t}) = \frac{1}{N} \sum_{i=1}^N \|q_i - (\mathbf{R}p_i + \vec{t})\|^2 \quad (5.9)$$

Hierbei stehen  $p_i$  und  $q_i$  jeweils für die korrespondierenden Punkte aus der Eingangspunktwolke  $P$  und Zielpunktwolke  $Q$ . Mit  $N$  wird die Anzahl der Korrespondenzen angegeben. Die zur Registrierung notwendige Transformation wird durch die Rotationsmatrix  $\mathbf{R}$  und den Translationsvektor  $\vec{t}$  dargestellt. Bei dem SWZ Verfahren kann die Berechnung der Rotation und Translation getrennt werden. Hierfür wird, wie bei der PCA, der Schwerpunkt von  $P$  und  $Q$  berechnet

$$c_q = \frac{1}{N} \sum_{i=1}^N q_i \quad c_p = \frac{1}{N} \sum_{i=1}^N p_i \quad (5.10)$$

---

<sup>1</sup>Ein Vergleich der Ergebnisse ist in Abschnitt B.3 dargestellt

und von allen  $p_i$  und  $q_i$  subtrahiert.

$$q'_i = q_i - c_q \quad \forall q_i \in Q \quad p'_i = p_i - c_p \quad \forall p_i \in P \quad (5.11)$$

Anschließend haben beide Punktmengen den Ursprung als Schwerpunkt. Somit kann nach [AHB87] die Formel 5.9 entkoppelt und  $\mathbf{R}$  durch folgende Minimierung gefunden werden.

$$E(\mathbf{R}, \vec{t}) = \sum_{i=1}^N \|q'_i - (\mathbf{R}p'_i)\|^2 \quad (5.12)$$

Nachdem  $\mathbf{R}$  ermittelt wurde, ist es möglich  $\vec{t}$  zu bestimmen.

$$\vec{t} = c_q - \mathbf{R}c_p \quad (5.13)$$

Für die Berechnung der Rotation wird nun die SWZ verwendet, wobei  $\mathbf{R}$  als orthonormale  $3 \times 3$  Matrix definiert ist. Die Spalten von orthonormalen Matrizen sind orthogonal zueinander und entsprechen der Länge 1. Zunächst wird eine Matrix  $\mathbf{H}$  aufgestellt.

$$K(x, y) = \sum_{i=1}^N q'_{ix} p'_{iy} \quad (5.14)$$

$$\mathbf{H} = \begin{pmatrix} K(x, x) & K(x, y) & K(x, z) \\ K(y, x) & K(y, y) & K(y, z) \\ K(z, x) & K(z, y) & K(z, z) \end{pmatrix} \quad (5.15)$$

Mit dieser Matrix wird nun die Singulärwertzerlegung  $\mathbf{H} = \mathbf{U}\Lambda\mathbf{V}^T$  durchgeführt. Hierbei sind  $\mathbf{U}$  und  $\mathbf{V}$  Orthogonalmatrizen und  $\Lambda$  eine Diagonalmatrix, also eine Matrix bei der nur die Einträge auf der Hauptdiagonalen ungleich null sind. Zunächst werden beide Seiten mit  $\mathbf{H}^T = \mathbf{V}\Lambda^T\mathbf{U}^T$  multipliziert

$$\mathbf{H}^T \mathbf{H} = \mathbf{V}\Lambda^T\mathbf{U}^T\mathbf{U}\Lambda\mathbf{V}^T \quad (5.16)$$

Da  $\mathbf{U}$  orthogonal ist, wird aus  $\mathbf{U}^T\mathbf{U}$  die Einheitsmatrix und da  $\Lambda$  diagonal ist, entsteht  $\Lambda^2$  aus  $\Lambda^T\Lambda$ .

$$\mathbf{H}^T \mathbf{H} = \mathbf{V}\Lambda^2\mathbf{V}^T = \mathbf{V} \begin{pmatrix} \sigma_1^2 & & \\ & \sigma_2^2 & \\ & & \sigma_3^2 \end{pmatrix} \mathbf{V}^T \quad (5.17)$$

Die Spalten von  $\mathbf{V}$  sind nun die Eigenvektoren von  $\mathbf{H}^T \mathbf{H}$  und die Werte  $\sigma_i^2$  die Eigenwerte von  $\mathbf{H}^T \mathbf{H}$ . Nun kann entweder nach  $\mathbf{U}$  aufgelöst oder dasselbe Verfahren mit  $\mathbf{H} \mathbf{H}^T$  durchgeführt werden. [Str09]

Die optimale Rotation ist nun durch  $\mathbf{R} = \mathbf{V} \mathbf{U}^T$  gegeben wenn  $\det(\mathbf{R}) = 1$  oder  $\det(\mathbf{R}) = -1$  und einer der Singulärwerte von  $\mathbf{H}$  null ist gilt. Andernfalls schlägt der Algorithmus fehl, da weder  $P$  noch  $Q$  koplanar sind. Dieser Fall trete laut Arun et al. nur sehr selten auf. Wie in Formel 5.13 beschrieben kann nun die Translation bestimmt werden. Der Beweis für die Anwendung der SWZ befindet sich in Abschnitt A.2.

# 6. Evaluierung

In diesem Kapitel findet eine Evaluierung des entwickelten Ansatzes statt. Dafür wird im folgenden Abschnitt zunächst der Versuchsaufbau vorgestellt. In [Abschnitt 6.2](#) werden Maße zur Bewertung der Ähnlichkeit zweier Octrees bestimmt, mit denen anschließend eine Auswertung der Versuchsdaten durchgeführt wird.

## 6.1 Versuchsaufbau

Der Versuch wurde mit einem bodengebundenen Sensorträger durchgeführt. Wie in [Abbildung 6.1](#) illustriert, wurde hierfür sowohl der midiARTIS, ein unbemannter Forschungshelikopter des [DLR](#), als auch ein Laserscanner fest auf einen Sensorträger montiert.

Der midiARTIS basiert auf einem Kunstflughelikopter des Typs Genesis 1800 der Firma Ludwig Benda GmbH. Um dem Helikopter autonomes Manövrieren zu ermöglichen, verfügt er über einen Bordrechner, der für die Navigation und Regelung zuständig ist. Außerdem ist der Hubschrauber mit einem [GPS](#)-Empfänger und einer Inertialplattform ausgestattet. Per [WLAN](#) und Funk ist es möglich mit dem midiARTIS zu kommunizieren. Alle Komponenten sind unter anderem wegen ihrer geringen Größe und niedrigen Stromverbrauchs ausgewählt und unterhalb des Helikopters angebracht worden.

Der Novatel OEM4 [GPS](#)-Empfänger kann in verschiedenen Modi betrieben werden. Während der Empfänger mit Hilfe der L1 Frequenz auf eine Genauigkeit von 1,8 m Kreisfehlerwahrscheinlichkeit ([CEP](#)) und in Kombination mit L2 auf 1,5 m [CEP](#) kommt, wird im [DGPS](#)-Modus eine Genauigkeit von 0,45 m [CEP](#) erreicht [[Nov05](#)]. Die [ISIS-IMU](#), ein Produkt der Inertial Science Inc, ist ein Inertialsensor mit sechs Freiheitsgraden und ist ausgelegt für mittlere Genauigkeit und dynamische Umge-

bungen [Ine99]. Anhand der technischen Spezifikationen lässt sich die ISIS-IMU in die Automotive-Klasse der Inertialplattformen einordnen (siehe Abschnitt 2.1.3). Der verwendete Laserscanner Velodyne High Definition LiDAR (**HDL-32E**) arbeitet mit 32 simultan-operierenden Laserstrahlen mit einer Wellenlänge von 905 nm. Die Laserstrahlen sind so angeordnet, dass sie ein vertikales Sichtfeld von +10° bis -30° ermöglichen. Weiterhin verfügt der mit einem rotierenden Kopf ausgestattete Scanner über ein horizontales Sichtfeld von 360° und kann demnach etwa ein Neuntel einer Kugelfläche der Umgebung erfassen. Während dieses Versuchs wird das Sichtfeld durch den Hubschrauber und den Sensorträger leicht eingeschränkt. Der **HDL-32E** generiert bis zu 700.000 Punkte pro Sekunde mit einer Reichweite bis zu 100 Metern und einer Genauigkeit von +/- 2 cm bei einer Wiederholungsrate von 10 Hz. In Kombination mit den bereits am Hubschrauber montierten Sensoren überschreitet der 1 kg schwere Laserscanner jedoch die Traglast des midiARTIS. Deshalb wurde der **HDL-32E** bei diesem Versuch am vorderen Ende des Sensorträgers angebracht<sup>1</sup>.

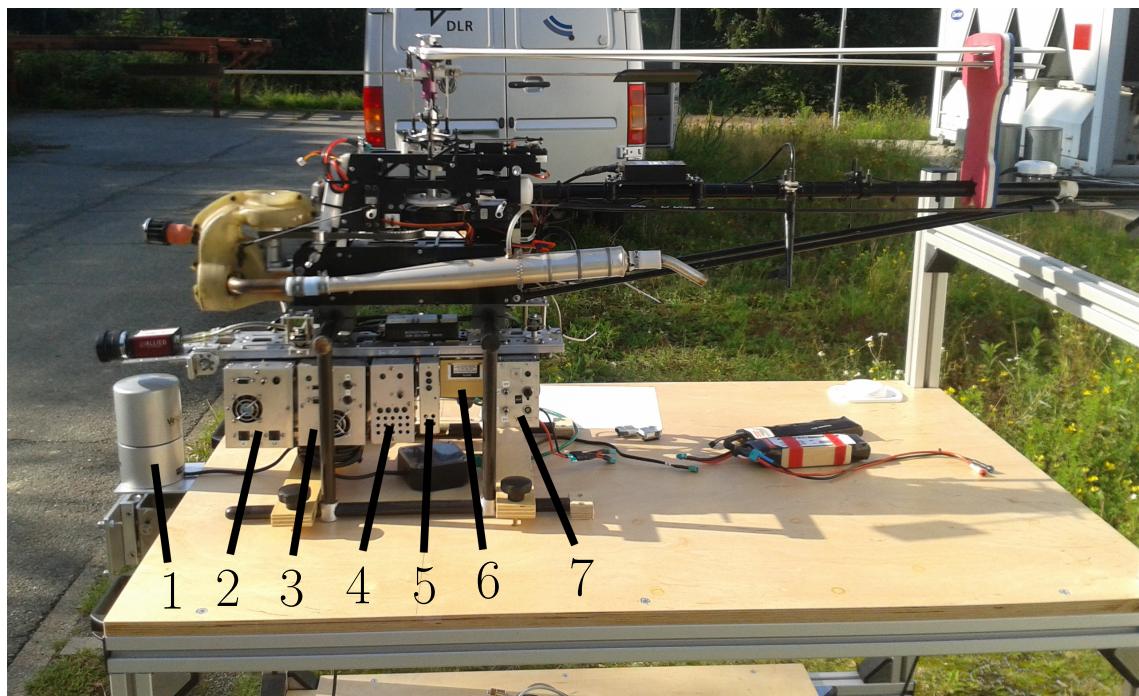


Abbildung 6.1: midiARTIS auf dem Sensorträger. 1) **HDL-32E** Laserscanner 2) Bildverarbeitungsrechner 3) Flugrechner 4) Telemetrie 5) **GPS** 6) **IMU** 7) Stromversorgung

Anschließend wurde eine ca. 50 m lange Strecke auf dem Gelände des **DLR** Braunschweig für die Versuchsdurchführung ausgewählt. Bei der Festlegung der Strecke wurde darauf geachtet, dass verschiedenartige Umgebungen erfasst werden, da die Geometrie der resultierenden Punktfolge einen Einfluss auf den **ICP** hat. Somit

<sup>1</sup>Weitere technische Informationen lassen sich dem Datenblatt des **HDL-32E** entnehmen. [Vel12]

kann eine höhere Generalisierbarkeit des verwendeten Konzepts gewährleistet werden. Die Varianz in der Umgebung liegt dabei in dem Anteil der Vegetation sowie der Beschaffenheit der urbanen Umgebung. Zwischen hohen Gebäuden kann es zudem beim GPS-Signal zu dem in Abschnitt 2.1.2 beschriebenen Mehrwegempfang kommen.

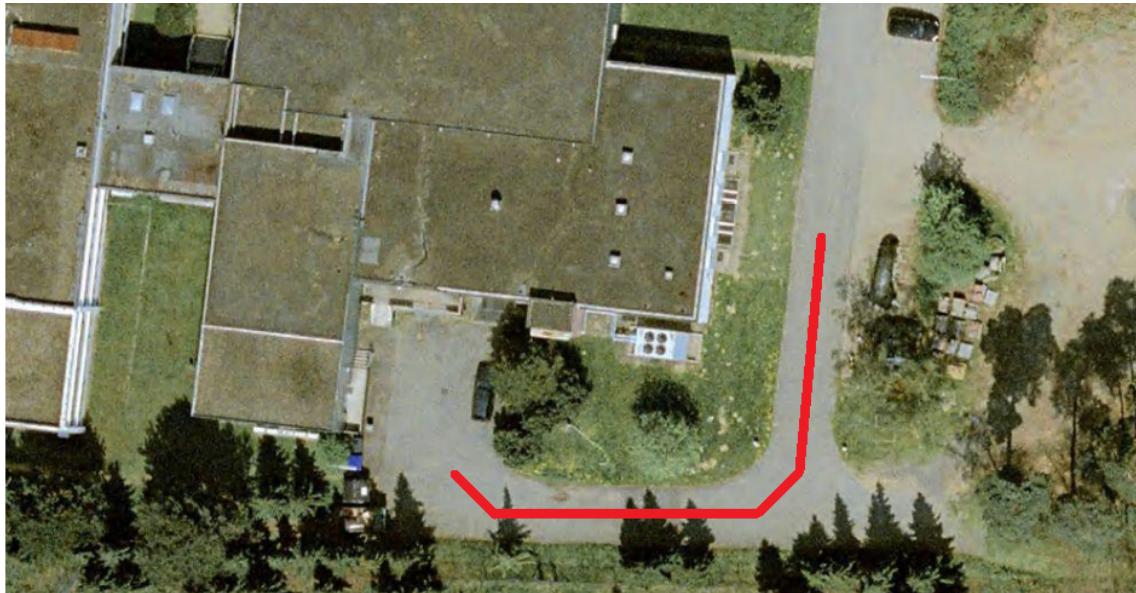


Abbildung 6.2: Versuchsstrecke auf dem DLR-Gelände<sup>2</sup>

Diese in Abbildung 6.2 dargestellte Strecke wurde dreimal mit dem Sensorträger befahren. Zur Gewinnung einer globalen Umgebungskarte (Octree), welche als Referenzobjekt dient und somit die sogenannte Ground Truth darstellt, wurde der Träger beim ersten Durchlauf mit einer Geschwindigkeit von unter 0,2 m/s bewegt. Außerdem wurde jeweils nach zehn Metern für zehn Minuten pausiert, um das DGPS-Signal zu stabilisieren und statische Scans zu erhalten, welche über die Zeit gemittelt wurden. Es wurde erwartet, dass der resultierende Octree als Vergleichsobjekt verwendet werden kann, da die probabilistisch-orientierte Belegungs- bzw. Freizeichnungsfunktion des Octrees durch die Pausen mit dem stabilen DGPS-Signal eine sehr gute Darstellung der Umgebung gewährleistet.

Beim zweiten Durchlauf wurde der Träger dieselbe Strecke mit einer konstanten Geschwindigkeit zwischen 1 - 2 m/s und ohne Pausen bewegt. Hierbei können Fehler auftreten, wenn der Empfänger den Satelliten wechselt. Außerdem hat der Mehrwegempfang einen viel größeren Einfluss. Wie in Abschnitt 2.1.2 dargelegt, kann ein RMS-Fehler von 1,2 m entstehen, der mit dem DOP-Wert multipliziert wird. Dieser Durchlauf bildet einen normalen Anwendungsfall ab. Zusätzlich wurde ein weite-

<sup>2</sup>Grafiken ©2014 AeroWest, DigitalGlobe, GeoBasis-DE/BKG, GeoContent, Kartendaten ©2014 GeoBasis-DE/BKG (©2009) Google

rer Durchlauf durchgeführt, bei dem die **GPS**-Antenne am Vehikel teilweise durch ein Signal undurchlässiges Material abgedeckt wurde. Dies hatte eine teilweise Störung des Signals zur Folge. Bei ausbleibendem **GPS**-Signal findet folglich auch keine Korrektur durch das **DGPS** statt. Dieser Versuch stellt somit einen Extremfall dar. Da auf diese Weise nur eine Störung des **GPS**-Signals generiert werden kann, wurde auf Basis des zweiten Durchlaufs ein vierter Durchlauf generiert. Generell wird der zweite Durchlauf ein weiteres mal verwendet, jedoch wurde ein synthetisches Rauschen über die durch die **IMU** bestimmten Lagewinkel gelegt. Dabei wurde ein normalverteiltes Rauschen von 0,075 Bogenmaß bei dem Gierwinkel 0,0025 Bogenmaß bei dem Rollwinkel und 0,0001 Bogenmaß bei dem Nickwinkel verwendet.

Die Werte für das Rauschen auf dem Roll- und Nickwinkel entsprechen jeweils ungefähr der Varianz aller Roll- und Nickwinkel des normalen Anwendungsfalls. Dies wurde so gewählt, da die Versuchsstrecke keine großen Höhenunterschiede aufweist und sich der Roll- und Nickwinkel sich dementsprechend wenig verändern. Da der Sensorträger eine größere Länge als Breite besitzt, ist die Varianz beim Rollwinkel durch Unebenheiten auf der Strecke größer.

Aufgrund der starken Unterschiede der Gierwinkel, ergibt eine Berechnung anhand der Varianz ein zu großes Rauschen, wodurch die Gefahr besteht, dass der **ICP** bei vielen Scandatensätzen zu einem falschen lokalen Minimum konvergiert. Daher wurde hierbei die Varianz aus den Gierwinkeln der ersten 1.000 Navigationsdaten des normalen Anwendungsfalls verwendet. Insgesamt wurden somit vor allem Ungenauigkeiten beim diesem Winkel eingeführt.

Demnach ergeben sich folgende vier Durchläufe, auf die im weiteren Verlauf mit den angegebenen Kurzformen verwiesen wird.

1. Durchlauf: Referenzdurchlauf (**RD**)
2. Durchlauf: Normaler Anwendungsfall (**ND**)
3. Durchlauf: Gestörtes **GPS**-Signal, was zu translativen Fehlern führt (**TFD**)
4. Durchlauf: Normaler Anwendungsfall mit synthetischem Rauschen auf den Lagewinkel, was zu rotatorischen Fehlern führt (**RFD**)

Abbildung 6.3 veranschaulicht den Octree des Referenzdurchlaufs mit einer Octreeauflösung von einem Meter. Die Farbe der Voxel gibt dabei die Höhe an. Die rechteckigen Formen im oberen Teil der Abbildung repräsentieren dabei die Gebäude des **DLR**. Die magentafarbenen Voxel gehören zu den umliegenden Baumkronen. Somit ist im rechten und unteren Teil der Grafik vor allem Vegetation angesiedelt. Der streifenförmige Freiraum am unteren, linken Rand stellt eine Straße außerhalb des

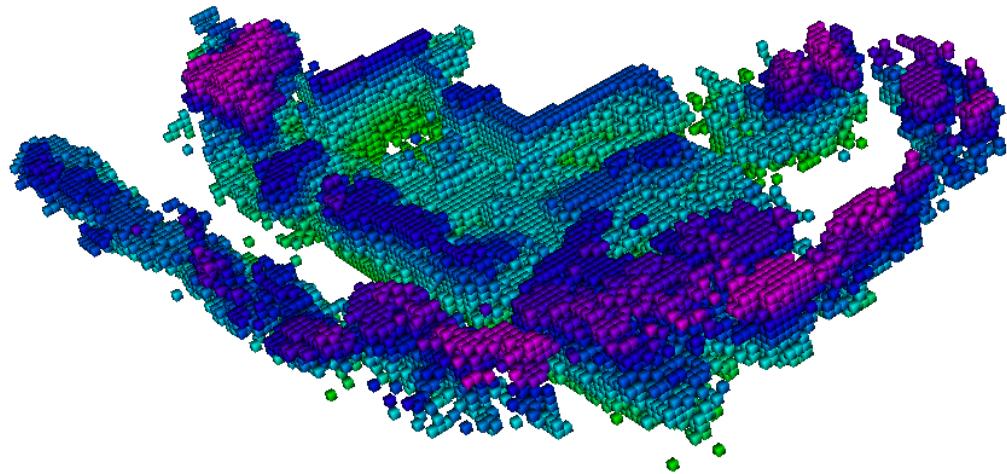


Abbildung 6.3: Referenzoctree der Versuchsstrecke mit einer Auflösung von 1 m

DLR-Geländes dar. Diese wurde aufgrund der davor liegenden Vegetation jedoch nicht durch die Laserstrahlen des Scanners erfasst. Wie zu erkennen ist, weist diese Umgebung sowohl urbane als auch vegetative Bereiche auf, weswegen diese Strecke für diesen Versuch ausgewählt wurde.

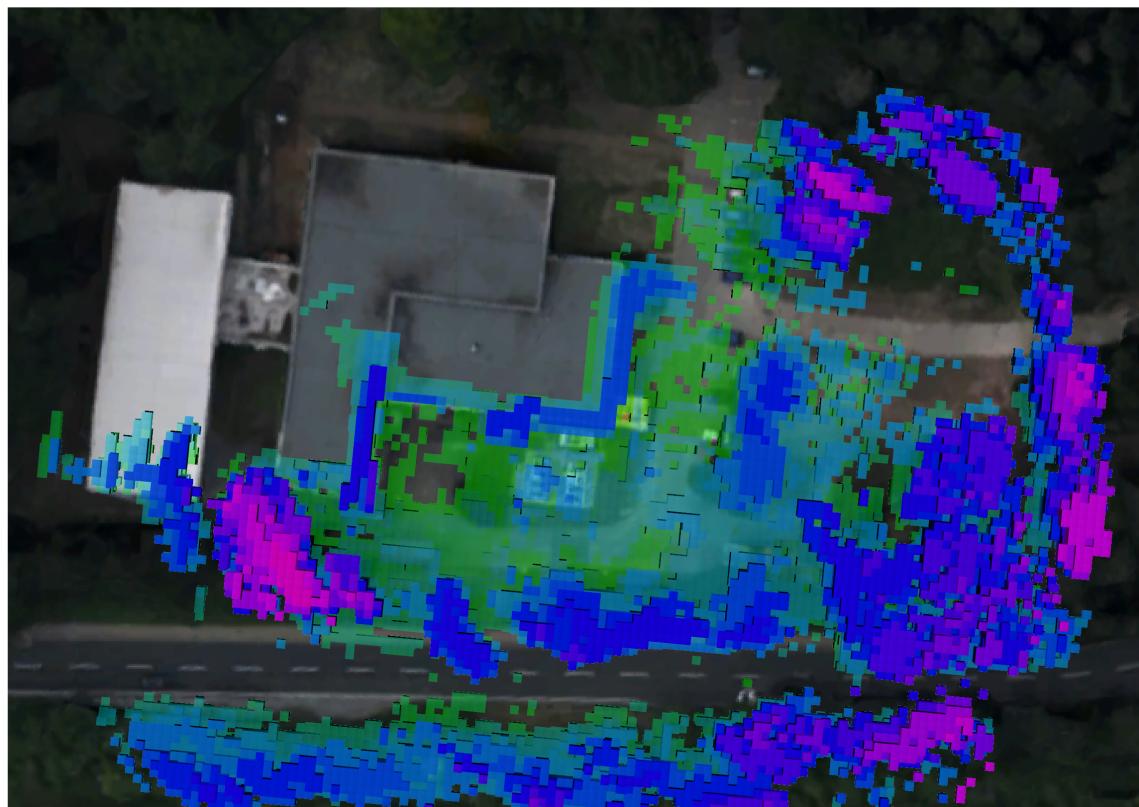


Abbildung 6.4: Referenzoctree der Versuchsstrecke im Vergleich zum Satellitenbild<sup>3</sup>

<sup>3</sup>Satellitenbild: Grafiken ©2014 AeroWest, DigitalGlobe, GeoBasis-DE/BKG, GeoContent, Kartendaten ©2014 GeoBasis-DE/BKG (©2009) Google

Zur Verdeutlichung wird in Abbildung 6.4 der Octree aus der Vogelperspektive im Vergleich zum Satellitenbild der abgebildeten Umgebung dargestellt. Besonders gut ist hierbei der Freiraum zu sehen, der durch die anliegende Straße entstanden ist. Außerdem ist die Darstellung des Bodens (grüne Voxel) in dieser Abbildung besser zu erkennen. Voxel, die sich innerhalb von Gebäuden befinden, sind durch Laserstrahlen entstanden, die auf Fensterscheiben in diesem Bereich getroffen sind.

## 6.2 Bewertungsprinzip

Während des Versuchs wurden sowohl die einzelnen Scandatensätze als auch die dazugehörigen Navigationsdaten gespeichert. Dies ergibt die Möglichkeit denselben Durchlauf mit verschiedenen Algorithmen und Parametrisierungen zu wiederholen. Beim Referenzdurchlauf werden die Scans anhand der Navigationsdaten transformiert und sukzessiv in einen Octree überführt, welcher für den jeweiligen Versuch als Vergleichsobjekt dient.

Bei den drei weiteren Durchläufen wurde zunächst dasselbe Verfahren angewandt, um anschließend den rekonstruierten Octree mit dem Referenzobjekt zu vergleichen und einen Ausgangsfehler ermitteln zu können. Anschließend wurden in den vorherigen Kapiteln beschriebenen Methoden auf die Durchläufe angewandt, mit dem Ziel den Ausgangsfehler zu verringern und sich somit der Referenzlösung anzunähern.

Zum Bestimmen des Fehlers wurden mehrere Metriken in Betracht gezogen, wobei drei zum Vergleich der Octrees ausgewählt wurden. Ein erster Ansatz ist die Beobachtung der Knoten der zu vergleichenden Octrees aus binärer Sicht. Demnach ist jeder Knoten entweder frei oder besetzt. Daher lässt sich eine Konfusionsmatrix aus Gütekriterien aufstellen.

		Referenz-Octree	
		Belegt	Frei
Rekonstruierter Octree	Belegt	True Positive	False Positive
	Frei	False Negative	True Negative

Tabelle 6.1: Konfusionsmatrix bei binärer Betrachtung der Octreeknoten

True Positive (**TP**) Fälle treten auf wenn Knoten, die in beiden Octrees an derselben Position im kartesischen Koordinatensystem positioniert sind, als besetzt gekennzeichnet sind. Hingegen ist False Positive (**FP**) der Fall wenn ein Knoten, der im rekonstruierten Octree belegt ist, jedoch nicht im Referenzoctree. Demnach ist der Knoten im rekonstruierten Octree fälschlicherweise belegt. Gegenteilig beschreibt False Negative (**FN**) den Fall, dass ein Knoten zwar im Referenzoctree besetzt ist,

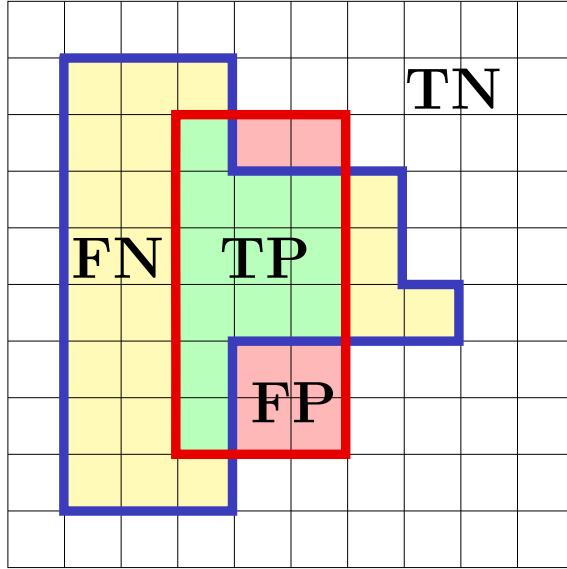


Abbildung 6.5: Darstellung der vier Gütekriterien. Die belegten Knoten des Referenzoctree sind blau und die des rekonstruierten Octrees rot umrandet

im rekonstruierten Octree aber nicht. Letztendlich definiert True Negatives (**TN**) die Fälle bei denen die Knoten in beiden Octrees frei sind.

Nach dem Bestimmen dieser Gütekriterien, lassen sich verschiedene Maße zur Beurteilung der Klassifikation aufstellen. Zwei dieser Maße sind die Precision Rate (**PR**) und Recall Rate (**RR**), die durch folgende Formeln berechnet werden.

$$PR = \frac{TP}{TP + FP} \quad (6.1)$$

$$RR = \frac{TP}{TP + FN} \quad (6.2)$$

Die Recall Rate, die auch als Sensitivität bezeichnet wird, gibt das Verhältnis der korrekt klassifizierten positiven Fälle zu der Gesamtheit der positiven Fälle an. Die Precision Rate hingegen berechnet den Anteil der korrekt klassifizierten positiven Fälle an der Gesamtheit der positiv klassifizierten Fälle. Da diese beiden Raten sich gegenseitig beeinflussen, wurden verschiedene Maße entwickelt, welche die Gesamtgenauigkeit wiedergeben sollen. Zu den bekanntesten Gütemaßen dieser Art zählen die F-Maße, welche das harmonische Mittel von **RR** und **PR** bilden. Weitere Maße berechnen das arithmetische oder geometrische Mittel. Der Nachteil liegt vor allem darin, dass die True Negatives vollständig außer Acht gelassen werden. Daher sind beim Bewerten der Performanz von binärer Klassifikation andere Gütemaße, wie Matthews Korrelationskoeffizient (**MCC**) vorzuziehen. [Pow01]

Der **MCC** ist ein ausbalanciertes Maß, das auch bei Klassen mit stark unterschiedlichen Größenverhältnissen anwendbar ist. Dabei liegt der Korrelationskoeffizient im-

mer zwischen -1 und +1. Während ein Wert von +1 eine optimale Übereinstimmung bedeutet, steht -1 für totale Ungleichheit. Ein Wert 0 sagt aus, dass der Vergleich nicht besser als eine zufällige Prädiktion ist. [BBC<sup>+00</sup>]

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}} \quad (6.3)$$

Dieses Gütemaß wird in dieser Arbeit zur ersten Bewertung der Gleichheit zweier Octrees verwendet, obwohl es zwei Aspekte vernachlässigt. Zum einen wird der Simplizität halber angenommen, dass die Knoten der Octrees binärer Natur sind, also entweder besetzt oder frei. Die Knoten der Octrees besitzen jedoch eine Wahrscheinlichkeit mit der sie belegt sind. Ist ein Knoten im Referenzoctree mit einer Wahrscheinlichkeit von 90% und im rekonstruierten Octree zu 70% belegt, ist dies bei der Benutzung der binären Klassifikation optimal. Optimalität wäre jedoch erreicht, wenn der Knoten in beiden Octrees dieselbe Wahrscheinlichkeit aufweist.

Bei probabilistischen Octrees ([HWB<sup>+13</sup>]) und Belegheitsgittern ([MB12] [Cha07]) wird unter anderem die Kullback-Leibler Divergenz (**KLD**) als Maß für die Ähnlichkeit genutzt.

$$KLD = \frac{1}{n} \sum_{i=1}^n \log \left( \frac{p_i}{q_i} \right) \cdot p_i + \log \left( \frac{1-p_i}{1-q_i} \right) \cdot (1-p_i) \quad (6.4)$$

$p_i$  und  $q_i$  stehen für die Belegtheitswahrscheinlichkeit eines Knotens im rekonstruierten Octree respektive Referenzoctree. Dabei haben beide Knoten dieselbe Position im Weltkoordinatensystem. Die **KLD** bestimmt demnach die Differenz zwischen zwei Wahrscheinlichkeiten, wobei der Wert der **KLD** 0 entspricht wenn die beiden Wahrscheinlichkeiten identisch sind. Problematisch ist jedoch, dass die Kullback-Leibler Divergenz asymmetrisch ist. Werden  $p_i$  und  $q_i$  vertauscht ergibt die Divergenz einen anderen Wert, was bei einem Vergleich nicht optimal ist.

Daher wurde in dieser Arbeit als zweites Maß die Jensen-Shannon Divergenz (**JSD**) verwendet, die eine symmetrische und geglättete Variante der **KLD** darstellt [Lin91]. Die Symmetrie wird durch die Verwendung des Mittelwerts  $m_i$  von  $p_i$  und  $q_i$  erreicht.

$$JSD = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \cdot \left( p_i \cdot \log_2 \left( \frac{p_i}{m_i} \right) \right) + \frac{1}{2} \cdot \left( q_i \cdot \log_2 \left( \frac{q_i}{m_i} \right) \right) \quad (6.5)$$

Durch die Nutzung des Logarithmus zur Basis 2 wird sichergestellt, dass die Divergenz durch 0 und 1 begrenzt ist, wobei 0 eine komplette Übereinstimmung bedeutet. Sind Knoten in einem der beiden Octrees nicht vorhanden wird die Wahrscheinlichkeit auf 0,5 gesetzt. Dies bedeutet, dass der Voxel weder belegt noch frei sondern unbekannt ist.

Die beiden bisher gewählten Maße berücksichtigen die geometrische Lage der Knoten nicht. Der Fehler, der durch einen fälschlicherweise belegten Knoten auftritt, ist umso schwerwiegender je weiter der Abstand zu einem korrekt belegten Knoten ist. Die naheliegenste Metrik ist die Point-to-Point Metrik, die der ICP als zu minimierende Fehlerfunktion verwendet. Ein Nachteil dieser Metrik ist, dass die Distanz nur in eine Richtung gemessen wird. Bei zwei kompakten Mengen  $A$  und  $B$  wird nur der Abstand der Punkte in  $A$  zu den nächstgelegenen Punkten in  $B$  summiert. Die gegenseitige Richtung von  $B$  nach  $A$  wird nicht berücksichtigt.

Die klassische Hausdorff-Distanz (siehe Abbildung 6.6) verwendet hingegen beide Richtungen um Festzustellen inwiefern sich zwei Mengen überschneiden. Zunächst wird ebenfalls für jeden Punkt in  $A$  der nächstgelegene Punkt in  $B$  ermittelt, wobei nur der maximale Abstand festgehalten wird. Mit demselben Verfahren wird der weiteste Abstand von  $B$  zu  $A$  ermittelt. Die Hausdorff-Distanz entspricht nun dem Maximum dieser beiden Werte. [Hau78] In der Abbildung 6.6 sind die weitesten Distanzen in beide Richtungen mit einem vergrößerten Pfeil markiert. Da der maximale Abstand bei 6.6b größer ist als bei 6.6a ist die Hausdorff-Distanz gleich dem weitesten Abstand von  $B$  zu  $A$ . Diese Metrik ist sehr anfällig gegenüber Ausreißern, da ein weit entfernter Punkt die Hausdorff-Distanz bestimmt. Beim Verwenden von Laserscannern können, wie in Abschnitt 2.1.1 beschrieben, unzuverlässigen Datenpunkten entstehen. Da der Abstand zu diesen Ausreißern sich nicht als Qualitätsmetrik zum Bewerten des Octrees eignet, wird in dieser Arbeit eine Variante der Hausdorff-Metrik verwendet.

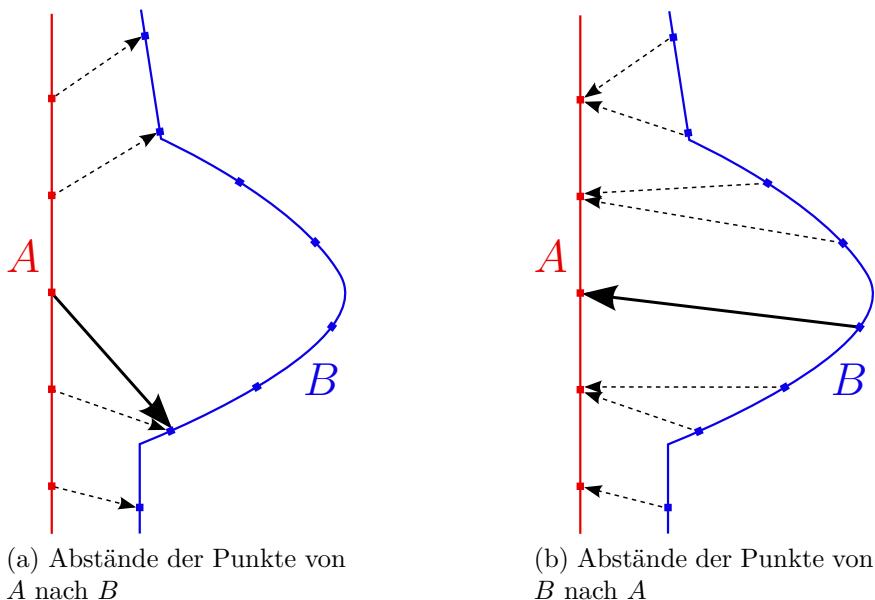


Abbildung 6.6: Beispiel der Hausdorff-Distanz. Die weiteste Distanz in beide Richtungen ist jeweils durch einen größeren Pfeil gekennzeichnet.

Die durchschnittliche Hausdorff-Distanz (**DHD**) berechnet den Mittelwert aller Distanzen in beiden Richtungen und berücksichtigt somit mehr geometrische Beziehungen zwischen den beiden Datensätzen [ZZ09]. Diese Metrik ist somit gegeben durch

$$DHD(A, B) = \frac{\sum_{a \in A} \min_{b \in B}(d(a, b)) + \sum_{b \in B} \min_{a \in A}(d(b, a))}{|A| + |B|} \quad (6.6)$$

## 6.3 Auswertung

Zum Vergleich von zwei Octrees wurde eine Auswertung anhand der drei gewählten Maße **MCC**, **JSD** und **DHD** vorgenommen. Diese Auswertung wird in diesem Abschnitt vorgestellt. Aufgrund der Einführung einer Zufallskomponente durch den **RANSAC** Algorithmus ist der **ICP** nicht mehr deterministisch. Daher wurden die Durchläufe jeweils zehnmal wiederholt, um ein aussagekräftiges Ergebnis zu erhalten. Zunächst wird eine Octreeauflösung von 0,5 m verwendet, da diese der Standardauflösung bisheriger Versuche der Arbeitsgruppe entspricht. Bei der Auswertung werden zunächst die prozentualen Veränderungen gegenüber dem Ausgangsfehler beschrieben. Anschließend werden die Ergebnisse in einem Fazit genauer betrachtet. In [Abschnitt 6.3.1](#) werden Tests mit weiteren Auflösungen durchgeführt, um der Aufgabenstellung entsprechend die Genauigkeitsunterschiede analysieren zu können. Nach dieser Bewertung wird untersucht wie viel Zeit der entwickelte Ansatz zur Registrierung benötigt und inwieweit die, in [Abschnitt 1.2](#) definierte, Anforderung zur nahen Echtzeit eingehalten werden kann.

### Matthews Korrelationskoeffizient

Der Boxplot in [Abbildung 6.7](#) zeigt die prozentualen Veränderungen des **MCC** in den drei Durchläufen in Bezug auf den Ausgangsfehler. Dabei wird die sogenannte Fünf-Punkte-Zusammenfassung verwendet, die 1977 von Tukey [[Tuk77](#)] eingeführt wurde. Demnach wird eine Datenreihe durch den minimalen und maximalen Wert, den Median sowie das untere und obere Quartil charakterisiert. Dabei trennt das untere Quartil die unteren 25% der Daten von den oberen 75% und das obere Quartil das obere Viertel von den unteren drei Vierteln. Der Bereich zwischen dem ersten und dritten Quartil beinhaltet die mittleren 50% der Daten und wird auch Interquartilabstand genannt. Im Boxplot ist dieser Abstand als blaues Rechteck dargestellt, während der Minimal- und Maximalwert als blaue horizontale Linie am oberen und unteren Ende gekennzeichnet sind. Die rote Linie innerhalb des Rechtecks zeigt den Median an. In der Abbildung werden die prozentualen Unterschiede gegenüber dem

Ausgangsfehler dargestellt. Dafür werden die absoluten Werte des Ausgangsfehlers in Tabelle 6.2 dargestellt. Die absoluten Werte, die nach den zehn Registrierungsdurchgängen entstanden sind, befinden sich außerdem in Abschnitt B.1.

	<b>ND</b>	<b>TFD</b>	<b>RFD</b>
Ausgangsfehler	0,32615	0,22998	0,31214

Tabelle 6.2: MCC Ausgangsfehler der drei Durchläufe bei einer Octreeauflösung von 0,5 m

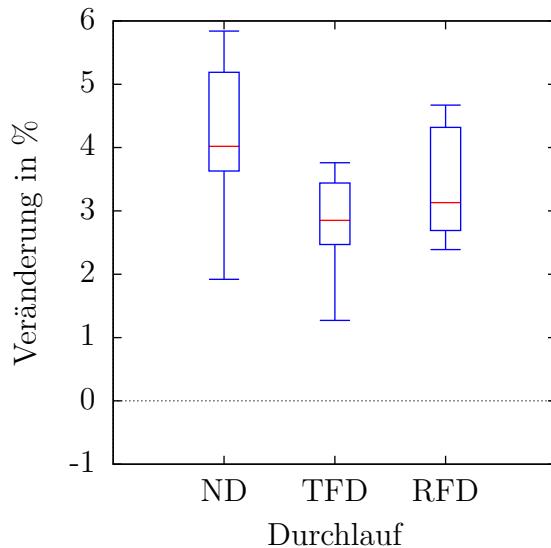


Abbildung 6.7: Prozentuale Veränderung des Matthews Korrelationskoeffizient bei einer Octreeauflösung von 0,5 m

Je ähnlicher der rekonstruierte Octree dem Referenzoctree ist, desto höher ist der Wert des MCC. Demnach ist zu erkennen, dass bei allen drei Durchläufen eine Reduktion des Ausgangsfehlers erzielt wird. Der ND, der den normalen Anwendungsfall widerspiegelt, weist mit einem Median von ca. 4% die größte Verbesserung auf. Außerdem ist die Varianz der Ergebnisse am größten. Die Resultate liegen zwischen 2% und 6%. Der TFD, welcher durch die translativen Ungenauigkeiten den größten Ausgangsfehler besitzt, hat den geringsten Interquartilabstand und liefert somit die zuverlässigsten Ergebnisse. Dabei erzielt er mit einem Median von 2,85% aber die geringste Zunahme des MCC. Der Durchlauf mit den rotatorischen Fehlern liegt mit einem Median von 3,13% zwischen den anderen beiden Durchläufen.

### Jensen-Shannon Divergenz

Die Belegtheitswahrscheinlichkeiten der Voxel sind ähnlicher umso niedriger der Wert der Jensen-Shannon Divergenz ist. Demnach repräsentiert eine negative prozentuale Veränderung in Abbildung 6.8 eine Verbesserung.

	<b>ND</b>	<b>TFD</b>	<b>RFD</b>
Ausgangsfehler	0,04463	0,04825	0,04517

Tabelle 6.3: JSD Ausgangsfehler der drei Durchläufe bei einer Octreeauflösung von 0,5 m

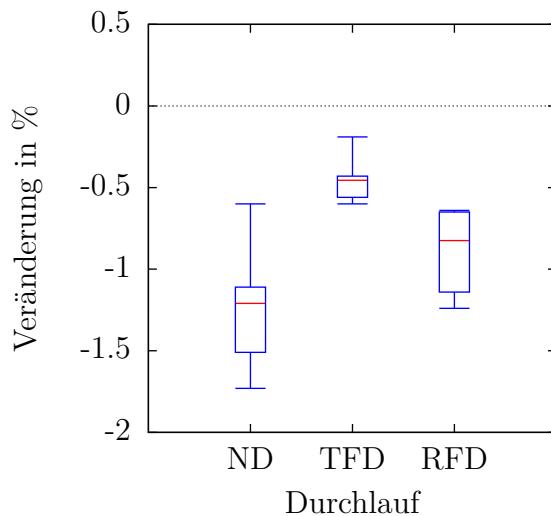


Abbildung 6.8: Prozentuale Veränderung der Jensen-Shannon Divergenz bei einer Octreeauflösung von 0,5 m

Es ergibt sich ein ähnliches Bild wie bei dem MCC. Der ND verzeichnet die besten Ergebnisse bei der größten Spanne verschiedener Ergebnisse. Der Durchlauf mit dem translativen Fehler erzielt die geringste Verbesserung, ist aber sehr zuverlässig bei den Ergebnissen. Der RFD liegt wiederum zwischen den beiden anderen Durchläufen.

### Durchschnittliche Hausdorff-Distanz

	<b>ND</b>	<b>TFD</b>	<b>RFD</b>
Ausgangsfehler	0,00527233	0,00634969	0,00508150

Tabelle 6.4: DHD Ausgangsfehler der drei Durchläufe bei einer Octreeauflösung von 0,5 m

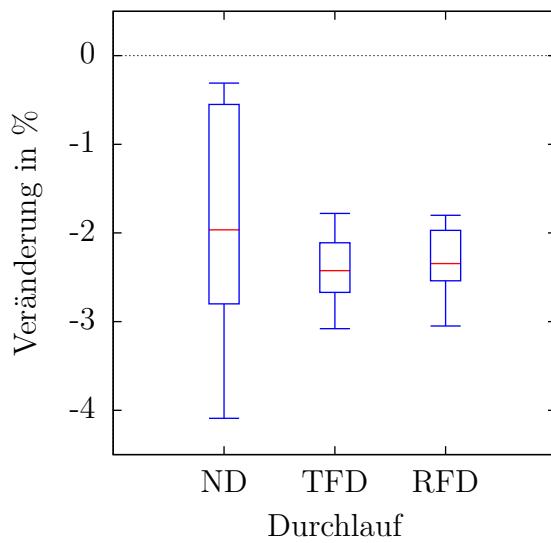


Abbildung 6.9: Prozentuale Veränderung der durchschnittlichen Hausdorff-Distanz bei einer Octreeauflösung von 0,5 m

Bei der durchschnittlichen Hausdorff-Distanz ist das Ziel ebenfalls eine negative Veränderung zum Ausgangsfehler, da somit eine kürzere Distanz zwischen den Voxelzentren vorliegt. Bei Betrachtung der Ausgangsfehler fällt auf, dass die eingeführten Rotationsfehler eine zufällige Verbesserung gegenüber dem normalen Anwendungsfall eingebracht haben.

Bei der **DHD** liegt der Median bei allen drei Durchläufen zwischen 2 - 2,5%. Die Varianz des normalen Anwendungsfalls ist, wie bei den beiden anderen Bewertungsmaßen, am größten. Die Fehler, die durch die **GPS** Störung und das Rauschen auf den Lagewinkel einbrachte wurden, werden beide konstant verbessert.

## Fazit

Das entwickelte Verfahren verbessert bei den drei Durchläufen die Verortung der Scandaten. Dies wird durch alle drei Bewertungsmaße bestätigt. Dabei wird der Fehler des normalen Anwendungsfalls prozentual am besten korrigiert, wobei der Ausgangsfehler bei diesem Durchlauf am geringsten ausfällt. Der **ND** und **TFD** besitzen jeweils translativ Fehler in verschiedenen Ausmaßen. Demnach war zu erwarten, dass der Durchlauf mit dem größeren Ausgangsfehler (**TFD**) die höchste Verbesserung erzielt. Dies ist jedoch weder bei den prozentualen noch bei den absoluten Werten der Fall. Ein Grund dafür könnte sein, dass der **ICP** bei einigen Registrierungen zu einem fehlerhaften lokalen Minimum konvergiert. Je größer der Ausgangsfehler ist, desto größer ist die Gefahr, dass ein falsches lokales Minimum erreicht wird. Es kann durch Ungenauigkeiten bei der Überführung der Scandaten

in den Octree dazu kommen, dass beispielsweise zusätzliche Voxel unterhalb der Oberfläche des Bodens belegt werden. Dies wird in Abbildung 6.10 im zweidimensionalen Raum abstrahiert. Grüne Zellen stehen für die korrekt belegten Voxel der Bodenfläche und die dunkelgrünen Punkte für die Voxelzentren. Die roten Zellen sind hingegen fehlerhaft besetzte Voxel unterhalb des Bodens. Die zu registrierenden Scandaten sind als schwarze Punkte dargestellt und der nächste Nachbar wird mit einem Pfeil markiert. Bei 6.10a ist zu erkennen, dass bei einem geringen Fehler eine korrekte Registrierung durchgeführt wird. Im Gegensatz dazu führt ein erhöhter Fehler zu einer fehlerhaften Registrierung (siehe 6.10b). Problematisch ist, dass solche fehlerhaft besetzten Voxel nicht mehr frei markiert werden können. Für das Freizeichnen muss ein Laserstrahl den Voxel durchqueren. Daher werden die Voxel nur freigezeichnet, wenn Laserstrahlen auf einen Voxel treffen, der noch weiter unterhalb der Oberfläche liegt, was einen noch größeren Fehler zur Folge hätte.

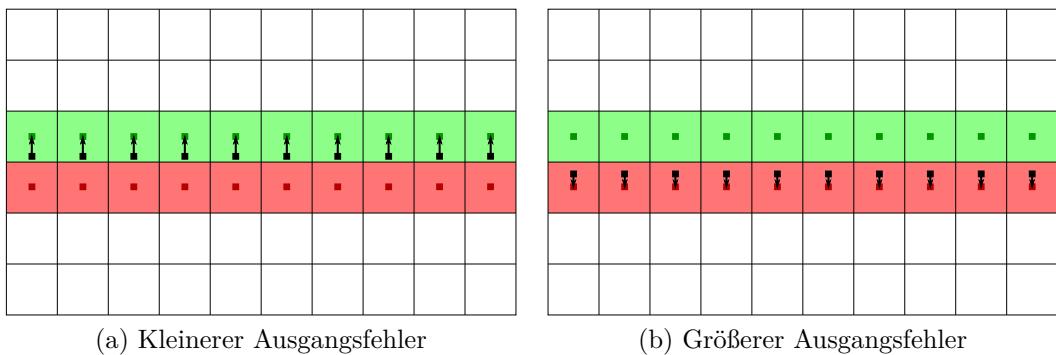


Abbildung 6.10: Unterschiedliche Registrierungsergebnisse bei verschiedenen Ausgangsfehlern

Weiterhin weist der ND die größte Varianz in den Ergebnissen auf. Dies liegt in der Parameterwahl des RANSAC Algorithmus begründet. Es wurde davon ausgegangen, dass der Schwellenwert für die Verwerfung von Ausreißern anhand der Auflösung des Octrees gewählt werden kann. In wie weit dieser Parameter zusätzlich von dem Ausgangsfehler und der Fehlerart abhängig ist, muss durch weitere Versuche untersucht werden. Da der Ausgangsfehler des RFD in den meisten Fällen dem des ND sehr ähnlich ist, bestätigt dass die Fehlerart zur Varianz beiträgt. Der Durchlauf mit den translativen Fehlern liefert jedoch die zuverlässigsten Resultate, obwohl er, ebenso wie der ND, translatable Fehler aufweist. Demnach müssen sowohl die Fehlerart als auch die Ausprägung des Fehlers in die Parameter beim RANSAC miteinfließen.

### 6.3.1 Weitere Octreeauflösungen

In diesem Abschnitt wird analysiert welche Genauigkeitsunterschiede beim Verwenden unterschiedlicher Auflösungen des Octrees entstehen. Da 0,5 m für die Kollisionsvermeidung bereits eine feine Auflösung bedeutet, werden in diesem Abschnitt gröbere Auflösungen betrachtet. Demnach werden folgend die Ergebnisse mit einer Octreeauflösung von einem und zwei Metern dargestellt und analysiert. Wie im vorhergehenden Abschnitt sind die absoluten Ergebnisse in [Abschnitt B.1](#).

#### Matthews Korrelationskoeffizient

	<b>ND</b>	<b>TFD</b>	<b>RFD</b>
Ausgangsfehler 1 m	0,49098	0,35222	0,48575
Ausgangsfehler 2 m	0,64924	0,48247	0,63142

Tabelle 6.5: [MCC](#) Ausgangsfehler der drei Durchläufe bei den Octreeauflösung von 1 und 2 m

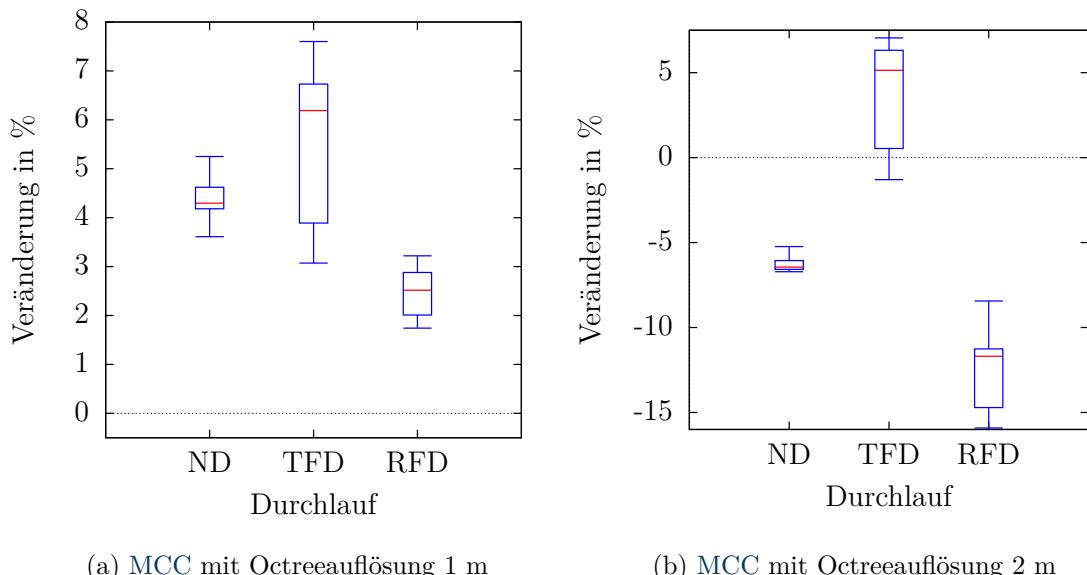


Abbildung 6.11: Prozentuale Veränderung des Matthews Korrelationskoeffizient bei weiteren Octreeauflösungen

Während die prozentuale Veränderung des [MCC](#) beim [TFD](#) noch am geringsten ausfiel, ist sie bei diesen beiden größeren Auflösungen jeweils am besten. Wie bei einer Auflösung von 0,5 m weist der Durchlauf mit der größten Verbesserung auch die größte Varianz auf. Die Resultate des [TFD](#) erstrecken sich bei einem Meter von 3% bis 8% und bei zwei Metern von -1,5% bis 7%. Bei einer Auflösung von einem

Meter verhält sich der **ND** ähnlich wie bei der 0,5 m Auflösung. Es liegt jedoch durch den leicht erhöhten Median und die Verringerung der Varianz eine geringe Verbesserung vor. Ein deutlicher Abfall des **MCC** ist jedoch bei der zwei Meter Auflösung zu erkennen, bei der der Median auf ca. -6,5% gefallen ist. Der Durchlauf mit dem rotatorischen Fehlern verschlechtert sich mit einem Median von -11,7% noch stärker bei der größten Auflösung. Auch bei der Auflösung von einem Meter nimmt der **MCC** gegenüber der 0,5 m Auflösung ab.

### Jensen-Shannon Divergenz

	<b>ND</b>	<b>TFD</b>	<b>RFD</b>
Ausgangsfehler 1 m	0,037355	0,043654	0,037956
Ausgangsfehler 2 m	0,028720	0,037687	0,030393

Tabelle 6.6: **JSD** Ausgangsfehler der drei Durchläufe bei den Octreeauflösung von 1 und 2 m

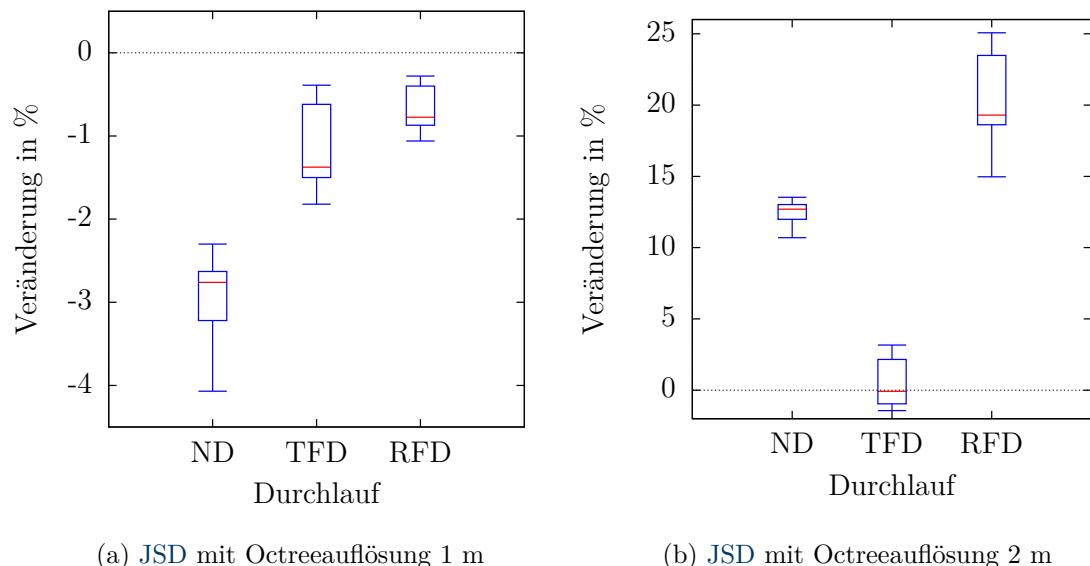


Abbildung 6.12: Prozentuale Veränderung der Jensen-Shannon Divergenz bei weiteren Octreeauflösungen

Bei der Belegtheitswahrscheinlichkeit und einer Auflösung von einem Meter verdoppelt der normale Durchlauf die Verbesserung gegenüber der 0,5 m Auflösung. Bei der zwei Meter Auflösung ist hingegen eine große Verschlechterung zu erkennen. Der Durchlauf mit dem zusätzlichen Rauschen auf den Lagewinkeln verhält sich wie bei der Bewertung durch den **MCC**. Bei der ein Meter Auflösung erzielt er noch positive Resultate, während bei der zwei Meter Auflösung eine starke Verschlechterung auftritt. Außerdem schneidet er von allen drei Durchläufen bei beiden Auflösungen am

schlechtesten ab. Auffällig ist, dass der **TFD** bei der zwei Meter Auflösung wieder am besten abschneidet, in dem der Median gegen 0% tendiert. Außerdem sind die Resultate bei der ein Meter Auflösung besser als bei der 0,5 m Auflösung.

### Durchschnittliche Hausdorff-Distanz

	<b>ND</b>	<b>TFD</b>	<b>RFD</b>
Ausgangsfehler 1 m	0,0117708	0,0144818	0,0113957
Ausgangsfehler 2 m	0,0283725	0,0364588	0,0295194

Tabelle 6.7: **DHD** Ausgangsfehler der drei Durchläufe bei den Octreeauflösung von 1 und 2 m

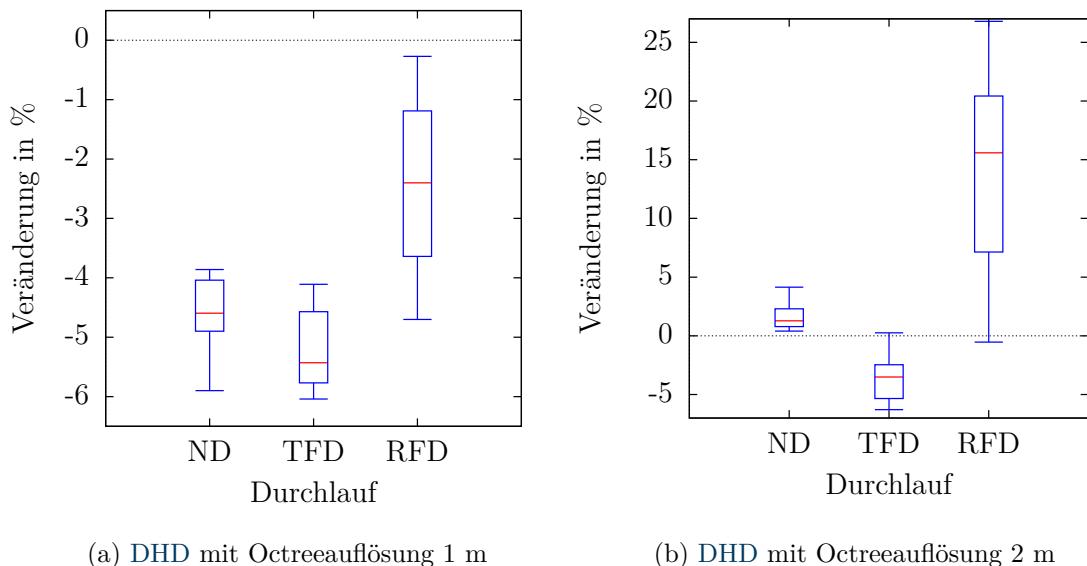


Abbildung 6.13: Prozentuale Veränderung der durchschnittlichen Hausdorff-Distanz bei weiteren Octreeauflösungen

Bei der durchschnittlichen Hausdorff-Distanz schneidet der Durchlauf mit den translativen Fehlern bei beiden Auflösungen am besten ab. Bei der 2 m Auflösung verschlechtern die anderen beiden Durchläufe den Ausgangsfehler. Besonders auffällig ist die große Varianz, die der **RFD** aufweist. Ähnlich der 0,5 m Auflösung, erzielen bei der 1 m Auflösung alle drei Durchläufe positive Resultate. Während der Durchlauf mit den rotatorischen Fehlern wiederum eine hohe Varianz verzeichnet, sind bei den anderen beiden Durchläufen konstant gute Ergebnisse, im Bereich zwischen 4% und 6%, zu erkennen.

## Fazit

Generell lässt sich feststellen, dass das entwickelte Verfahren bei der Octreeauflösung von 1 m, wie bei der 0,5 m Auflösung, bei allen drei Durchläufen durchweg positive Resultate liefert. Die prozentualen Verbesserungen liegen im einstelligen Bereich. Während bei der 0,5 m Auflösung der normale Anwendungsfall die aussichtsreichsten Ergebnisse erzielt, weist der **TFD** bei der 1 m Auflösung die beste Korrektur auf. Demnach werden größere translative Fehler bei sinkender Auflösung besser reduziert. Die Abbildung 6.14 illustriert noch einmal das Beispiel aus dem vorhergehenden Abschnitt, bei dem falsch belegte Voxel unterhalb der Oberfläche des Bodens vorhanden sind. Bei dieser Darstellung wird deutlich, dass die Wahrscheinlichkeit dieser fehlerhaften Voxel bei einer 1 m Auflösung geringer ist als bei der 0,5 m Auflösung. Aufgrund des größeren Diskretisierungsgrads führen erst größere Verortungsfehler zu diesem Umstand. Außerdem ist die Wahrscheinlichkeit höher, dass das lokale Minimum, welches der **ICP** erreicht, dem globalen Minimum entspricht. Während in 6.14a die nächsten Nachbarn negativ zur Registrierung beitragen, wird durch die nächsten Nachbarn in 6.14b eine Verbesserung erzielt.

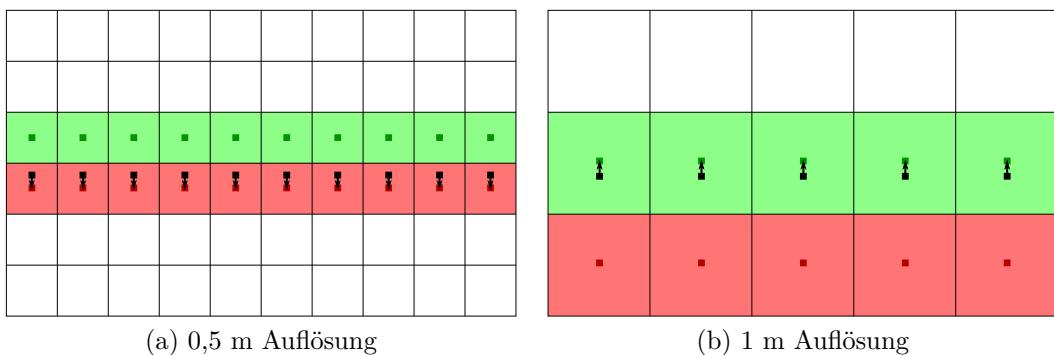


Abbildung 6.14: Unterschiedliche Registrierungsergebnisse bei verschiedenen Auflösungen

Bei der Auflösung von 2 m liefert nur der **TFD** Verbesserungen. Die anderen beiden Durchläufe zeigen zum Teil starke Verschlechterungen an. Je größer die Auflösung des Octrees ist, desto schlechter sind die Ergebnisse des Durchlaufs mit den rotatorischen Fehlern. Demnach besteht bei dem Registrierungsverfahren ein Unterschied bei der Korrektur von translativen und rotatorischen Ungenauigkeiten. Die Zielpunktwolke muss demnach eine ausreichende Dichte aufweisen um eine erfolgreiche Registrierung des **RFD** durchzuführen.

### 6.3.2 Visuelle Darstellung der Ergebnisse

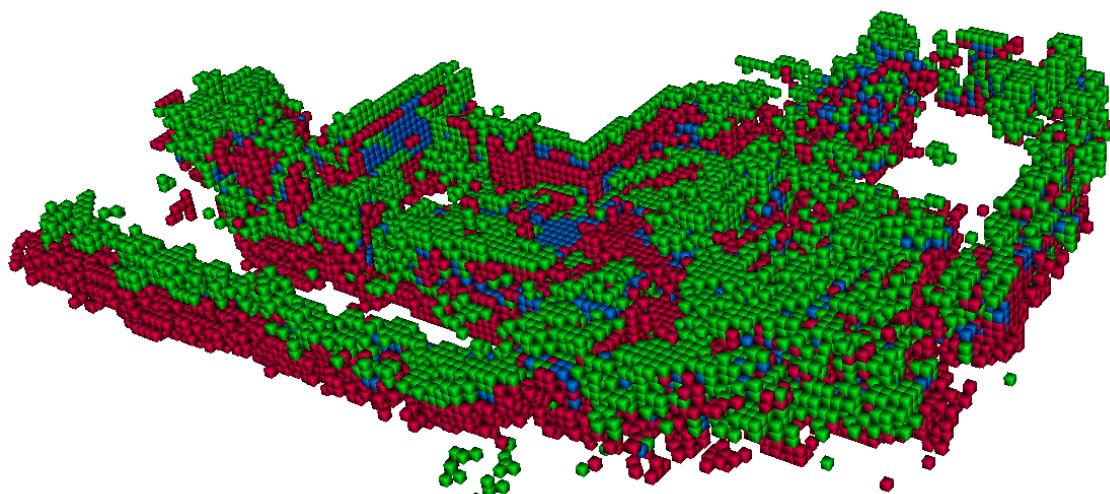
Da die Verbesserungen insgesamt im zumeist niedrigen, einstelligen Prozentbereich liegen, wird nachfolgend nur der Durchlauf mit den translativen Ungenauigkeiten dargestellt. Wie in den beiden vorhergehenden Abschnitten dargestellt, besitzt der TFD den größten Ausgangsfehler bei allen drei Bewertungsmaßen. Außerdem weist dieser Durchlauf bei der durchschnittlichen Hausdorff-Distanz, welche am besten die geometrische Lage widerspiegelt, sowohl prozentual als auch bei den absoluten Werten die größte Verbesserung auf. In Tabelle 6.8 ist die absolute Veränderung der DHD bei den drei Auflösungen für die Durchläufe aufgeführt. Hierbei wurde der Median der zehn Registrierungsdurchgänge verwendet. In den folgenden Darstellung wird der Octree mit der Auflösung von einem Meter dargestellt.

	<b>ND</b>	<b>TFD</b>	<b>RFD</b>
<b>Ausgangsfehler 0,5 m</b>	0,0052723	0,0063497	0,0050815
<b>Veränderung 0,5 m</b>	-0,0001226	-0,0001539	-0,0001222
<b>Ausgangsfehler 1 m</b>	0,0117708	0,0144818	0,0113957
<b>Veränderung 1 m</b>	-0,0005407	-0,0007864	-0,0002734
<b>Ausgangsfehler 2 m</b>	0,0283725	0,0364588	0,0295194
<b>Veränderung 2 m</b>	0,0003625	-0,0012768	0,0046014

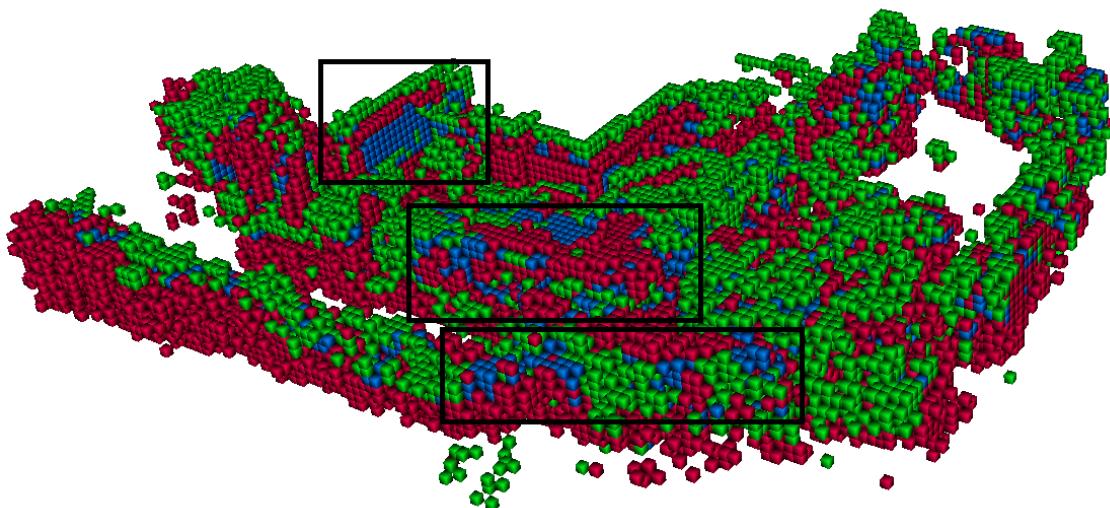
Tabelle 6.8: Absolute Veränderung der DHD in Metern für die drei Durchläufe

Die folgenden Abbildungen zeigen den Referenzoctree und den rekonstruierten Octree. Dabei werden Voxel, die von beiden Octrees belegt werden, in blau dargestellt. Grüne Voxel sind nur im Referenzoctree und rote Voxel nur im rekonstruierten Octree als belegt markiert. Da innere Voxel bei dieser Darstellung nicht betrachtet werden können, befinden sich dieselben Abbildungen mit halbtransparenten Voxeln in Abschnitt B.2.

Im oberen Teil der Abbildung 6.15 ist zunächst der rekonstruierte Octree zu sehen, der ohne Registrierung entstanden ist und somit den Ausgangsfehler darstellt. Anhand der vielen oberhalb liegenden, grünen Voxel ist zu erkennen, dass die Translationsfehler vor allem die  $z$ -Dimension betreffen. Durch die Voxelgröße von einem Meter lassen sich die Fehler in dieser Dimension auf ca. 2 - 3 m bestimmen. Zum Vergleich wird in 6.15b der Octree dargestellt, der mit dem entwickelten Verfahren entstanden ist. An ausgewählten Stellen ist eine Verbesserung durch die Zunahme an blauen Voxeln zu erkennen. Diese Stellen sind in der Abbildung mit einem schwarzen Rechteck markiert. Außerdem ist am unteren linken Rand zu erkennen, dass der Octree zwar an Höhe gewinnt aber auch fehlerhaft besetzte Voxel im unteren Teil enthält. Ursächlich dafür ist, dass bei einer fehlerhaften Registrierung diese Voxel nicht mehr freigezeichnet werden können.



(a) Ausgangsfehler



(b) Nach Anwendung der 3D Registrierung

Abbildung 6.15: Vergleich des rekonstruierten Octrees mit dem Referenzoctree

Abbildung 6.16 zeigt noch einmal den oberen linken Abschnitt der Abbildung 6.15 aus einer anderen Perspektive. Bei dem Gebäude im mittleren Rechteck sind nur an der rechten Seite mehr blaue Voxel hinzugekommen. Am oberen Rand ist jedoch eine weitere Reihe roter Pixel belegt worden, welche die Mauer der Höhe nach besser an den Referenzoctree angleicht. Dieser Umstand ist auch in den anderen beiden Abschnitten zu erkennen. Links unten sind die Voxel beispielsweise einen Meter höher belegt als im Referenzoctree. Für die Kollisionsvermeidung ist es vorteilhaft, wenn fehlerhafte Voxel höher liegen, als das abzubildene Objekt, damit dieses Objekt weiträumiger umflogen wird.

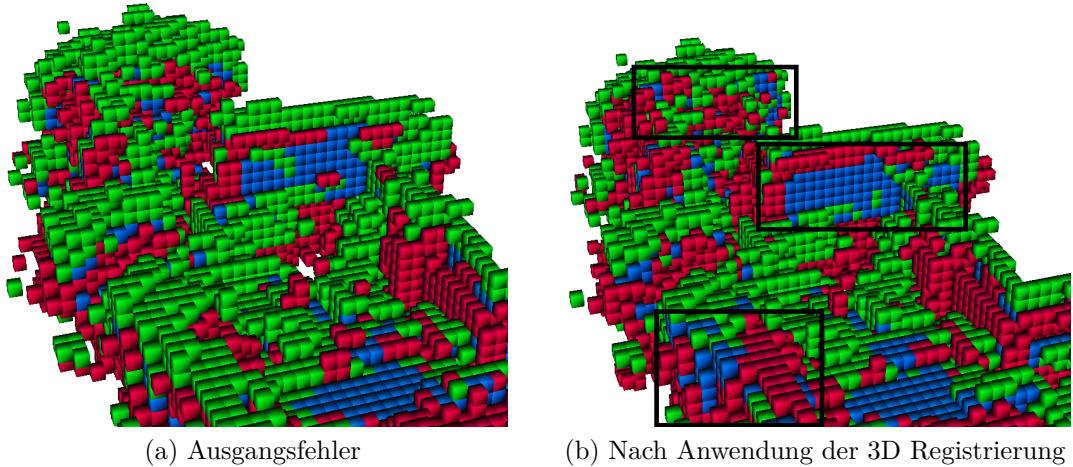


Abbildung 6.16: Vergleich des rekonstruierten Octrees mit dem Referenzoctree im Nahbereich

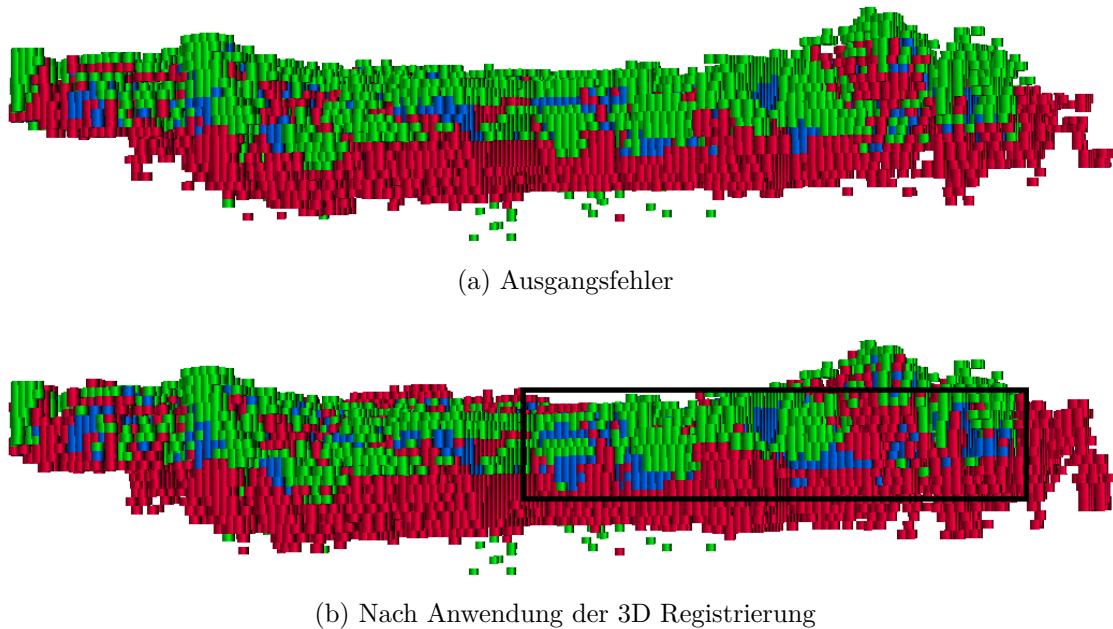


Abbildung 6.17: Vergleich des rekonstruierten Octrees mit dem Referenzoctree in der Seitenansicht

Die inkorrekt belegten Voxel treten vermehrt in der  $z$ -Dimension auf. Um dies zu verdeutlichen, werden die Octrees in Abbildung 6.17 aus der Seitenansicht betrachtet. Auch hier sind vermehrt blaue Voxel zu erkennen. Außerdem fällt auf, dass durch die Registrierung mehr Voxel belegt wurden.

### 6.3.3 Betrachtung der Höhenkorrektur

Wie in den Abbildungen im vorherigen Abschnitt zu erkennen ist, befindet sich der Fehler vor allem in der Höhe der Voxel. Demnach werden im Folgenden die

Höhenwerte der Navigationsdaten des **TFD** mit und ohne Registrierung verglichen. Dies wird in [Abbildung 6.18](#) dargestellt. Um einen Vergleich zu den tatsächlichen Höhenwerten zu ermöglichen, werden Daten aus dem Referenzdurchlauf verwendet. Dabei wurde manuell bestimmt an welcher Stelle der **TFD** die Positionen passiert, bei denen der Sensorträger beim Referenzdurchlauf pausiert hat.

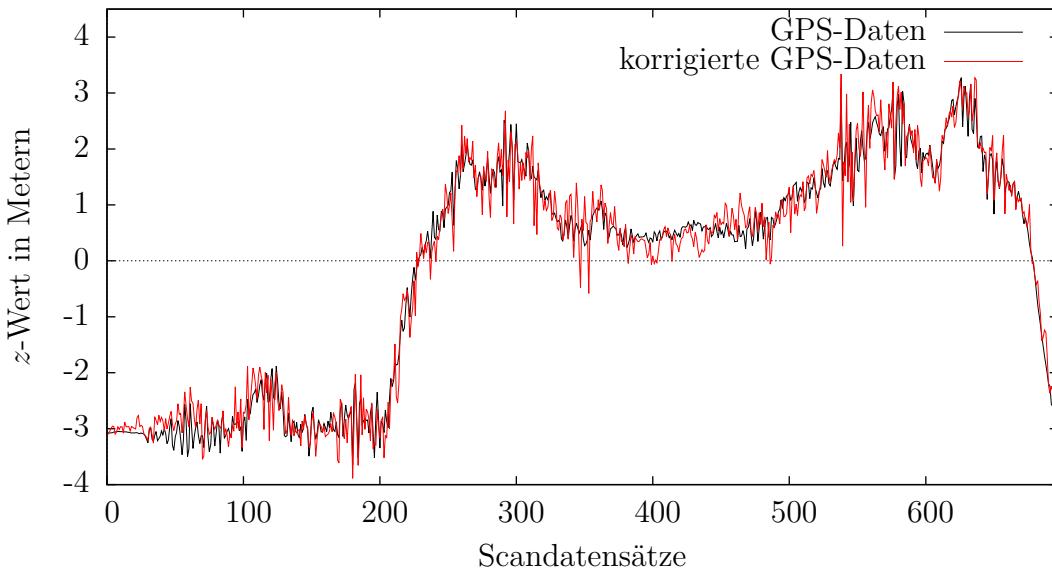


Abbildung 6.18: Test

Zunächst ist zu erkennen, dass der erste Scandatensatz des Durchlaufs mit einem Fehler von ca. zwei Metern in den Octree überführt wird. Bei dem ersten Datensatz kann jedoch noch keine Registrierung stattfinden, da der Octree zu diesem Zeitpunkt leer ist. Aus diesem Grund

### 6.3.4 Überprüfung der nahen Echtzeitanforderung

In diesem Abschnitt wird überprüft, ob der entwickelte Ansatz die nahe Echtzeitanforderung erfüllen kann, die in [Abschnitt 1.2](#) mit 1 - 2 Hz definiert wurde. Hierbei wurde gemessen, wie lange die Auswahl der Punkte und die Registrierung mit dem **ICP** für jeden neuen Scandatensatz dauert. Da der Durchlauf mit den translativen Ungenauigkeiten den größten Ausgangsfehler besitzt, wurde er für die Zeitmessungen ausgewählt. [Abbildung 6.19](#) veranschaulicht die Messungen für die drei verschiedenen Octreeauflösungen. Es wurde ein Rechner mit einem Intel 2,8 GHz Vierkernprozessor und 8 GB Arbeitsspeicher verwendet, wobei die entwickelte Anwendung keinen Gebrauch von Multithreading macht. Außerdem wurde die `QueryPerformanceCounter` Funktion für hochauflöste Zeitstempel von Windows verwendet. Zudem wurden in dieser Arbeit für jede Auflösung jeweils fünf Messungen durch-

geführt, wovon das arithmetische Mittel präsentiert wird. Um dabei die zufällige Komponente des **RANSAC** zu eliminieren, wurde für den Zufallsgenerator ein fester Startwert gesetzt.

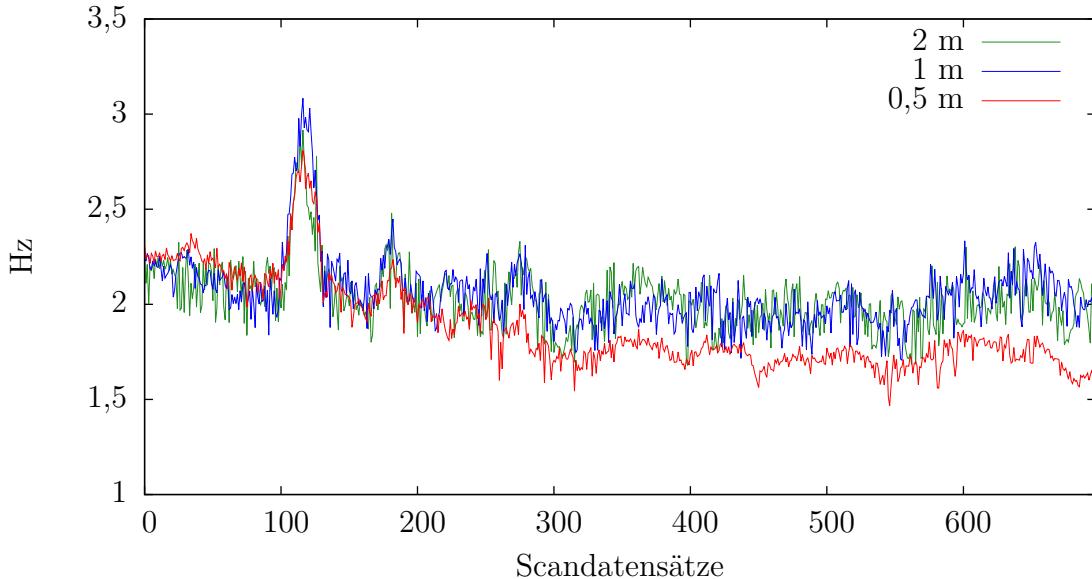


Abbildung 6.19: Zeitmessungen bei den Octreeauflösungen 0,5, 1 und 2 m

Die Zeit, die das Verfahren zur Verarbeitung der Daten braucht, liegt für alle drei getesteten Octreeauflösungen mindestens im geforderten Bereich von 1 - 2 Hz. Anzunehmen war eine Zunahme der Rechenzeit bei steigender Auflösung. Dies trifft bei der Auflösung von 0,5 m zu, da diese die meiste Zeit benötigt. Wider Erwarten sind die Ergebnisse der beiden weiteren Auflösungen sehr ähnlich. Hierzu werden im Folgenden die drei Faktoren betrachtet, die Einfluss auf die Rechenzeit haben. Diese Faktoren sind die Anzahl der gewählten Scanpunkte des aktuellen Datensatzes, die Anzahl der gefilterten Octree-Zentren und die Anzahl an Iterationen bei der Verwendung des **ICP**.

Bei allen drei Datensätzen befindet sich in etwa beim 100. Datensatz ein positiver Ausreißer. In Abbildung 6.20 ist zu erkennen, dass die Ursache in der Menge an gewählten Scanpunkten in der Eingangspunktwolke liegt. Diese Menge erhöht den Rechenaufwand innerhalb jeder Iteration des **ICP**, da für jeden Punkt die Suche des nächsten Nachbarn durchgeführt werden muss. Außerdem fließen diese Punktpaare in die Fehlermetrik mit ein. Wie in Abschnitt 4.1.4 beschrieben, ist diese Menge abhängig von der Anzahl an Punkten im Scandatensatz und basiert demnach nicht auf der Auflösung des Octrees. Eine Verringerung des Scandatensatzes entsteht, wenn mehrere Laserstrahlen auf kein Objekt innerhalb der Sensorreichweite treffen. Bei diesen Datensätzen dreht sich der Sensorträger, so dass für eine kurze Zeit die Laserstrahlen einige Bäume nicht mehr treffen, da der Hubschrauber die Strahlen

blockiert. Dieser Umstand zeigt, dass das entwickelte Verfahren Steigerungspotential besitzt wenn weniger Punkte verwendet werden. Der Nachteil liegt jedoch in der geringeren Genauigkeit bei einer Nutzung von weniger Scanpunkten.

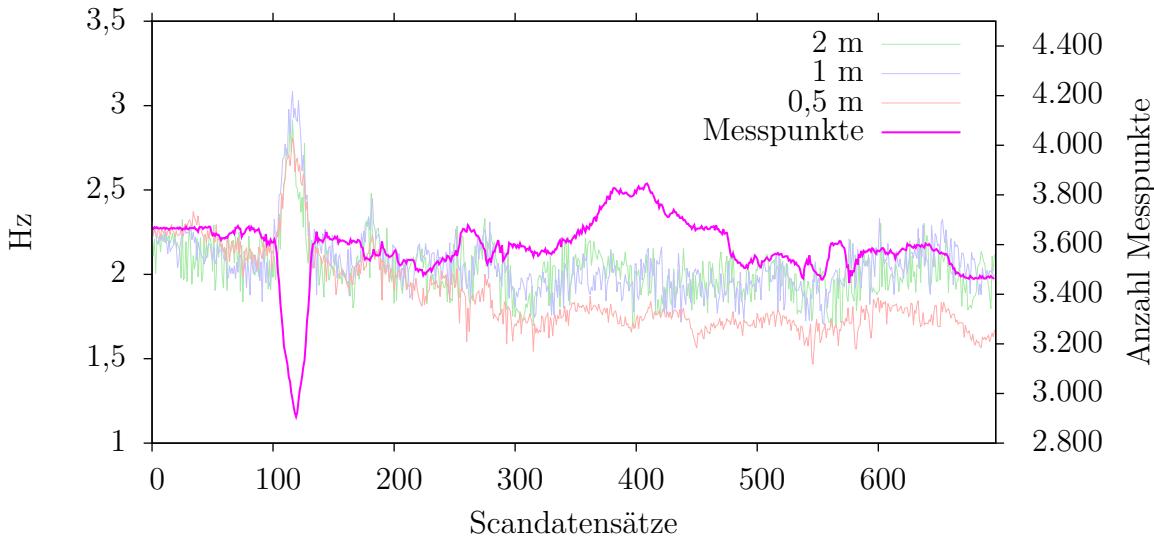


Abbildung 6.20: Einfluss der Anzahl der Scanpunkte auf die Zeitmessungen

Da die zeitlichen Unterschiede zwischen den drei Durchläufen nicht von der Menge der Punkte im Scandatensatz abhängig sind, muss die Auflösung des Octrees genauer analysiert werden. Bei Betrachtung des ersten und letzten Scandatensatzes mit der 0,5 m Auflösung, ist zu erkennen, dass die Berechnungszeit beim letzten Datensatz größer ist obwohl eine geringere Anzahl an Messpunkten vorliegt. Hierbei zeigt sich der Einfluss der Menge an gefilterten Octree-Zentren. Bei weniger Punkten in der Zielpunktewolke wird der Aufbau des *kD*-Baums beschleunigt. Dies hat zur Folge, dass bei der Suche des nächsten Nachbarn ein kleinerer Baum verwendet wird. Bei einer größeren Auflösung findet außerdem eine Beschleunigung bei der Reduzierung des Suchraumes statt, da weniger Voxel untersucht werden müssen.

In Abbildung 6.21 wird die Anzahl der gefilterten Octree-Zentren bei den verschiedenen Auflösungen illustriert. Zu Beginn eines Durchlaufs ist der Octree leer. Die Menge belegter Voxel steigt mit der Zeit an, da der auf den Sensorträger montierte Laserscanner stetig neue Areale erschließt und in den Octree überführt.

Die Abbildung zeigt, dass die Anzahl der gefilterten Voxel zwischen den einzelnen Auflösungen jeweils etwa um das Drei- bis Vierfache zunimmt. Durch die kubische Form der Octreevoxel, also die einheitliche Ausdehnung in alle drei Dimensionen, ist jedoch bei einer Verdopplung der Auflösung ein Anstieg um das Achtfache zu erwarten. Ein Voxel wird bei halber Kantenlänge in acht Voxel aufgeteilt. Das die Anzahl der zur Registrierung aus dem Octree gefilterten Voxel nicht um das Acht-

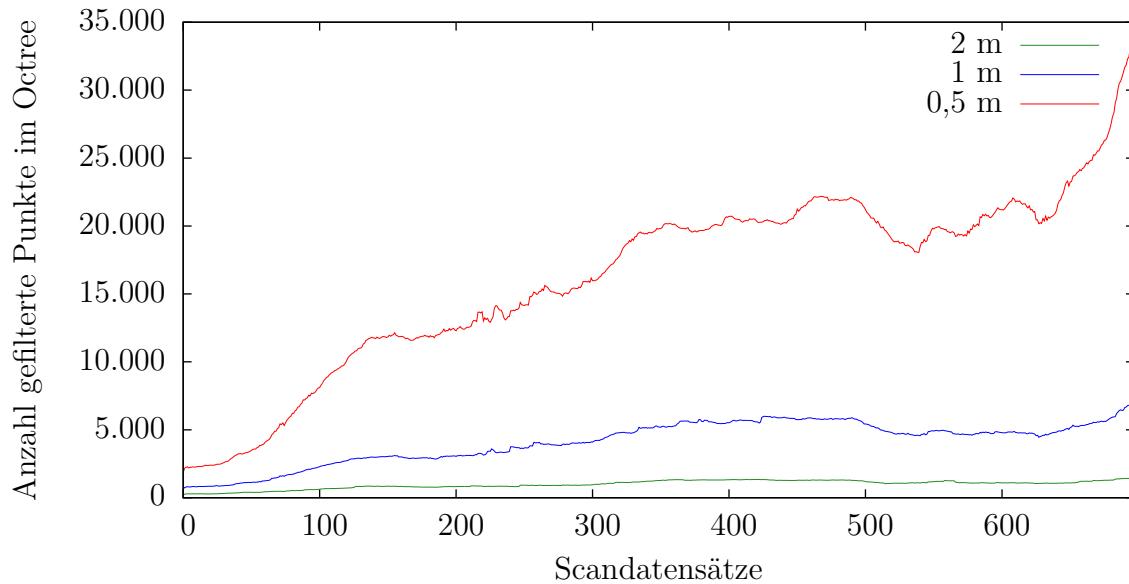


Abbildung 6.21: Anzahl der gefilterten Octree-Zentren bei den Octreeauflösungen 0,5, 1 und 2 m

fache zunimmt liegt vor allem an der Tatsache, dass bei planaren Flächen nur die vorderen vier Voxel belegt werden. Außerdem kann es vorkommen, dass nicht alle vier Voxel von einem Laserstrahl getroffen werden.

Die stark erhöhte Anzahl belegter Voxel bei der 0,5 m Auflösung beeinträchtigt die Rechenzeit deutlich. Die beiden anderen Auflösungen benötigen trotz unterschiedlich großer Mengen an Octree-Zentren ungefähr dieselbe Zeit. Dies lässt sich auf die verschiedene Iterationsanzahl beim Verwenden des ICP zurückführen. Während die 1 m Auflösung im Schnitt 8,24 Iterationen benötigt, liegt dieser Wert bei der 2 m Auflösung bei 12,13. Wie in Abschnitt 3.1 beschrieben, hängt die Anzahl an Iterationen von drei Terminierungskriterien ab. Die Auflösung beeinflusst dabei hauptsächlich die Fehlermetrik, da weniger Punkte in der Zielpunktwolke eine größere Distanz zwischen korrespondierenden Punkten bedeutet. Somit wird eine Fehlermetrik, die den Schwellenwert unterschreitet, der die Präzision des ICPs angibt, bei größeren Auflösungen langsamer erreicht. Bei den durchgeführten Registrierungen wurde für alle drei Auflösungen derselbe Schwellenwert verwendet. Sowohl eine Steigerung der Genauigkeit durch das Herabsetzen des Schwellenwertes als auch eine Beschleunigung des Verfahrens durch Erhöhen des Schwellenwertes kann erwartet werden. Dies wurde in dieser Arbeit nicht weiter betrachtet. Die Anzahl an Iterationen liegt bei der 0,5 m Auflösung bei 3,97. Bei diesem Durchlauf kann dadurch die erhöhte Menge an gefilterten Octree-Zentren jedoch nicht kompensiert werden.

## 7. Zusammenfassung und Ausblick

Diese Arbeit widmet sich der Korrektur fehlerhafter Navigationsdaten durch die 3D Registrierung von Laserscannerdaten. Ziel ist eine präzise Verortung von Scandaten in einer Hinderniskarte, welche unbemannte Luftfahrzeuge befähigen soll sich innerhalb unbekannter Umgebungen bewegen zu können. Die globale Hinderniskarte wird mittels eines Octrees repräsentiert. Diese Datenstruktur bietet eine speichereffiziente Datenhaltung und eine probabilistische Belegungsfunktion. Das entwickelte Registrierungsverfahren verwendet den Iterative Closest Point Algorithmus um neu akquirierte Scandaten dem Octree anzulegen. Dafür wurden die Zentren der belegten Voxel des Octrees als Zielpunktwolke definiert.

Um diesen Ansatz in naher Echtzeit verwenden zu können, wurden Beschleunigungsmethoden für die verschiedenen Phasen des ICPs betrachtet. Hierbei wurden Methoden aufgrund der Eignung für den vorliegenden Anwendungsfall ausgewählt. Die Selektion der Punkte, die im ICP verwendet werden sollen, hat einen großen Anteil an der Berechnungszeit. Demnach werden Punkte mit einem weiten Abstand zum Laserscanner entfernt, da die Messungenauigkeiten mit steigender Distanz zum Sensor zunehmen. Außerdem wird eine Untermenge der Scandatensätze durch das Covariance Sampling gefiltert. Dieses Verfahren verwendet ein Maß der geometrischen Stabilität für die Auswahl der Punkte. Anhand des Bereiches, indem die verbliebenen Scandaten liegen, werden Voxel im Octree bestimmt, die für die Registrierung relevant sind.

Der ICP führt in jeder Iteration eine Suche des nächsten Nachbarn für alle Punkte durch. Dieser Schritt ist sehr aufwändig und wurde daher durch die Verwendung eines kD-Baums beschleunigt. Weiterhin findet in jeder Iteration eine Verwerfung von Ausreißern durch einen RANSAC Algorithmus statt. Diese Ausreißer sind Punkte, die nicht positiv zur Registrierung beitragen, da sie keine korrekten Korresponden-

zen in der Zielpunktwolke aufweisen. Das Entfernen dieser Punkte führt zu einer schnelleren Berechnung der Fehlermetrik sowie einem besseren Registrierungsergebnis.

Zur Überprüfung der Funktionsfähigkeit des entwickelten Verfahrens wurde ein Versuch mit einem bodengebundenen Sensorträger durchgeführt. Bei diesem Versuch wurde ein Teil des DLR-Geländes in mehreren Durchläufen befahren. Somit konnte ein Referenzdurchlauf sowie mehrere Durchläufe mit fehlerhaften Navigationsdaten generiert werden. Hierbei wurde festgestellt, dass bei einer Octreeauflösung von 0,5 m und 1 m eine Verbesserung im einstelligen prozentualen Bereich erzielt wird. Dies bezieht sich sowohl auf rotatorische als auch auf translativen Fehler. Rotatorische Fehler werden bei sinkender Auflösung und somit bei sinkender Dichte der Zielpunktwolke in geringerem Maß korrigiert. Bei einer Auflösung von 2 m findet durch den starken Diskretisierungsgrad bereits eine Verschlechterung statt. Grobe translativen Fehler können bei dieser Auflösung jedoch korrigiert werden. Außerdem wird für die Auflösungen 0,5, 1 und 2 m die Anforderung an die nahe Echtzeit erfüllt und zum Teil übertroffen.

Da die Versuche mit einem bodengebundenen Sensorträger durchgeführt wurden, ist es notwendig die Ergebnisse mit einem Flugversuch zu verifizieren.

Eine Möglichkeit der Verbesserung des Verfahrens stellt eine Anpassung der Knoten des Octrees dar, damit die Registrierung nicht mehr anhand der Octree-Zentren stattfindet. Wie in der Arbeit von Dröschel et al. [DSB13] könnte eine bestimmte Anzahl an Punkten aus vorangegangenen Scandatensätzen in den Knoten gespeichert werden. Alternativ könnte anstatt des ICP die 3D-NDT Methode [MLD07] verwendet werden, um eine erfolgreiche Registrierung zu erhalten. Dafür müsste jeder Knoten Informationen über die vorliegende Normalverteilung der Punkte innerhalb des zugehörigen Voxels enthalten.

Zudem ist es möglich die korrigierten Navigationsdaten zur Schätzung nachfolgender Posen des UAs zu verwenden. Die Bestimmung der Pose erfolgt durch eine Fusion der Daten mehrerer Umgebungssensoren. Somit können die Korrekturdaten in die Sensorfusion integriert werden.

# A. Herleitungen

## A.1 Herleitung der Point-to-Plane Minimierung

Die folgende Herleitung basiert auf der Arbeit von Rusinkiewicz [Rus]. Die Point-to-Plane Metrik ist gegeben durch:

$$E = \sum_{i=1}^k ((\mathbf{R}p_i + \vec{t} - q_i) \cdot \vec{n}_i)^2 \quad (\text{A.1})$$

Nun wird aufgrund der Linearisierung

$$\mathbf{R} = \begin{pmatrix} c(\beta)c(\gamma) & s(\alpha)s(\beta)c(\gamma) - c(\alpha)s(\gamma) & c(\alpha)s(\beta)c(\gamma) + s(\alpha)s(\gamma) \\ c(\beta)s(\gamma) & s(\alpha)s(\beta)s(\gamma) + c(\alpha)c(\gamma) & c(\alpha)s(\beta)s(\gamma) - s(\alpha)c(\gamma) \\ -s(\beta) & s(\alpha)c(\beta) & c(\alpha)c(\beta) \end{pmatrix}^1$$

durch

$$\mathbf{R}_{app} \approx \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix} \quad (\text{A.2})$$

ersetzt. Dabei stehen  $\alpha$ ,  $\beta$  und  $\gamma$  jeweils für die Rotation um die  $x$ -,  $y$ - und  $z$ -Achse.  
Ersetzt man nun  $\mathbf{R}$  mit  $\mathbf{R}_{app}$  in der Formel A.1 ergibt dies

---

<sup>1</sup>s steht in dieser Matrix für den Sinus und c für den Cosinus

$$\begin{aligned}
E = & \sum_{i=1}^k ((p_{i,x} - \gamma p_{i,y} + \beta p_{i,z} + t_x - q_{i,x}) n_{i,x} + \\
& (\gamma p_{i,x} + p_{i,y} - \alpha p_{i,z} + t_y - q_{i,y}) n_{i,y} + \\
& (-\beta p_{i,x} + \alpha p_{i,y} - p_{i,z} + t_z - q_{i,z}) n_{i,z})^2
\end{aligned} \tag{A.3}$$

Wird diese Gleichung nun umgeformt in

$$\begin{aligned}
E = & \sum_{i=1}^k ((p_i - q_i) \cdot \vec{n}_i + \vec{t} \cdot \vec{n}_i + \\
& \alpha(p_{i,y} n_{i,z} - p_{i,z}) n_{i,y} + \\
& \beta(p_{i,z} n_{i,x} - p_{i,x}) n_{i,z} + \\
& \gamma(p_{i,x} n_{i,y} - p_{i,y}) n_{i,x})^2
\end{aligned} \tag{A.4}$$

ist zu erkennen, dass der hintere Teil durch das Kreuzprodukt  $c_i$  von  $p_i$  und  $\vec{n}_i$  und einen Vektor  $\vec{r} = (\alpha \ \beta \ \gamma)^T$  ersetzt werden kann. Daher erhält der Ausrichtungsfehler folgende Form:

$$E = \sum_{i=1}^k ((p_i - q_i) \cdot \vec{n}_i + \vec{t} \cdot \vec{n}_i + \vec{r} \cdot c_i)^2 \tag{A.5}$$

Dieser Fehler kann nun in Bezug auf  $\alpha, \beta, \gamma, t_x, t_y$  und  $t_z$  minimiert werden indem die partiellen Ableitungen null gesetzt werden.

$$\begin{aligned}
\frac{\delta E}{\delta \alpha} &= \sum_{i=1}^k 2c_{i,x}((p_i - q_i) \cdot \vec{n}_i + \vec{t} \cdot \vec{n}_i + \vec{r} \cdot c_i) = 0 \\
\frac{\delta E}{\delta \beta} &= \sum_{i=1}^k 2c_{i,y}((p_i - q_i) \cdot \vec{n}_i + \vec{t} \cdot \vec{n}_i + \vec{r} \cdot c_i) = 0 \\
\frac{\delta E}{\delta \gamma} &= \sum_{i=1}^k 2c_{i,z}((p_i - q_i) \cdot \vec{n}_i + \vec{t} \cdot \vec{n}_i + \vec{r} \cdot c_i) = 0 \\
\frac{\delta E}{\delta t_x} &= \sum_{i=1}^k 2n_{i,x}((p_i - q_i) \cdot \vec{n}_i + \vec{t} \cdot \vec{n}_i + \vec{r} \cdot c_i) = 0 \\
\frac{\delta E}{\delta t_y} &= \sum_{i=1}^k 2n_{i,y}((p_i - q_i) \cdot \vec{n}_i + \vec{t} \cdot \vec{n}_i + \vec{r} \cdot c_i) = 0 \\
\frac{\delta E}{\delta t_z} &= \sum_{i=1}^k 2n_{i,z}((p_i - q_i) \cdot \vec{n}_i + \vec{t} \cdot \vec{n}_i + \vec{r} \cdot c_i) = 0
\end{aligned} \tag{A.6}$$

Diese Gleichungen können auch kompakter in Matrixform dargestellt werden.

$$\sum_{i=1}^k \begin{pmatrix} c_{i,x}c_{i,x} & c_{i,x}c_{i,y} & c_{i,x}c_{i,z} & c_{i,x}n_{i,x} & c_{i,x}n_{i,y} & c_{i,x}n_{i,z} \\ c_{i,y}c_{i,x} & c_{i,y}c_{i,y} & c_{i,y}c_{i,z} & c_{i,y}n_{i,x} & c_{i,y}n_{i,y} & c_{i,y}n_{i,z} \\ c_{i,z}c_{i,x} & c_{i,z}c_{i,y} & c_{i,z}c_{i,z} & c_{i,z}n_{i,x} & c_{i,z}n_{i,y} & c_{i,z}n_{i,z} \\ n_{i,x}c_{i,x} & n_{i,x}c_{i,y} & n_{i,x}c_{i,z} & n_{i,x}n_{i,x} & n_{i,x}n_{i,y} & n_{i,x}n_{i,z} \\ n_{i,y}c_{i,x} & n_{i,y}c_{i,y} & n_{i,y}c_{i,z} & n_{i,y}n_{i,x} & n_{i,y}n_{i,y} & n_{i,y}n_{i,z} \\ n_{i,z}c_{i,x} & n_{i,z}c_{i,y} & n_{i,z}c_{i,z} & n_{i,z}n_{i,x} & n_{i,z}n_{i,y} & n_{i,z}n_{i,z} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ t_x \\ t_y \\ t_z \end{pmatrix} = - \sum_{i=1}^k \begin{pmatrix} c_{i,x}(p_i - q_i) \cdot \vec{n}_i \\ c_{i,y}(p_i - q_i) \cdot \vec{n}_i \\ c_{i,z}(p_i - q_i) \cdot \vec{n}_i \\ n_{i,x}(p_i - q_i) \cdot \vec{n}_i \\ n_{i,y}(p_i - q_i) \cdot \vec{n}_i \\ n_{i,z}(p_i - q_i) \cdot \vec{n}_i \end{pmatrix} \quad (\text{A.7})$$

Diese lineare Gleichung hat nun die Form  $\mathbf{C}\vec{x} = \vec{b}$ , wobei  $\mathbf{C}$  die  $6 \times 6$  Kovarianzmatrix darstellt, die durch  $c_i$  und  $\vec{n}_i$  geformt wurde.  $\vec{x}$  ist ein  $6 \times 1$  Vektor mit den Unbekannten und  $\vec{b}$  ein  $6 \times 1$  Vektor, der auf den Datenpunkten basiert. Diese Gleichung kann durch Standardmethoden wie der Cholesky Zerlegung gelöst und somit die optimale inkrementelle Transformation gefunden werden.

## A.2 Beweis zur Anwendbarkeit der SWZ zur Rotationsfindung

Arun et al. [AHB87] liefern den Beweis, dass die Rotation durch die Singulärwertzerlegung gefunden werden kann. Die Rotation verändert nicht die Länge eines Vektors, demnach kann die rechte Seite der Ausgangsformel

$$E(\mathbf{R}, \vec{t}) = \sum_{i=1}^N (q'_i - (\mathbf{R}p'_i))^2 \quad (\text{A.8})$$

erweitert werden

$$\begin{aligned} E(\mathbf{R}, \vec{t}) &= \sum_{i=1}^N (q'_i - (\mathbf{R}p'_i))^T (q'_i - (\mathbf{R}p'_i)) \\ &= \sum_{i=1}^N (q'^T q'_i + p'^T \mathbf{R}^T \mathbf{R} p'_i - q'^T \mathbf{R} p'_i - p'^T \mathbf{R}^T q'_i) \\ &= \sum_{i=1}^N (q'^T q'_i + p'^T p'_i - 2q'^T \mathbf{R} p'_i) \end{aligned} \quad (\text{A.9})$$

Da die Rotation nur den letzten Term beeinflusst ist das Minimieren von  $E(\mathbf{R}, \vec{t})$  äquivalent zum Maximieren von

$$F = \sum_{i=1}^N q'^T \mathbf{R} p'_i \quad (\text{A.10})$$

Mittels der Spur lässt sich [Formel A.10](#) umformen. Die Spur ist definiert als die Summe über die Einträge auf der Hauptdiagonalen einer Matrix.

$$\text{Spur}\left(\sum_{i=1}^N q_i'^T \mathbf{R} p_i'\right) = \text{Spur}(\mathbf{RH}) \quad (\text{A.11})$$

Wobei  $\mathbf{H}$  die in [Abschnitt 5.4.1](#) beschriebenen Matrix darstellt.

**Lemma:** Für jede positiv-definite Matrix  $\mathbf{AA}^T$  und jede orthonormale Matrix  $\mathbf{B}$  gilt

$$\text{Spur}(\mathbf{AA}^T) \geq \text{Spur}(\mathbf{BA} \mathbf{A}^T) \quad (\text{A.12})$$

**Beweis des Lemma:** Wenn  $a_i$  die  $i$ te Spalte von  $\mathbf{A}$  ist, dann

$$\text{Spur}(\mathbf{BA} \mathbf{A}^T) = \text{Spur}(\mathbf{A}^T \mathbf{B} \mathbf{A}) = \sum_{i=1}^N a_i^T (\mathbf{B} a_i) \quad (\text{A.13})$$

Durch die Cauchy-Schwarzsche Ungleichung ergibt sich

$$a_i^T (\mathbf{B} a_i) \leq \sqrt{(a_i^T a_i)(a_i^T \mathbf{B}^T \mathbf{B} a_i)} = a_i^T a_i \quad (\text{A.14})$$

Dies führt zum Beweis des Lemma:

$$\text{Spur}(\mathbf{BA} \mathbf{A}^T) \leq \sum_{i=1}^N a_i^T a_i = \text{Spur}(\mathbf{AA}^T) \quad (\text{A.15})$$

Anschließend ist die Singulärwertzerlegung durch  $\mathbf{H} = \mathbf{U} \Lambda \mathbf{V}^T$  gegeben, wobei  $\mathbf{U}$  und  $\mathbf{V}$  orthonormale  $3 \times 3$  Matrizen und  $\Lambda$  eine  $3 \times 3$  Diagonalmatrix mit nichtnegativen Einträgen ist. Sei nun

$$\mathbf{X} = \mathbf{V} \mathbf{U}^T \quad (\text{A.16})$$

ergibt sich mit

$$\mathbf{XH} = \mathbf{V} \mathbf{U}^T \mathbf{U} \Lambda \mathbf{V}^T = \mathbf{V} \Lambda \mathbf{V}^T \quad (\text{A.17})$$

eine symmetrische und positiv-definite Matrix. Durch das Lemma gilt für jede orthonormale  $3 \times 3$  Matrix  $\mathbf{B}$ ,

$$\text{Spur}(\mathbf{XH}) \geq \text{Spur}(\mathbf{BXH}) \quad (\text{A.18})$$

Dementsprechend maximiert die orthonormale Matrix  $\mathbf{X}$  die [Formel A.10](#) und ergibt damit die Rotationsmatrix.

## B. Weitere Ergebnisse

### B.1 Absolute Werte

Im Folgenden werden die Ergebnisse der Versuche in tabellarischer Form dargestellt. Dabei enthalten die Tabellen die Werte der drei Maße in allen zehn Durchgängen.

MCC		ND	TFD	RFD
	Ausgangsfehler	0,32615	0,22998	0,31214
	Durchgang 1	0,34486	0,23633	0,32156
	Durchgang 2	0,33798	0,23289	0,32670
	Durchgang 3	0,33240	0,23442	0,32562
	Durchgang 4	0,33840	0,23668	0,32055
	Durchgang 5	0,34519	0,23789	0,31961
	Durchgang 6	0,34013	0,23638	0,32608
	Durchgang 7	0,33756	0,23565	0,31993
	Durchgang 8	0,33798	0,23788	0,32245
	Durchgang 9	0,34308	0,23839	0,32128
	Durchgang 10	0,34076	0,23862	0,32225

Tabelle B.1: Absolute Werte des Matthews Korrelationskoeffizient bei einer Octreeauflösung von 0,5 m

		<b>ND</b>	<b>TFD</b>	<b>RFD</b>
<b>JSD</b>	Ausgangsfehler	0,04463	0,04825	0,04517
	Durchgang 1	0,04388	0,04803	0,04480
	Durchgang 2	0,04414	0,04816	0,04461
	Durchgang 3	0,04437	0,04810	0,04466
	Durchgang 4	0,04412	0,04803	0,04485
	Durchgang 5	0,04386	0,04798	0,04488
	Durchgang 6	0,04407	0,04804	0,04464
	Durchgang 7	0,04418	0,04806	0,04488
	Durchgang 8	0,04414	0,04798	0,04488
	Durchgang 9	0,04396	0,04797	0,04473
	Durchgang 10	0,04403	0,04796	0,04480

Tabelle B.2: Absolute Werte der Jensen-Shannon Divergenz bei einer Octreeauflösung von 0,5 m

		<b>ND</b>	<b>TFD</b>	<b>RFD</b>
<b>DHD</b>	Ausgangsfehler	0,00527233	0,00634969	0,00508150
	Durchgang 1	0,00505680	0,00618433	0,00496178
	Durchgang 2	0,00525378	0,00623663	0,00496328
	Durchgang 3	0,00523036	0,00622708	0,00492632
	Durchgang 4	0,00524354	0,00618013	0,00499026
	Durchgang 5	0,00512463	0,00619691	0,00498128
	Durchgang 6	0,00514136	0,00621592	0,00494527
	Durchgang 7	0,00516266	0,00619948	0,00497689
	Durchgang 8	0,00525614	0,00616085	0,00495235
	Durchgang 9	0,00507974	0,00615392	0,00498872
	Durchgang 10	0,00517476	0,00619476	0,00495680

Tabelle B.3: Absolute Werte der durchschnittlichen Hausdorff-Distanz bei einer Octreeauflösung von 0,5 m

		<b>ND</b>	<b>TFD</b>	<b>RFD</b>
<b>MCC</b>	Ausgangsfehler	0,49098	0,35222	0,48575
	Durchgang 1	0,51369	0,37613	0,49830
	Durchgang 2	0,51155	0,37568	0,49539
	Durchgang 3	0,50871	0,37899	0,49420
	Durchgang 4	0,51149	0,36592	0,49974
	Durchgang 5	0,51674	0,36639	0,50139
	Durchgang 6	0,51156	0,36302	0,49982
	Durchgang 7	0,51573	0,37541	0,49765
	Durchgang 8	0,50904	0,37594	0,49710
	Durchgang 9	0,51308	0,36580	0,49914
	Durchgang 10	0,51260	0,37261	0,49552

Tabelle B.4: Absolute Werte des Matthews Korrelationskoeffizient bei einer Octreeauflösung von 1 m

		<b>ND</b>	<b>TFD</b>	<b>RFD</b>
<b>JSD</b>	Ausgangsfehler	0,037355	0,043654	0,037956
	Durchgang 1	0,036244	0,043002	0,037635
	Durchgang 2	0,036371	0,042983	0,037804
	Durchgang 3	0,036497	0,042860	0,037850
	Durchgang 4	0,036368	0,043372	0,037664
	Durchgang 5	0,036103	0,043332	0,037555
	Durchgang 6	0,036372	0,043482	0,037609
	Durchgang 7	0,036153	0,042998	0,037661
	Durchgang 8	0,036480	0,042978	0,037782
	Durchgang 9	0,036279	0,043385	0,037625
	Durchgang 10	0,035836	0,043106	0,037821

Tabelle B.5: Absolute Werte der Jensen-Shannon Divergenz bei einer Octreeauflösung von 1 m

		<b>ND</b>	<b>TFD</b>	<b>RFD</b>
<b>DHD</b>	Ausgangsfehler	0,0117708	0,0144818	0,0113957
	Durchgang 1	0,0112497	0,0136076	0,0109154
	Durchgang 2	0,0113162	0,0136640	0,0113652
	Durchgang 3	0,0111937	0,0136503	0,0112913
	Durchgang 4	0,0112213	0,0138249	0,0111134
	Durchgang 5	0,0110767	0,0137964	0,0108602
	Durchgang 6	0,0112956	0,0138859	0,0111140
	Durchgang 7	0,0111111	0,0137268	0,0111742
	Durchgang 8	0,0112959	0,0136075	0,0109805
	Durchgang 9	0,0112260	0,0138199	0,0111306
	Durchgang 10	0,0112341	0,0136455	0,0112603

Tabelle B.6: Absolute Werte der durchschnittlichen Hausdorff-Distanz bei einer Octreeauflösung von 1 m

		<b>ND</b>	<b>TFD</b>	<b>RFD</b>
<b>MCC</b>	Ausgangsfehler	0,64924	0,48247	0,63142
	Durchgang 1	0,61529	0,51647	0,55750
	Durchgang 2	0,60989	0,50574	0,57815
	Durchgang 3	0,60656	0,50264	0,53816
	Durchgang 4	0,60662	0,51297	0,56035
	Durchgang 5	0,61173	0,51269	0,55776
	Durchgang 6	0,60723	0,50887	0,55780
	Durchgang 7	0,60976	0,48507	0,53093
	Durchgang 8	0,60565	0,47831	0,53853
	Durchgang 9	0,60626	0,47626	0,55212
	Durchgang 10	0,60761	0,51499	0,56202

Tabelle B.7: Absolute Werte des Matthews Korrelationskoeffizient bei einer Octreeauflösung von 2 m

		<b>ND</b>	<b>TFD</b>	<b>RFD</b>
<b>JSD</b>	Ausgangsfehler	0,028720	0,037687	0,030393
	Durchgang 1	0,031794	0,037147	0,036232
	Durchgang 2	0,032247	0,037707	0,034942
	Durchgang 3	0,032425	0,037867	0,037553
	Durchgang 4	0,032605	0,037326	0,036053
	Durchgang 5	0,032111	0,037361	0,036283
	Durchgang 6	0,032371	0,037612	0,036158
	Durchgang 7	0,032163	0,038500	0,038014
	Durchgang 8	0,032460	0,038859	0,037530
	Durchgang 9	0,032494	0,038881	0,036542
	Durchgang 10	0,032366	0,037267	0,036002

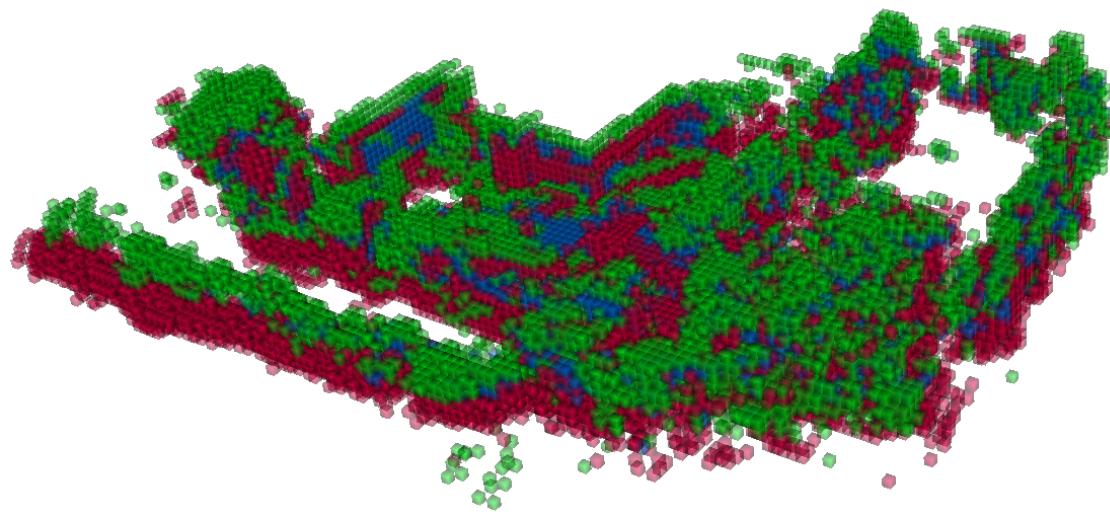
Tabelle B.8: Absolute Werte der Jensen-Shannon Divergenz bei einer Octreeauflösung von 2 m

		<b>ND</b>	<b>TFD</b>	<b>RFD</b>
<b>DHD</b>	Ausgangsfehler	0,0283725	0,0364588	0,0295194
	Durchgang 1	0,0290252	0,0342356	0,0339035
	Durchgang 2	0,0285204	0,0352074	0,0293627
	Durchgang 3	0,0293513	0,0359241	0,0374276
	Durchgang 4	0,0287287	0,0345105	0,0305846
	Durchgang 5	0,0284899	0,0345693	0,0350811
	Durchgang 6	0,0285968	0,0351566	0,0316270
	Durchgang 7	0,0287607	0,0355623	0,0370848
	Durchgang 8	0,0295501	0,0365532	0,0355506
	Durchgang 9	0,0287413	0,0353742	0,0343381
	Durchgang 10	0,0286804	0,0341667	0,0316972

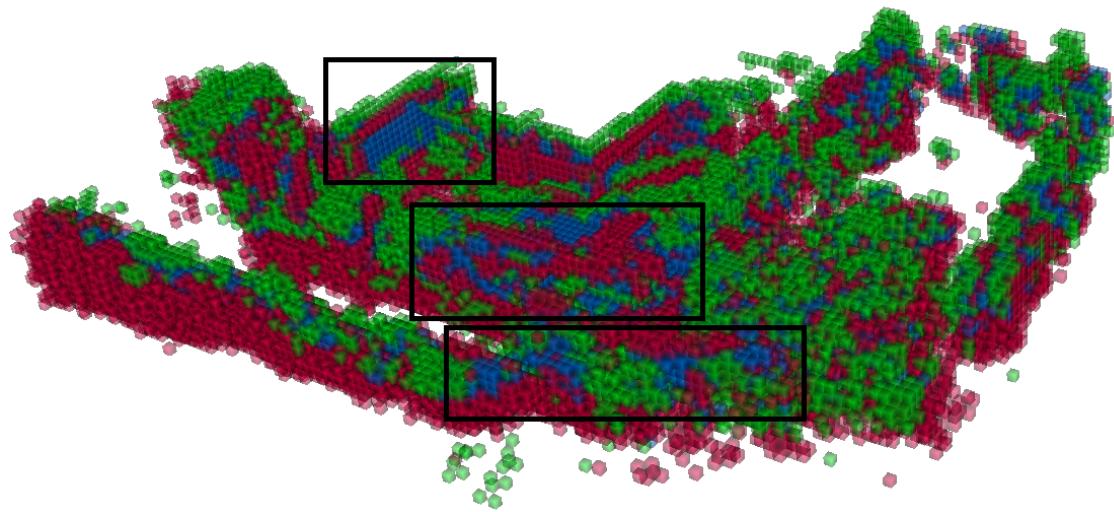
Tabelle B.9: Absolute Werte der durchschnittlichen Hausdorff-Distanz bei einer Octreeauflösung von 2 m

## B.2 Visuelle Ergebnisse

Nachfolgend werden die Abbildungen aus Abschnitt 6.3.2 mit halbtransparenten Voxel dargestellt.



(a) Ausgangsfehler



(b) Nach Anwendung der 3D Registrierung

Abbildung B.1: Vergleich von rekonstruierten Octrees mit dem Referenzoctree

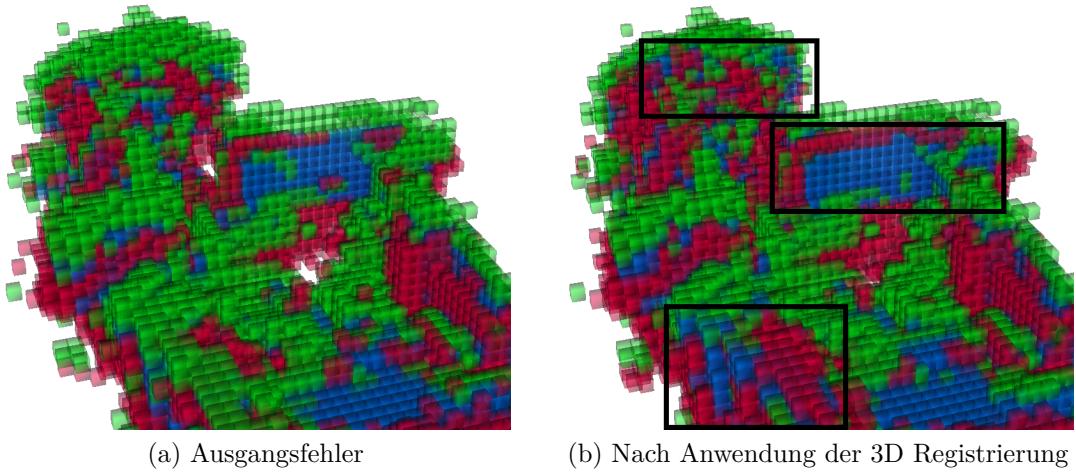


Abbildung B.2: Vergleich von rekonstruierten Octrees mit dem Referenzoctree im Nahbereich

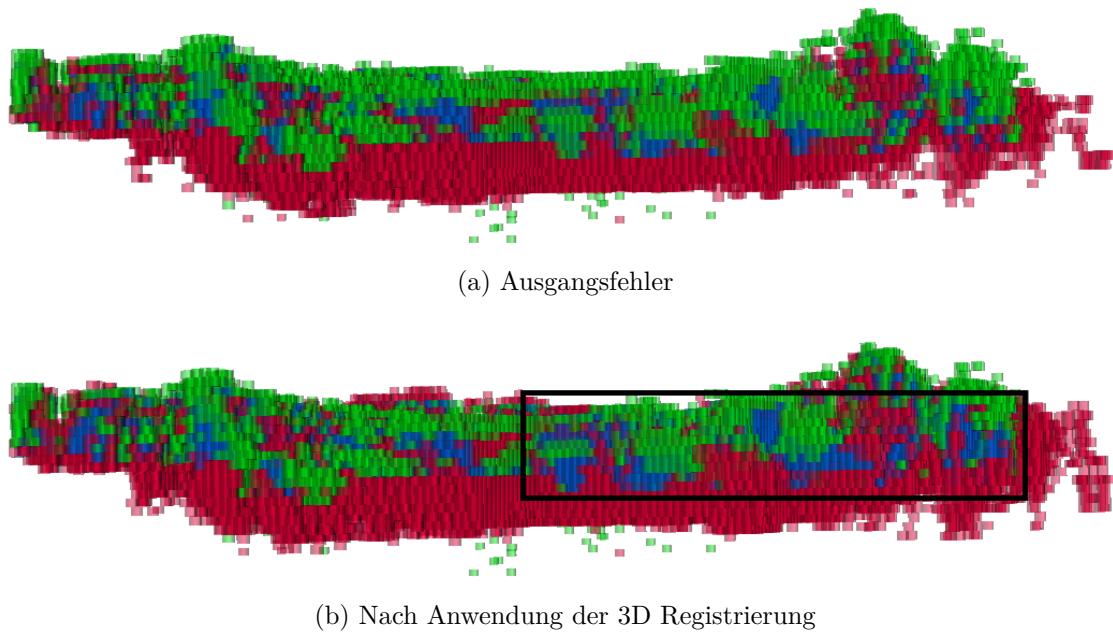


Abbildung B.3: Vergleich von rekonstruierten Octrees mit dem Referenzoctree in der Seitenansicht

### B.3 Vergleich der Fehlermetriken

In diesem Abschnitt werden die Point-to-Point und Point-to-Plane Fehlermetrik in tabellarischer Form verglichen. Hierfür wurden die drei Durchläufe aus [Abschnitt 6.1](#) und die Bewertungsmaße aus [Abschnitt 6.2](#) verwendet. Dabei entspricht ein höherer Wert bei dem [MCC](#) und ein niedriger Wert bei der [JSD](#) und [DHD](#) einer Verbesserung. Es wurde für alle drei Durchläufe ein fester Startwert für den Zufallsgenerator des [RANSAC](#) Algorithmus [Abschnitt 5.3.2](#) festgelegt. Außerdem wurde eine Octree-

auflösung von einem Meter verwendet. In den nachfolgenden Tabellen ist zu erkennen, dass die Point-to-Plane Fehlermetrik in fast allen Fällen eine Verbesserung gegenüber dem Ausgangsfehler erzielt. Die Ergebnisse der Point-to-Point Fehlermetrik übertreffen diese Verbesserungen jedoch bei allen Durchläufen und Bewertungsmaßen.

<b>ND</b>	<b>MCC</b>	<b>JSD</b>	<b>DHD</b>
<b>Ausgangsfehler</b>	0,490982839	0,0117708	0,037354913
<b>Point-to-Point</b>	0,513688055	0,0112497	0,036243721
<b>Point-to-Plane</b>	0,518924967	0,0112893	0,036272683

Tabelle B.10: Vergleich der Point-to-Point und Point-to-Plane Fehlermetrik beim normalen Anwendungsfall

<b>TFD</b>	<b>MCC</b>	<b>JSD</b>	<b>DHD</b>
<b>Ausgangsfehler</b>	0,352218872	0,0144818	0,04365394
<b>Point-to-Point</b>	0,376133262	0,0136076	0,04300160
<b>Point-to-Plane</b>	0,353103998	0,0141225	0,04405412

Tabelle B.11: Vergleich der Point-to-Point und Point-to-Plane Fehlermetrik beim Durchlauf mit translativen Fehlern

<b>RFD</b>	<b>MCC</b>	<b>JSD</b>	<b>DHD</b>
<b>Ausgangsfehler</b>	0,485753677	0,0113957	0,03795637
<b>Point-to-Point</b>	0,498297962	0,0109154	0,03763457
<b>Point-to-Plane</b>	0,491985207	0,0114726	0,03903333

Tabelle B.12: Vergleich der Point-to-Point und Point-to-Plane Fehlermetrik beim Durchlauf mit rotatorischen Fehlern

## C. Beispiel zur Suche des nächsten Nachbarn in einem $k$ D-Baum

Die Suche des nächsten Nachbarn in einem  $k$ D-Baum wird folgend an einem Beispiel im zweidimensionalen Raum erläutert. Zunächst wird der Baum durchlaufen als sollte der Anfragepunkt  $q$  eingefügt werden. Der Punkt in dem Blatt, das auf diese Weise erreicht wird, ist der aktuelle Kandidat. In Abbildung C.1 ist dies der Punkt  $p_{10}$ . Anschließend wird ein Kreis um den Anfragepunkt gelegt, dessen Radius der Distanz zum Kandidaten entspricht.

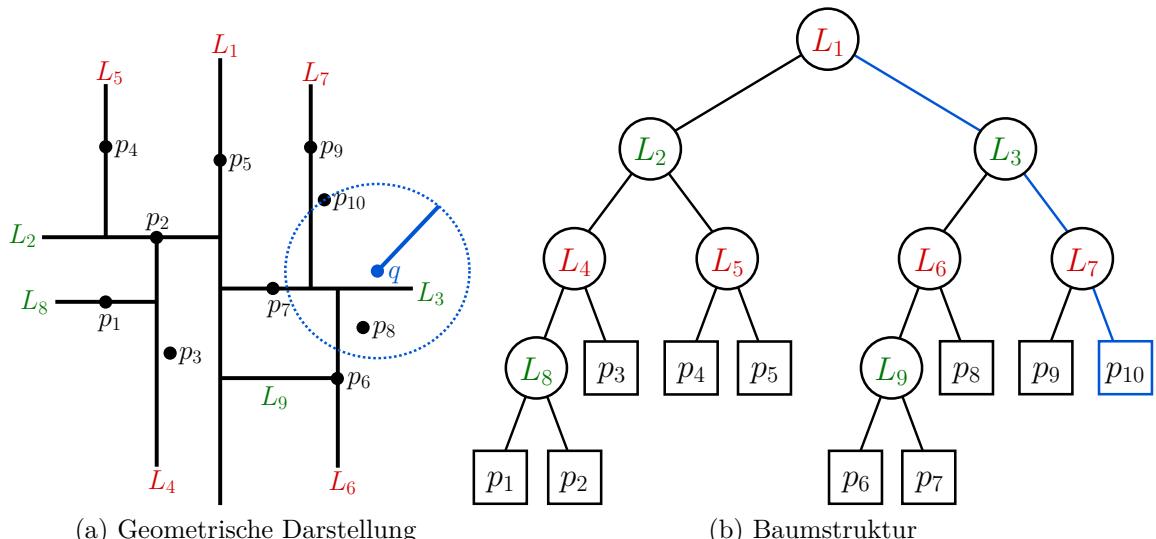


Abbildung C.1: Suche des nächsten Nachbarn im  $k$ D-Baum (1. Schritt)

Nun wird der Baum rückwärts durchlaufen. Dabei werden nur noch Knoten besucht, bei denen sich das zugehörige Rechteck mit dem Kreis überschneidet. In Abbildung C.2 ist zu sehen, dass das Rechteck links von  $L_7$  eine Überlappung mit dem

Kreis aufweist. Daher wird geprüft, ob der Punkt  $p_9$  näher an dem Anfragepunkt liegt als der Kandidat. Da dies nicht der Fall ist wird der Punkt verworfen.

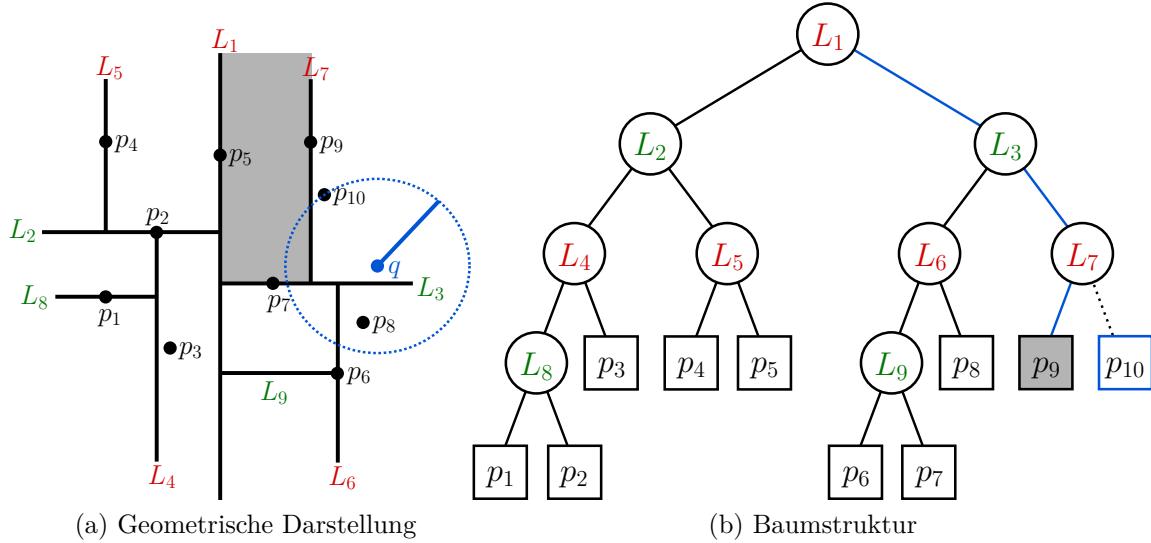


Abbildung C.2: Suche des nächsten Nachbarn im kD-Baum (2. Schritt)

Im dritten Schritt wird der Teilbereich unterhalb der Teilungslinie  $L_3$  betrachtet. Hierbei wird festgestellt, dass der Punkt  $p_8$  näher an  $q$  liegt und somit als neuer Kandidat bestimmt wird.

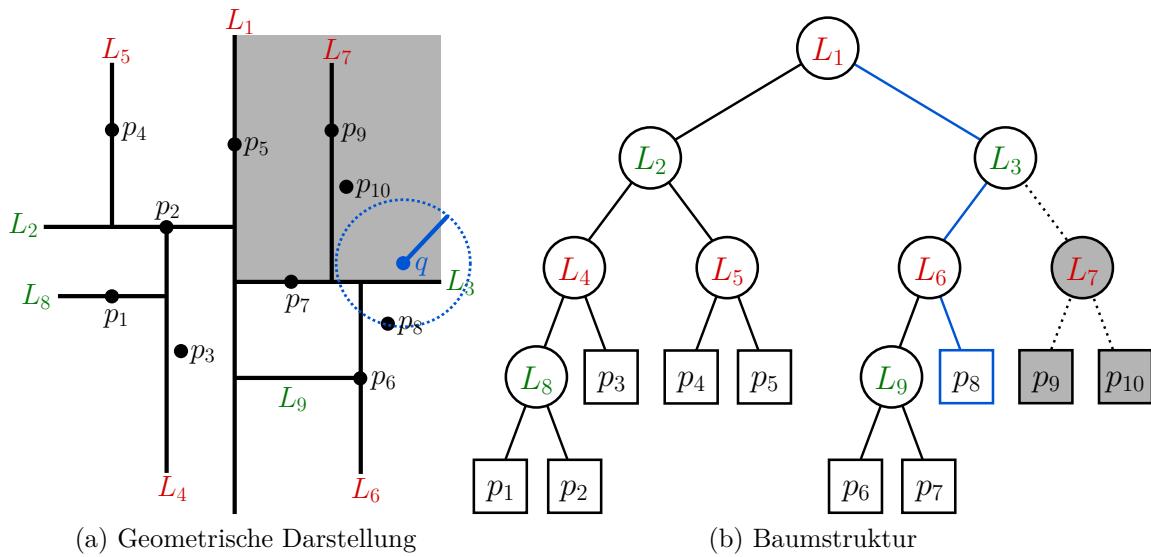


Abbildung C.3: Suche des nächsten Nachbarn im kD-Baum (3. Schritt)

Es verbleibt ein Rechteck, dass von dem Kreis geschnitten wird. Es befindet sich oberhalb von  $L_9$  und links von  $L_6$ . Der Punkt  $p_7$  liegt jedoch nicht näher an dem Anfragepunkt als der Kandidat. Demnach ist der aktuelle Kandidat der nächste Nachbar, da sich keine weiteren Bereiche mit dem Kreis überschneiden. Mit Hilfe

des kD-Baums ist demnach eine schnelle Suche möglich, da nur eine geringe Anzahl an Punkten überprüft werden muss.

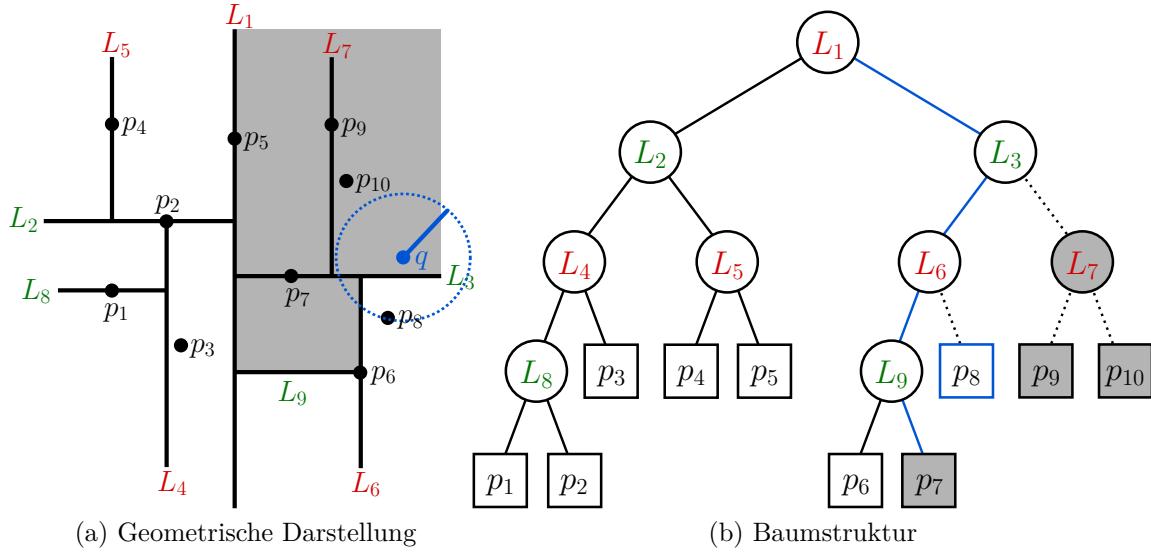


Abbildung C.4: Suche des nächsten Nachbarn im kD-Baum (4. Schritt)

# Literaturverzeichnis

- [AHB87] K.S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, September 1987. (zitiert auf Seite 65, 66 und 97)
- [AMN<sup>+</sup>94] Sunil Arya, David M Mount, Nathan S Netanyahu, Ruth Silverman, and Angela Wu. An optimal algorithm for approximate nearest neighbor searching. In *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 573–582. Society for Industrial and Applied Mathematics, 1994. (zitiert auf Seite vii, 55 und 56)
- [And11] Franz Andert. *Bildbasierte Umgebungserkennung für autonomes Fliegen*. PhD thesis, Technische Universität Braunschweig, Deutschland, 2011. (zitiert auf Seite vi, 22 und 24)
- [Arv90] James Arvo. A simple method for box-sphere intersection testing. In Andrew S. Glassner, editor, *Graphics Gems*, pages 335–339. Academic Press Professional, Inc., San Diego, CA, USA, 1990. (zitiert auf Seite 50)
- [BBC<sup>+</sup>00] Pierre Baldi, Søren Brunak, Yves Chauvin, Claus AF Andersen, and Henrik Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000. (zitiert auf Seite 75)
- [BC94] Jens Berkmann and Terry Caelli. Computation of surface geometry and segmentation using covariance techniques. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(11):1114–1116, 1994. (zitiert auf Seite 40)
- [BCKO08] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag Berlin Heidelberg, Heidelberg, Deutschland, 3rd edition, 2008. (zitiert auf Seite vi und 15)
- [Ben75] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, September 1975. (zitiert auf Seite 14)

- [BM92] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Februar 1992. (zitiert auf Seite 28 und 61)
- [Boa06] Mary L Boas. *Mathematical Methods in the Physical*. John Wiley & Sons., Inc, 2006. (zitiert auf Seite 42)
- [Bun12] Physikalisch-Technische Bundesanstalt. Wie genau gehen die Atomuhren der PTB. Website, Mai 2012. Online verfügbar unter <http://www.ptb.de/cms/index.php?id=fragenzurzeit12>; zuletzt besucht am 22. September 2014. (zitiert auf Seite 8)
- [Cha07] Her-Jye Chang. *Simultaneous localization and mapping algorithms with environmental-structure prediction for single-and multi-robot systems*. ProQuest, 2007. (zitiert auf Seite 75)
- [CM91] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proc. of the IEEE International Conference on Robotics and Automation 1991*, volume 3, pages 2724–2729, Apr 1991. (zitiert auf Seite 30)
- [CTL04] Chi Kin Chow, Hung Tat Tsui, and Tong Lee. Surface registration using a dynamic genetic algorithm. *Pattern Recognition*, 37(1):105–117, 2004. (zitiert auf Seite 27)
- [DP00] Ralf Der and Thomas Pantzer. Echtzeit-lernverfahren zur fehlerbegrenzung und driftkorrektur für odometrie-basierte lokalisationsverfahren. Online verfügbar unter <http://stinfwww.informatik.uni-leipzig.de/~pantec/khepera/diffeq.ps.gz>; zuletzt besucht am 17. Oktober 2014., 2000. (zitiert auf Seite 34)
- [DSB13] David Droseschel, Michael Schreiber, and Sven Behnke. Omnidirectional perception for lightweight uavs using a continuously rotating 3d laser scanner. *UAV-g2013. The international archives of the photogrammetry, remote sensing and spatial information sciences, Rostock, XL-1 W, 2:107–112*, 2013. (zitiert auf Seite 36 und 94)
- [Edw13] Ezzaldeen Edwa. *Novel Approaches for Improved Performance of Inertial Sensors and Integrated Navigation Systems*. PhD thesis, Universität Siegen, Deutschland, 2013. (zitiert auf Seite ix und 11)
- [ELF97] D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-d rigid body transformations: A comparison of four major algorithms. *Mach. Vision Appl.*, 9(5-6):272–290, March 1997. (zitiert auf Seite 65)
- [FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. (zitiert auf Seite 62 und 63)

- [FBF77] Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977. (zitiert auf Seite 55)
- [FBS75] Jerome H. Friedman, Forest Baskett, and Leonard J. Shustek. An algorithm for finding nearest neighbors. *IEEE Transactions on computers*, 24(10):1000–1006, 1975. (zitiert auf Seite 55)
- [FJ08] Eric Feron and Eric N Johnson. Aerial robotics. *Springer Handbook of Robotics*, pages 1009–1029, 2008. (zitiert auf Seite 1)
- [GGT00] M. Greenspan, G. Godin, and J. Talbot. Acceleration of Binning Nearest Neighbour Methods. In *Proceedings of the Vision Interface conference*, pages 337–344, Montreal, QUE, Canada, May 2000. (zitiert auf Seite 54)
- [GIRL03] Natasha Gelfand, Leslie Ikemoto, Szymon Rusinkiewicz, and Marc Levoy. Geometrically stable sampling for the icp algorithm. In *3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on*, pages 260–267. IEEE, 2003. (zitiert auf Seite vi, 39, 42, 43 und 45)
- [Hau78] Felix Hausdorff. *Set theory*. Chelsea Publishing Company, New York, USA, 3rd edition, 1978. (zitiert auf Seite 76)
- [HHN88] Berthold K. P. Horn, H.M. Hilden, and Shariar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127–1135, 1988. (zitiert auf Seite 65)
- [Hon12] Honeywell Aerospace, Phoenix, AZ, USA. *HG1700 Inertial Measurement Unit*, rev e edition, 2012. (zitiert auf Seite 11)
- [Hor87] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987. (zitiert auf Seite 65)
- [HWB<sup>+</sup>13] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, pages 189—206, 2013. (zitiert auf Seite 75)
- [Ich13] Alexandru-Eugen Ichim. Rgb-d handheld mapping and modeling. Master’s thesis, École Polytechnique Fédérale de Lausanne, Lausanne, 2013. (zitiert auf Seite 45 und 59)
- [Inc14] NovAtel Inc. APN-064 IMU Errors and Their Effects, Februar 2014. (zitiert auf Seite 10)
- [Ine99] Inertial Science, Inc., Newbury Park, CA, USA. *Inertial Measurement Unit (IMU), ISIS-IMU*, rev c edition, 1999. (zitiert auf Seite 69)

- [Jek01] Christopher Jekeli. *Inertial navigation systems with geodetic applications*. Walter de Gruyter, 2001. (zitiert auf Seite 11)
- [Ker03] Freddie Kern. *Automatisierte Modellierung von Bauwerksgeometrien aus 3D-Laserscanner-Daten*. PhD thesis, Technische Universität Braunschweig, Deutschland, 2003. (zitiert auf Seite vi und 6)
- [Lin91] Jianhua Lin. Divergence measures based on the shannon entropy. *Information Theory, IEEE Transactions on*, 37(1):145–151, 1991. (zitiert auf Seite 75)
- [Mas02] T. Masuda. Object shape modelling from multiple range images by matching signed distance fields. In *Proc. of First International Symposium on 3D Data Processing Visualization and Transmission*, pages 439–448, 2002. (zitiert auf Seite 26)
- [MB12] R.S. Merali and T.D. Barfoot. Patch map: A benchmark for occupancy grid algorithm evaluation. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012*, pages 3481–3488, Oktober 2012. (zitiert auf Seite 75)
- [Mea82] Donald Meagher. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 19(2):129–147, 1982. (zitiert auf Seite 13)
- [MLD07] Martin Magnusson, Achim Lilienthal, and Tom Duckett. Scan registration for autonomous mining vehicles using 3d-ndt. *Journal of Field Robotics*, 24(10):803–827, 2007. (zitiert auf Seite 31 und 94)
- [Moo90] Andrew William Moore. Efficient memory-based learning for robot control. Technical Report UCAM-CL-TR-209, University of Cambridge, Computer Laboratory, November 1990. (zitiert auf Seite 58)
- [MS08] Kurt Mehlhorn and Peter Sanders. *Algorithms and Data Structures: The Basic Toolbox*. Springer, 2008. (zitiert auf Seite 12 und 13)
- [MS11] Frank Moosmann and Christoph Stiller. Velodyne slam. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 393–398. IEEE, 2011. (zitiert auf Seite 35)
- [MSY96] Takeshi Masuda, K Sakaue, and Naokazu Yokoya. Registration and integration of multiple range images for 3-d model construction. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 1, pages 879–883. IEEE, 1996. (zitiert auf Seite 60)
- [Nü06] Andreas Nüchter. *Semantische dreidimensionale Karten für autonome mobile Roboter*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, Deutschland, 2006. (zitiert auf Seite 25, 56, 57 und 59)
- [NLHS05] Andreas Nüchter, Kai Lingemann, Joachim Hertzberg, and Hartmut Surmann. Heuristic-based laser scan matching for outdoor 6d slam. In

- KI 2005: *Advances in Artificial Intelligence*, pages 304–319. Springer, 2005. (zitiert auf Seite 35)
- [Nov05] NovAtel Inc., Calgary, Kanada. *OEM4 Family Installation and Operation User Manual*, rev 19 edition, 2005. (zitiert auf Seite 68)
- [OC12] National Oceanic and Atmospheric Administration (NOAA) Coastal Services Center. Lidar 101: An Introduction to Lidar Technology, Data, and Applications, 2012. (zitiert auf Seite 4 und 5)
- [OD00] IS. Okatani and K. Deguchi. A method for fine registration of multiple view range images considering the measurement error properties. In *2000. Proc. of the 15th International Conference on Pattern Recognition*, volume 1, pages 280–283, 2000. (zitiert auf Seite 31)
- [OGW<sup>+</sup>13] Johannes Otepka, Sajid Ghuffar, Christoph Waldhauser, Ronald Hochreiter, and Norbert Pfeifer. Georeferenced point clouds: A survey of features and point cloud management. *ISPRS International Journal of Geo-Information*, 2(4):1038–1065, 2013. (zitiert auf Seite 12)
- [PB07] N. Pfeifer and C. Briese. Laser Scanning - Principles and Applications. In *3rd International Exhibition and Scientific Congress on Geodesy, Mapping, Geology, Geophysics, Cadaster GEO-SIBERIA 2007*, pages 93–112, April 2007. (zitiert auf Seite vi, 5 und 6)
- [Pow01] David M. W. Powers. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 201. (zitiert auf Seite 74)
- [Pul99] Kari Pulli. Multiview registration for large data sets. In *3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on*, pages 160–168. IEEE, 1999. (zitiert auf Seite 60 und 61)
- [RL01] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proc. of the Third International Conference on 3-D Digital Imaging and Modeling 2001*, pages 145–152, 2001. (zitiert auf Seite 45, 53 und 59)
- [Rus] Szymon Rusinkiewicz. Derivation of point-to-plane minimization. Online verfügbar unter <http://www.cs.princeton.edu/~smr/papers/icpstability.pdf>; zuletzt besucht am 23. Oktober 2014. (zitiert auf Seite 95)
- [Rus09] Radu Bogdan Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Technische Universität München, Deutschland, 2009. (zitiert auf Seite 40)
- [Sch11] Christian Schaller. *Time-of-Flight-A New Modality for Radiotherapy*. PhD thesis, Universität Erlangen-Nuremberg, Deutschland, 2011. (zitiert auf Seite 6)

- [SJH97] C. Schütz, T. Jost, and H. Hügli. Semi-automatic 3d object digitizing system using range images. In *Computer Vision — ACCV'98*, volume 1351 of *Lecture Notes in Computer Science*, pages 490–497. Springer-Verlag Berlin Heidelberg, 1997. (zitiert auf Seite 60)
- [SMFF07] Joaquim Salvi, Carles Matabosch, David Fofi, and Josep Forest. A review of recent range image registration methods with accuracy evaluation. *Image Vision Comput.*, 25(5), Mai 2007. (zitiert auf Seite 26, 28 und 33)
- [Smi02] Lindsay I Smith. A tutorial on principal components analysis. *Cornell University, USA*, 51:52, 2002. (zitiert auf Seite 16)
- [Str09] Gilbert Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, Wellesley, MA, 2009. (zitiert auf Seite 67)
- [TL94] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 311–318. ACM, 1994. (zitiert auf Seite 44 und 60)
- [TPB06] Rudolph Triebel, Patrick Pfaff, and Wolfram Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2276–2282. IEEE, 2006. (zitiert auf Seite 35)
- [Tuk77] John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, Reading, MA, 1977. (zitiert auf Seite 77)
- [TW04] David H. Titterton and John L. Weston. *Strapdown Inertial Navigation Technology*. The Institution of Electrical Engineers, The American Institute of Aeronautics and Astronautics, 2nd edition, 2004. (zitiert auf Seite 10)
- [Vel12] Velodyne LiDAR, Inc, Morgan Hill, CA, USA. *HDL-32E, User's Manual and Programming Guide*, rev e edition, November 2012. (zitiert auf Seite 69)
- [WHB<sup>+</sup>10] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, AK, USA, Mai 2010. Software available at <http://octomap.github.com>. (zitiert auf Seite vi, 13 und 14)
- [Win13] Stefan Winkvist. *Low computational SLAM for an autonomous indoor aerial inspection vehicle*. PhD thesis, University of Warwick, Großbritannien, 2013. (zitiert auf Seite 36)
- [WSV91] Michael W. Walker, Lejun Shao, and Richard A. Volz. Estimating 3-d location parameters using dual number quaternions. *CVGIP: Image Underst.*, 54(3):358–367, Oktober 1991. (zitiert auf Seite 65)

- [Zog11] Jean-Marie Zogg. *GPS und GNSS: Grundlagen der Ortung und Navigation mit Satelliten*. Thalwil, u-blox, Oktober 2011. (zitiert auf Seite vi, ix, 7, 8, 9 und 10)
- [ZSN03a] Timo Zinßer, Jochen Schmidt, and Heinrich Niemann. Performance analysis of nearest neighbor algorithms for icp registration of 3-d point sets. In *Vision, Modeling and Visualization 2003*, pages 199–206, 2003. (zitiert auf Seite 54, 55, 56 und 57)
- [ZSN03b] Timo Zinßer, Jochen Schmidt, and Heinrich Niemann. A refined icp algorithm for robust 3-d correspondence estimation. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages 695–698. IEEE, 2003. (zitiert auf Seite 60)
- [ZZ09] Min-Ling Zhang and Zhi-Hua Zhou. Multi-instance clustering with applications to multi-instance prediction. *Applied Intelligence*, 31(1):47–68, 2009. (zitiert auf Seite 77)

# Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Braunschweig, den 21. November 2014

---

Sebastian Stellmacher