

Institut für parallele und verteilte Systeme

Abteilung Bildverstehen

Universität Stuttgart

Universitätsstraße 38

D-70569 Stuttgart

Diplomarbeit Nr. 3363

**Analyse paarweiser und  
punktbasierter  
Registrierungsverfahren auf  
hochauflösten, sphärischen  
3D-Punktwolken**

Simon Kontschak

**Studiengang:** Informatik

**Prüfer:** PD. Dr. rer. nat. Michael Schanz

**Betreuer:** PD. Dr. rer. nat. Michael Schanz  
Dipl.-Inf. Stefan Bertele

**begonnen am:** 12. Juli 2012

**beendet am:** 11. Januar 2012

**CR-Klassifikation:** I.2.10, I.3.7, I.4.3, I.4.7, I.4.8, I.5.4



## **Kurzfassung**

**E**INE möglichst realistische und somit dreidimensionale Repräsentation von Szenen und Objekten ist Grundlage heutiger Echtzeit- und Vermessungssysteme. Damit erhält dieses Thema auch in der Computer Vision Einzug. Die Datenerhebung findet meist mittels Laserscanner statt indem ein, um zwei orthogonal zueinander stehende Achsen, rotierender Laserstrahl die Umgebung punktweise abtastet. Die Rundumaufnahmen einer Farbkamera können zusätzlich Farbinformationen liefern, wie dies bei Laserscannern der Firma FARO der Fall ist.

Um eine vollständige Beschreibung von Szene und Objekt zu erhalten, muss dieser Vorgang von verschiedenen Standpunkten aus wiederholt durchgeführt werden. Aufnahmen aus unterschiedlichen Blickwinkeln, die Überschneidungen beinhalten, sind das Ergebnis. Im Rahmen dieser Diplomarbeit sollen Verfahren zur automatisierten Ausrichtung zweier Aufnahmen, unter Verwendung der Punktinformationen untersucht werden. Der Schwerpunkt liegt dabei auf punktbasierten Verfahren die sich nicht der Extraktion geometrischer Merkmale bedienen.



# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>11</b>
1.1. Motivation . . . . .	12
1.2. Aufgabenstellung . . . . .	13
1.3. Gliederung . . . . .	14
<b>2. Grundlagen</b>	<b>15</b>
2.1. Laserscanner . . . . .	15
2.1.1. Funktionsprinzip des Laserscanners . . . . .	15
2.1.2. Sensoren . . . . .	17
2.1.3. Einsatzbereich . . . . .	19
2.1.4. FARO SCENE . . . . .	22
2.2. Punktwolken . . . . .	22
2.2.1. Oberflächen-Punktwolken . . . . .	23
2.2.2. Filter . . . . .	23
2.3. Datenstrukturen . . . . .	24
2.3.1. 1D-Array . . . . .	24
2.3.2. 2D-Array . . . . .	25
2.3.3. K-dimensionaler-Baum (KD-Tree) . . . . .	25
<b>3. Registrierung von Punktwolken</b>	<b>27</b>
3.1. Mathematische Definition . . . . .	28
3.2. Klassifikation . . . . .	29
3.2.1. Grobregistrierung . . . . .	30
Künstliche Merkmale . . . . .	30
Geometrische Merkmale . . . . .	31
Weitere Verfahren . . . . .	33
3.2.2. Feinregistrierung . . . . .	34
<b>4. Fehlermaße</b>	<b>37</b>
4.1. Gesamtfehler . . . . .	37
4.2. Translationsfehler . . . . .	38
4.3. Rotationsfehler . . . . .	38
<b>5. Testumgebung</b>	<b>41</b>
5.1. Funktionsumfang . . . . .	41
5.2. Funktionsweise . . . . .	42
5.3. Testszenarien . . . . .	43

<b>6. Sub-Sampling</b>	<b>47</b>
6.1. Entfernungsunabhängiges Sub-Sampling	47
6.2. Entfernungsabhängiges Sub-Sampling	49
<b>7. Point Feature Histogram (PFH)</b>	<b>53</b>
7.1. Funktionsprinzip	53
7.2. Erweiterungen	55
7.2.1. Punktwahl	55
7.2.2. PFH Größe	56
7.2.3. Einflussbereich	56
7.2.4. Metrik	57
7.3. Analysierte Variante	58
<b>8. Spin-Images (SI)</b>	<b>61</b>
8.1. Funktionsprinzip	61
8.2. Erweiterungen	63
8.3. Analysierte Variante	64
<b>9. Iterative Closest Point (ICP)</b>	<b>65</b>
9.1. Funktionsprinzip	65
9.2. Varianten	66
9.2.1. Punktwahl	66
9.2.2. Korrespondenzfindung	67
9.2.3. Korrespondenzgewichtung	67
9.2.4. Korrespondenzverwerfung	68
9.2.5. Fehlermaß und Minimierung	69
9.3. Analysierte Variante	70
<b>10. Normal Distribution Transformation (NDT)</b>	<b>73</b>
10.1. Funktionsprinzip	73
10.2. Varianten	74
10.3. Analysierte Variante	75
<b>11. Analyse</b>	<b>77</b>
11.1. Point Feature Histogram	77
11.1.1. Normalenradius	77
11.1.2. Merkmalsradius	79
11.1.3. Abweichung Durchschnittshistogramm	80
11.1.4. Korrespondenzgrenzwert	82
11.1.5. Konvergenz	83
11.1.6. Effizienz	85
11.1.7. Stabilität	86
11.2. Spin-Images	87
11.2.1. Normalenradius	87
11.2.2. Suchradius	89

11.2.3. Support-Winkel . . . . .	90
11.2.4. Abweichung Durchschnittshistogramm . . . . .	91
11.2.5. Korrespondenzgrenzwert . . . . .	92
11.2.6. Konvergenz . . . . .	94
11.2.7. Effizienz . . . . .	95
11.2.8. Stabilität . . . . .	97
11.3. Iterative Closest Point . . . . .	97
11.3.1. Fehlergrenzwert . . . . .	97
11.3.2. Transformationsgrenzwert . . . . .	98
11.3.3. Iterationen . . . . .	99
11.3.4. Korrespondenzgrenzwert . . . . .	100
11.3.5. Konvergenz . . . . .	101
11.3.6. Effizienz . . . . .	102
11.3.7. Stabilität . . . . .	104
11.4. Normal-Distribution Transformation . . . . .	104
11.4.1. Iterationen . . . . .	105
11.4.2. Transformationsgrenzwert . . . . .	106
11.4.3. Zellgröße . . . . .	107
11.4.4. Ausreißerrate . . . . .	107
11.4.5. Konvergenz . . . . .	108
11.4.6. Effizienz . . . . .	110
11.4.7. Stabilität . . . . .	112
<b>12. Fazit und Ausblick</b>	<b>113</b>
<b>13. Zusammenfassung</b>	<b>117</b>
<b>A. Transformationsbestimmung mittels Singulärwertzerlegung</b>	<b>119</b>
<b>B. Normalen-Berechnung mittels Hauptkomponenten Analyse (PCA)</b>	<b>121</b>
<b>C. Initiale Ausrichtungen</b>	<b>123</b>
<b>Literaturverzeichnis</b>	<b>127</b>
<b>Danksagung</b>	<b>133</b>

# Abbildungsverzeichnis

---

1.1.	Vergleich natürliche und künstliche Umgebung	12
2.1.	FARO® Focus <sup>3D</sup> Laserscanner	16
2.2.	Funktionsprinzip FARO® Focus <sup>3D</sup> Laserscanner	17
2.3.	Einsatzbereich Forstwirtschaft	20
2.4.	Einsatzbereich Denkmalschutz	20
2.5.	Einsatzbereich Vermessung	21
2.6.	Einsatzbereich Forensik	21
2.7.	Punktwolken Software SCENE	22
2.8.	Oberflächen-Punktwolke	23
3.1.	Registrierung von Punktwolken	28
3.2.	Künstliche Marker	31
5.1.	Testszenarien	44
6.1.	Diskretes Sub-Sampling	48
6.2.	Räumliches Sub-Sampling Zeit-Punkt-Zellgrößen-Vergleich	49
6.3.	Räumliches Sub-Sampling-Szenen	50
7.1.	Point Feature Histogramm Deskriptor	54
7.2.	Darboux Frame	55
8.1.	Spin Image Deskriptor	62
9.1.	Korrespondenz-Findung	68
9.2.	Korrespondenzverwerfung	69
11.1.	FPFH Normalenradius	78
11.2.	FPFH Merkmalsradius	80
11.3.	FPFH Abweichung Durchschnittshistogramm	81
11.4.	FPFH Korrespondenzgrenzwert	83
11.5.	FPFH Konvergenz	84
11.6.	FPFH Effizienz	86
11.7.	SI Normalenradius	88
11.8.	SI Suchradius	89
11.9.	SI Support-Winkel	91
11.10.	SI Abweichung Durchschnittshistogramm	92

11.11. SI Korrespondenzgrenzwert . . . . .	93
11.12. SI Konvergenz . . . . .	95
11.13. SI Effizienz . . . . .	96
11.14. ICP Fehlergrenzwert . . . . .	98
11.15. ICP Transformationsgrenzwert . . . . .	99
11.16. ICP Iterationen . . . . .	100
11.17. ICP Korrespondenzgrenzwert . . . . .	101
11.18. ICP Konvergenz . . . . .	102
11.19. ICP Effizienz . . . . .	103
11.20. NDT Iterationen . . . . .	105
11.21. NDT Transformationsgrenzwert . . . . .	106
11.22. NDT Zellgröße . . . . .	107
11.23. NDT Ausreißerrate . . . . .	108
11.24. NDT Konvergenz . . . . .	109
11.25. NDT Effizienz . . . . .	110
11.26. NDT Effizienz . . . . .	111
12.1. Übersicht Laufzeit . . . . .	113
12.2. Übersicht Genauigkeit . . . . .	114
C.1. Globale Ausrichtung . . . . .	124
C.2. Initiale Ausrichtung . . . . .	125

# Tabellenverzeichnis

---

2.1.	Scanauflösungen des Focus <sup>3D</sup> . . . . .	16
5.1.	Korrekte Transformationen der Testszenarien . . . . .	45
C.1.	Initiale Fehler . . . . .	126

# 1. Einleitung

Die dreidimensional hochauflöste Repräsentation von Szenen und Objekten ist heute gefragter als niemals zuvor. Realisiert wird dies mit Hilfe moderner Laserscanner. Der rotierende Laserstrahl tastet dabei punktweise die Distanzen der Umgebung in alle Richtungen ab. Eine integrierte Digitalkamera erzeugt zusätzlich Farbbilder die Farbinformationen zu den einzelnen Punkten, liefern. Aus den so akquirierten Daten wird am Computer mittels Software eine gefärbte 3D-Punktwolke generiert.

Um eine Szene oder ein Objekt vollständig erfassen zu können müssen mehrere Aufnahmen aus verschiedenen Blickwinkeln erzeugt werden. Diese Tatsache erfordert eine spätere Überführung und Ausrichtung in ein gemeinsames Koordinatensystem. Der hohe Zeitaufwand und die mangelnde Genauigkeit einer manuellen Überführung verlangen automatisierte Verfahren. Diese verwenden meist charakteristische Merkmale, wie zum Beispiel künstliche Marker, die in der Szene platziert und aufgenommen werden. Eine Identifikation gleicher Marker in verschiedenen Punktwolken gibt dann Aufschluss über deren globale Zusammenhänge.

Eine starke Preisreduzierung und die stets akkurate werdende Arbeitsweise der Geräte, führen zu einem immer größer werdenden Anwendungsbereich. So werden in der Architektur Gebäude vermessen, in Produktionsstätten sichert man die Qualität, Lagerbestände werden überwacht und neben Reverse-Engineering von Bauteilen kommen sie auch bei der Tatortssicherung in der Forensik zum Einsatz. In vielen dieser Einsatzbereiche können keine künstlichen Marker eingesetzt werden, da diese die Szene verändern würden. Ist das der Fall, werden stattdessen geometrische Merkmale, wie Linien, Eckpunkte oder Flächen, aus den akquirierten Daten extrahiert und als zusammengehörig identifiziert. Natürliche Umgebungen enthalten oft nur sehr wenige oder keine dieser geometrischen Merkmale, wie in Abbildung 1.1, demonstriert wird.

Betrachtet man das Problem der Ausrichtung von 3D-Punktwolken in Echtzeit-Systemen, wie z.B. der Roboternavigation oder Objekterkennung, werden punktbasierte Merkmale herangezogen oder einzelne Punkte miteinander verglichen. Das hat den Vorteil, dass sie unabhängig von der betrachteten Szene operieren können. Einige dieser Verfahren sind sogar in der Lage Vorabinformationen über eine wahrscheinliche Lage der Punktwolken zu verarbeiten. Dadurch kann die Problemkomplexität reduziert werden. Bei einer großen Anzahl von Punkten sind diese Verfahren jedoch sehr rechenaufwändig und kommen daher meist nur auf Punktwolken geringer Dichte zum Einsatz.

Sensoren zur Gewinnung von Vorabinformationen, über Position und Ausrichtung erzeugter Punktwolken, sind in heutigen Laserscannern häufig integriert. Diese werden jedoch meist nur zur Erkennung einer fehlerhaften Ausrichtung oder der Verbesserung des Ergebnisses,

## 1. Einleitung

---

eingesetzt. Mit dem Einsatz von Sensoren, neuen Techniken der Datenreduktion und mit den immer leistungsfähiger werdenden Computersystemen, könnte es möglich sein punktbasierte Algorithmen zur Ausrichtung hochaufgelöster 3D-Punktwolken zu verwenden. Die daraus resultierende Möglichkeit der automatisierten Verarbeitung von Szenen bei welchen bisher eingesetzte Verfahren versagen, würde zu einer Erschließung neuer Anwendungsgebiete führen.



**Abbildung 1.1.: Vergleich natürliche und künstliche Umgebung** - *Links:* Harte Kanten und große Flächen bieten die Möglichkeit aussagekräftige geometrische Merkmale zu extrahieren. *Rechts:* Überwiegend vertreten sind Rundungen. Aussagekräftige, vergleichbare Flächen oder Kanten sind kaum vorhanden.

### 1.1. Motivation

Die Entwicklung punktbasierter Algorithmen, zur Ausrichtung von Punktwolken, wird seit geraumer Zeit vorangetrieben. Dies wird dadurch begründet, dass sie häufig zur Navigation von Robotern sowie zur Erkennung von Objekten, beides wesentliche Anwendungsgebiete der heutigen Zeit, eingesetzt werden. Das Ergebnis sind hoch-optimierte Algorithmen, die spezifische Eigenschaften von Punktwolken ausnutzen. Oft führt dies zu einer deutlich effizienteren Verarbeitung der Daten, als dies bei Verfahren möglich ist, die auf der Grundlage geometrischer Merkmale arbeiten. Erschwerend kommt hinzu, dass die Genauigkeit einer Ausrichtung hier auf die Präzision, der erkannten Merkmale, beschränkt ist.

Bei der Verwendung geometrischer Merkmale entsteht außerdem oft das Problem, dass zu viele oder gar zu wenige, oftmals keine, Merkmale identifiziert werden. Eine Über- bzw. Unterbestimmung des Problems und somit eine fehlerhafte oder gar ausbleibende Ausrichtung der Punktwolken, ist die Folge.

Die Verwendung und Untersuchung punktbasierter Algorithmen erfolgt häufig auf Punktwolken geringer Dichte oder sehr geringer Punktmengen. Dies führt schnell zu der Annahme, dass diese Art von Algorithmen nicht für die Ausrichtung hochaufgelöster Punktwolken geeignet ist.

Die meisten zur Erzeugung hochaufgelöster Punktwolken verwendeten Geräte wie z.b der FARO Focus3D besitzen Sensoren, wie Inklinometer, Altimeter, GPS und Kompass. Mit Hilfe dieser können Informationen über die räumlichen Beziehungen der einzelnen Punktwolken zueinander ermittelt werden. Einige der punktbasierten Verfahren sind in der Lage diese zu berücksichtigen. Neben der generellen Reduktion der Problemkomplexität kann dadurch ein sehr zielgerichtetes Suchen der korrekten Lösung erfolgen. Das Ausbleiben oder Ermitteln eines, von Grund auf falschen, Ergebnisses kann dadurch umgangen werden.

Unter Berücksichtigung dieser Tatsachen, könnte es möglich sein, automatisierte Verfahren zur Ausrichtung hochaufgelöster Punktwolken zu verwenden und, unabhängig von der betrachteten Szene eine korrekte Lösung, in angemessener Zeit, zu finden.

## 1.2. Aufgabenstellung

In dieser Diplomarbeit soll das Problem der automatisierten, paarweisen Ausrichtung hochaufgelöster Punktwolken unter der Verwendung punktbasierter Verfahren untersucht werden. Diese sollen sich nicht der Extraktion geometrischer Merkmale oder künstlicher Marker bedienen.

Dabei soll betrachtet werden, inwiefern zusätzliche Informationen, die aus Richtungs- und Lagesensoren gewonnen werden, dabei hilfreich sind. Hierfür ist es notwendig punktbasierte Algorithmen ihrer Lösungsstrategie entsprechend, zu klassifizieren und deren Stellvertreter zu ermitteln. Diese sollen dann in einer Testumgebung, in Zusammenarbeit mit der Firma FARO, in die Software SCENE (siehe Abschnitt 2.1.4) implementiert werden. Abschließend ist eine Analyse der Algorithmen in verschiedenen Szenarien durchzuführen. Diese soll Fragen zu folgenden Anforderungen beantworten:

- **Konvergenz** - Das Verfahren findet eine Lösung, die einer Verbesserung der Ausgangssituation entspricht.
- **Stabilität** - Das Verfahren konvergiert für verschiedene Ausgangssituationen.
- **Effizienz** - Das Verfahren findet in angemessener Zeit eine Lösung.
- **Reproduzierbarkeit** - Das Verfahren findet für identische Ausgangssituationen stets die selbe Lösung.

### 1.3. Gliederung

Die Arbeit ist in folgender Weise gegliedert:

**Kapitel 2 – Grundlagen** - erläutert einführenden Begriffe und stellt die verwendete Hardware bzw. Software vor.

**Kapitel 3 – Registrierung von Punktwolken** - definiert das betrachtete Problem und klassifiziert die dafür vorhandenen Ansätze. Dabei wird auch die Wahl der zur genaueren Untersuchung herangezogenen Algorithmen motiviert.

**Kapitel 4 – Fehlermaße** - beschreibt die zur Evaluierung der erzielten Ergebnisse verwendeten Fehlermaße einer Transformation.

**Kapitel 5 – Testumgebung** - erörtert die zur Analyse verwendeten Rahmenbedingungen. Enthalten ist eine Beschreibung einzelner Testszenarien sowie der Ablauf der durchgeführten Untersuchungen.

**Kapitel 6 – Sub-Sampling** - stellt unterschiedliche Methoden zur Datenreduktion hochaufgelöster 3D-Punktwolken vor. Außerdem werden dabei Vor- und Nachteile aufgezeigt und erläutert.

**Kapitel 7 – Point Feature Histogram (PFH)** - vermittelt den Point Feature Histogramm Ansatz, erläutert dessen Funktionsweise und geht auf mögliche Erweiterungen ein. Abschließend wird die zur Analyse bereitgestellte Variante erläutert.

**Kapitel 8 – Spin-Images (SI)** - beleuchtet die Arbeitsweise und dabei notwendige Modifikationen des Spin-Image Algorithmus.

**Kapitel 9 – Iterative Closest Point (ICP)** - erklärt das Funktionsprinzip des Iterative Closest Point Algorithmus und stellt einige unterschiedliche Varianten dessen vor. Anschließend wird die implementierte Variante vorgestellt.

**Kapitel 10 – Normal Distribution Transformation (NDT)** - gibt Einblick in die generelle Verfahrensweise des Normal Distribution Transformation Ansatz und erläutert mögliche Modifikationen.

**Kapitel 11 – Analyse** - präsentiert die Ergebnisse durchgeführter Analysen der vorgestellten Varianten von PFH, SI, ICP und NDT.

**Kapitel 12 – Fazit und Ausblick** - vergleicht die einzelnen Registrierungsverfahren untereinander und gibt Ausblick auf weitere Untersuchungen.

**Kapitel 13 – Zusammenfassung** - resümiert die in dieser Arbeit behandelten Themen.

## 2. Grundlagen

**I**n diesem Kapitel werden die verwendeten Technologien und Werkzeuge vorgestellt. Dies soll dem Leser einen Einstieg in das Thema ermöglichen. Außerdem sind Grundlagen für die im späteren Verlauf erläuterten Zusammenhänge notwendig. Eine allumfassende Betrachtung des Themengebietes ist im Rahmen dieser Arbeit weder möglich noch gewünscht.

Der erste Teil dieses Kapitels befasst sich mit der Akquirierung der Daten. Dazu wird der in dieser Diplomarbeit verwendete Laserscanner und dessen Funktionsweise beschrieben. Anschließend werden die erfassten Daten in ihrer Form als Punktwolken genauer erläutert um deren Eigenschaften zu verstehen. Der darauf folgende Abschnitt befasst sich mit verschiedenen Repräsentationen einer Punktwolke im Arbeitsspeicher des Computers. Diese sind an vielen Stellen für das Laufzeitverhalten der Algorithmen relevant. Zuletzt soll ein Einblick in die Methoden zur Reduktion der Punktmengen gegeben werden.

### 2.1. Laserscanner

Der dieser Diplomarbeit zugrunde liegende Laserscanner wurde von der Firma FARO entwickelt und trägt die Bezeichnung FARO® Focus<sup>3D</sup> (siehe Abbildung 2.1). Dessen Aufbau, Eigenschaften, Funktionsprinzip und Einsatzbereiche werden in diesem Abschnitt behandelt. Außerdem wird Einblick in die ebenfalls von der Firma FARO® entwickelte Software SCENE gegeben werden.

#### 2.1.1. Funktionsprinzip des Laserscanners

In Abbildung 2.1 ist ein Produktfoto des Focus<sup>3D</sup> zu sehen. Die Ausmaße des Focus<sup>3D</sup> sind 240 × 200 × 100 mm. Auf der rechten Seite des Gehäuses ist ein Touchscreen zu sehen über welchen Konfigurationen wie z.b die Auflösung einer Punktwolke vorgenommen werden können. Die unterstützten Auflösungen sind in Tabelle 2.1 festgehalten. Diese beziehen sich auf eine maximale Reichweite von 120 Metern. Um den Scanvorgang zu starten muss lediglich ein Button auf dem Touchscreen gedrückt werden.

In der linken Seite des Gerätes wird der Laserstrahl erzeugt und über den um 45° geneigten Spiegel im Zentrum des Laserscanners in jede beliebige Richtung der Umgebung entsendet. Dazu rotiert der Spiegel horizontal um 360°, während sich der Scanner langsam, vertikal um 180° seiner eigenen Achse dreht (siehe Abbildung 2.2). Durch die Kombination dieser Rotationen wird eine kugelförmige Umgebung des Laserscanner abgetastet. Die kompakte

## 2. Grundlagen

---



**Abbildung 2.1.: FARO® Focus<sup>3D</sup> Laserscanner** - Der von der Firma FARO® entwickelte Laserscanner Focus<sup>3D</sup>, ist über einen Touchscreen bedienbar. Laser und Kamera sind in der linken Hälfte des Gerätes untergebracht.[[FAR12](#)]

und schlanke Bauweise des Scanners erlaubt ein vertikales Sichtfeld von 305°. Der Laser arbeitet mit einer Wellenlänge von 905 Nanometern und gehört mit einer Leistung von 20 Miliwatt zu den nicht augensicheren Lasern der Klasse 3R.

Die Abtastung der Umgebung erfolgt diskret, da jeder Punkt von einem Laserstrahl gemessen wird. Pro Sekunde werden dabei 976.000 Strahlen abgefeuert und ebenso viele Punkte gemessen. Ein ausgesandter Laserstrahl wird vom ersten Hindernis diffus reflektiert. Teile des reflektierten Strahles werden vom Laserscanner wieder aufgefangen. Die Distanz des

Auflösung	Punkte Zeile × Punkte Spalte	Winkelauflösung
1/1	17067 × 40960	0,009° /Pixel
1/2	8534 × 20480	0,018° /Pixel
1/4	4267 × 10240	0,036° /Pixel
1/8	2133 × 5120	0,072° /Pixel
1/16	1067 × 2560	0,144° /Pixel

**Tabelle 2.1.: Scanauflösungen des Focus<sup>3D</sup>** - Die hier aufgeführten Auflösungen werden vom Focus<sup>3D</sup> unterstützt. Eine Zeile entspricht einer vertikalen Rotation des Spiegels, eine Spalte einer halben, horizontalen Rotation des Gerätes.

zu messenden Punktes wird mittels Phasenverschiebung des modulierten Laserstrahles errechnet. Eine hochfrequente Sinusschwingung erlaubt dabei Genauigkeiten von bis zu 2 mm. Niedere Frequenzen ermöglichen das Messen großer Entfernung und sind notwendig um Mehrdeutigkeiten aufzulösen. Die X- und Y-Koordinate zu der gemessenen Distanz wird über die Winkelpositionen des Spiegels und des Gerätes ermittelt. Ein 3D-Punkt ist das Resultat.

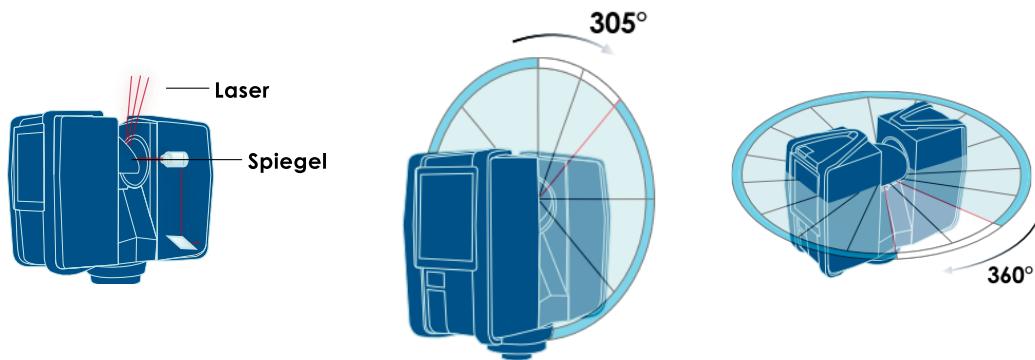


Abbildung 2.2.: Funktionsprinzip FARO® Focus<sup>3D</sup> Laserscanner - *Links:* Der Laserstrahl wird über den Spiegel in die Umgebung entsendet. *Mitte:* Die vertikale Rotation des Spiegels lenkt den Laserstrahl in vertikalen Kreisen um den Laserscanner herum ab. *Rechts:* Die zusätzliche horizontale Rotation des Gerätes gewährleistet eine vertikale Abtastung der Umgebung .[FAR12]

Ebenfalls in der linken Seite des Gerätes (siehe Abbildung 2.1), ist eine Digitalkamera mit Blick auf den Spiegel montiert. Während erneuter horizontaler Rotationen des Spiegels und einer vertikalen Rotation des Scanners um weitere 180°, erzeugt diese 84 Farbbilder kombiniert mit einer Auflösung von 70 Mio. Pixeln. Aus ihnen wird in einem späteren Schritt mittels Software (siehe Abschnitt 2.1.4) ein Panoramabild erzeugt, das die gesamte Umgebung des Laserscanner umfasst. Anschließend kann die Farbinformation der Bildpixel auf die vom Laser gemessenen 3D Punkte projiziert werden um die Punktwolke zu kolorieren.

Die akquirierten Daten werden auf einer SD-Karte gespeichert, deren Slot sich auf der rechten Seite des Gerätes unterhalb des Touchscreens befindet. Mit ihr können die erzeugten Punktwolken auf den Computer übertragen werden um weitere Verarbeitungsschritte durchzuführen.

### 2.1.2. Sensoren

Versucht man die Lage eines Objektes abhängig von einer Bezugsposition im 3D-Raum vollständig zu beschreiben, das entspricht der Problematik des Ausrichtens zweier Punktwolken zueinander (siehe Kapitel 3), sind 6 Freiheitsgrade notwendig. Dabei handelt es sich um die Verschiebung in X-, Y- und Z- Richtung, sowie die Rotationen um X-, Y-, und Z-Achse.

## 2. Grundlagen

---

Einige der im Späteren betrachteten Algorithmen sind in der Lage, diese Art der Information zu verarbeiten. Dies führt zu einer Reduktion der Freiheitsgrade und somit der Problemkomplexität. Im Folgenden wird daher ein Überblick der im Focus<sup>3D</sup> verfügbaren sowie weiteren Sensoren zur Lage- und Positionsbestimmung vorgestellt. Dies lassen sich allgemein in zwei Kategorien fassen, Lage- und Positionsbestimmung.

Die Z-Achse ist dabei als die Achse definiert, um welche sich der Scanner beim Scanvorgang dreht. X- und Y-Achse bilden dann die Ebene auf welcher die Z-Achse orthogonal steht.

- **Lagesensoren** Diese geben Aufschluss über die Neigung des Laserscanners. Die Neigung ist im 3D-Raum eine Bewegung die mittels Rotationen um die Koordinatenachsen beschrieben werden kann.

Ein im Focus<sup>3D</sup> verfügbarer Sensor ist das Inklinometer. Dabei handelt es sich um einen Zweiachsen-Kompensator. Dieser ist in der Lage, die Rotationen um zwei verschiedene Achsen zu messen, die X und Y-Achse. Sind die Rotationen um 2 Achsen bekannt, reduziert sich die Problemkomplexität auf 4 Freiheitsgrade. Diese sind die Verschiebung in X-, Y- und Z- Richtung sowie die Rotation um die Z-Achse. Bei einer Genauigkeit von  $\pm 0,015^\circ$  [FAR12] können diese Messergebnisse ohne zusätzliche Korrekturen verwendet werden.

Bei einem weiteren, im Focus<sup>3D</sup> verfügbaren Lagesensor handelt es sich um den Kompass. Dieser orientiert sich am Erdmagnetfeld und nutzt als Bezugspunkt den geografischen Nordpol. Der Einsatz eines Kompass löst demnach einen Freiheitsgrad der Drehung. Im System des Laserscanner liefert er Informationen bezüglich der Rotation um die Z-Achse. Bei einer Genauigkeit von  $\pm 20^\circ$  [FAR12] ist diese Information sehr unsicher. Externe, omnipräsente Magnetfelder, meist verursacht durch benachbarte Elektrogeräte, überdecken das Erdmagnetfeld was zu falschen Informationen führen kann.

- **Positions-Sensoren** Sie dienen im allgemeinen dazu, die Position eines Objektes im 3D-Raum zu bestimmen. Dies entspricht den Verschiebungen in Richtung der Koordinatenachsen (X, Y, Z).

Der wohl bekannteste Ansatz ist die Satellitennavigation wie z.B. GPS (NAVSTAR GPS) [Ameo08], Galileo, Compass etc. Bezugspunkte sind dabei um die Erde kreisende Satelliten. Aus deren ausgesandten und überlagerten Signalen kann die Position eines Objektes auf der Erde bestimmt werden. Sie lösen daher 3 Freiheitsgrade und reduzieren das betrachtete Problem auf Rotationen um die Koordinatenachsen. Die Genauigkeit dieser Information hängt stark von der Stärke des erhaltenen Signals und der verwendeten Technik ab. Allgemein kann dabei eine Genauigkeit von  $\pm 2 - 7m$  [Ameo08, JRE<sup>+</sup>04] unter freiem Himmel erwartet werden.

Nur die Höhe eines Objektes (Z-Koordinate) bestimmen arbeiten Höhenmesser (Altimeter). Realisiert wird dies über das Messen des Luftdruckes. Bezogen auf die Position des Laserscanner wird eine Verschiebung in Richtung der Z-Achse gemessen. Sein Beitrag zur Reduktion der Komplexität beschränkt sich auf diesen einen Freiheitsgrad.

Dieser kann mit einer Genauigkeit von  $\pm 5$  m [FAR12] bestimmt werden. Ein Altimeter ist im Focus<sup>3D</sup> enthalten.

Die relativ zu einem bestimmten Ausgangspunkt zurückgelegte Trajektorie zu erfassen ist eine weiter Methode, um eine Position zu bestimmen. Dies wird als Odometrie bezeichnet [Hoco5, Nüo2]. Der Fehler dieser Information ist abhängig von der Länge der zurückgelegten Strecke und bewegt sich im Meter-Bereich. Bestimmt werden kann dabei die Verschiebung in Richtung der X- und Y- Koordinate, wodurch zwei Freiheitsgrade der Translation gelöst sind.

Betrachtet man die Kombination aller hier aufgeführten Sensoren, könnte der Schluss gezogen werden, dass Informationen für alle Freiheitsgrade bestimmt werden können. Die Lagesensoren liefern Informationen bezüglich der Rotationen um X-, Y-, Z-Achse, die Positions-Sensoren lösen die Verschiebung in selbige. Ungenauigkeiten der Sensoren, sowie verrauschte oder ganz ausbleibende Signale führen jedoch dazu, dass ein Großteil dieser Information lediglich als Hilfestellung benutzt werden kann.

### 2.1.3. Einsatzbereich

Die wahrheitsgetreue, dreidimensionale Repräsentation einer Szene und die sich hierdurch ergebenden Möglichkeiten, machen 3D-Laserscanner für eine Vielzahl verschiedener Anwendungen interessant. Besonders beliebt sind dabei Anwendungsbereiche, bei welchen die Visualisierung und Vermessung von Objekten oder ganzer Umgebungen eine Rolle spielt. Sinkende Anschaffungskosten sorgen dafür, dass 3D-Laserscanner auch in Anwendungsbereiche Einzug halten, bei welchen es nicht nur darum geht kostenintensive, industrielle Prozesse zu optimieren. Im Folgenden werden einige typische Einsatzbereiche vorgestellt.

**Forstwirtschaft** Zur nachhaltigen Forstwirtschaft gehört die Dokumentation der Waldbestände. Mit Hilfe von 3D- Laserscannern ist es möglich diese flächendeckend und ohne erheblichen Zeitaufwand zu erstellen. Die Dokumentation kann dann in die Software voll-automatischer Holzernter eingespielt werden [Tre12]. Zur Abholzung freigegeben Bäume können darin markiert werden. Sind Bäume gefällt oder durch Unwetterschäden nicht mehr vorhanden, kann die Dokumentation durch das Entfernen der entsprechenden Daten stets aktuell gehalten werden. Außerdem können Kollateralschäden, verursacht durch das Schlagen von Zufahrtswegen, im Voraus ermittelt und minimal gehalten werden.

## 2. Grundlagen

---



Abbildung 2.3.: Einsatzbereich Forstwirtschaft - Links: Akquirierte Daten eines Waldes. Rechts: Zur Abholzung freigegebene und für die Bestandspflege zu erhaltende, markierte Bäume. [Tre12]

**Denkmalschutz** Um Denkmäler zu schützen und zu erhalten sind häufig Restaurierungen notwendig. Mit Hilfe von 3D-Laserscannern ist es möglich Objekte oder ganze Gebäude zu erfassen, ohne dabei empfindliche Bereiche zu strapazieren. Dies ist auf Grund der kontaktfreien Erfassung der Umgebung möglich. Anschließend kann eine ausführliche Planung der Restaurierung durchgeführt werden.



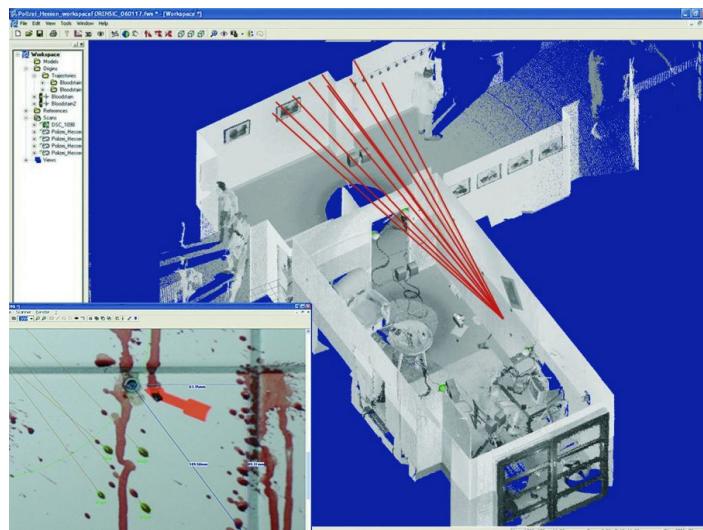
Abbildung 2.4.: Einsatzbereich Denkmalschutz - Das King's Theatre in Glasgow wurde im Zuge einer Sanierung 2011 mittels 3D-Laserscanner vollständig erfasst.[FAR12]

**Vermessung** Seit Menschengedenken besteht die Notwendigkeit, Objekte, Umgebungen, Landschaften etc. zu vermessen. Oft ist dies mit einem sehr hohen Zeitaufwand verbunden und bestimmte Bereiche sind nur schwer zugänglich. Werden Messungen dabei fehlerhaft durchgeführt oder gar vergessen, bedarf es einer erneuten, aufwendigen Durchführung dieser. 3D-Laserscannern ermöglichen es Daten aus sicherer Entfernung einmal zu erfassen und unabhängig davon beliebige Messungen durchzuführen.



**Abbildung 2.5.: Einsatzbereich Vermessung** - Erfassen eines Straßenzuges, beliebige Messungen sind jederzeit möglich ohne die Straße erneut besuchen zu müssen.[[FAR12](#)]

**Forensik** Im Bereich der Forensik und der Unfallforschung werden Laserscanner eingesetzt, um Tat- und Unfallorte zu dokumentieren. Im Vergleich zu herkömmlichen Methoden kann dies mittels 3D-Laserscanner kontaktfrei und präzise erfolgen. Außerdem ist ein späteres Analysieren und Rekonstruieren des Tat- bzw. Unfallhergangs beliebig oft möglich, ohne Daten zu verfälschen.



**Abbildung 2.6.: Einsatzbereich Forensik** - Beispiel einer forensischen Analyse. Anhand der Verteilung der Blutspritzer und der Formung des Einschussloches wird der Schusswinkel rekonstruiert.[[FAR12](#)]

## 2. Grundlagen

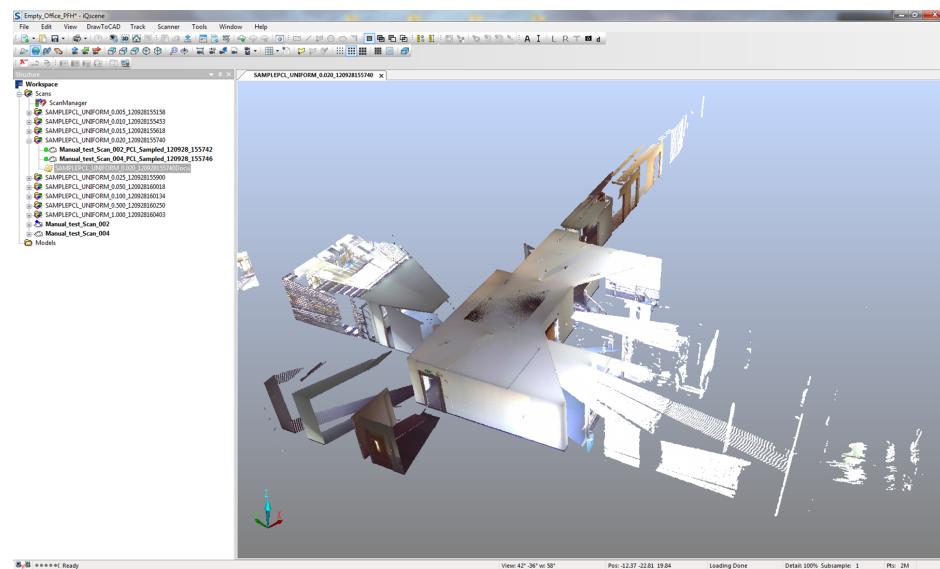
---

### 2.1.4. FARO SCENE

SCENE ist eine für den FARO Focus<sup>3D</sup> konzipierte Software zur Verwaltung und Manipulation von Punktwolken. Mit ihr können ganze Scanprojekte in einem eigenen Repository verwaltet werden und nach Belieben importiert, exportiert und wiederhergestellt werden. Beim Import neuer Punktwolken werden diese zunächst gefiltert (siehe Abschnitt 2.2) und auf Wunsch des Benutzers mit Farbe versehen.

Wurden mehrere Punktwolken der selben Szene aus verschiedenen Blickwinkeln erzeugt, bietet SCENE die Möglichkeit, diese zueinander auszurichten. Dies kann dabei manuell oder automatisiert (siehe Kapitel 3) durch die Extraktion künstlicher Marker und geometrischer Merkmale, erfolgen.

Verschiedenste Arten der Visualisierung, von planaren Ansichten über Tiefen- und Intensitätsbildern bis hin zur 3D-Ansicht, werden unterstützt. Neben der Visualisierung stehen dem Benutzer diverse Werkzeuge zur Verfügung. Diese reichen von Messungen aller Art über das Erkennen geometrischer Objekte, bis hin zur Analyse der akquirierten Daten.



**Abbildung 2.7.: Punktwolken Software SCENE** - Auf der linken Seite befindet sich eine Übersicht der im Projekt enthaltenen Punktwolken. Ebenfalls dort einzusehen sind extrahierte Objekte, Messungen etc. Rechts befindet sich der Bereich in welchem die Visualisierung der Punktwolken stattfindet; hier zu sehen eine 3D-Visualisierung.

## 2.2. Punktwolken

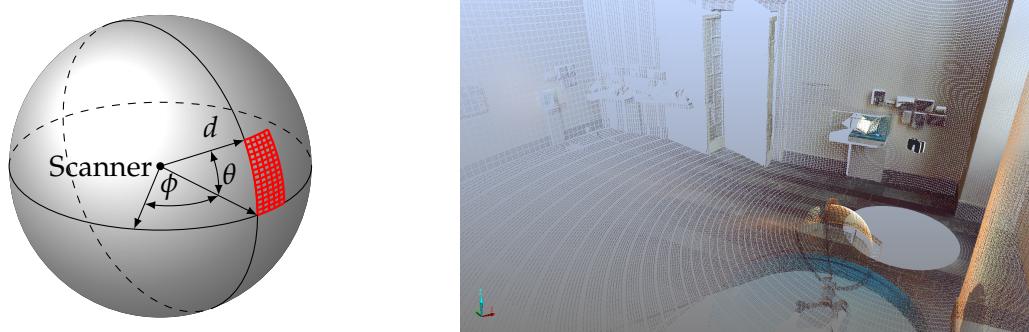
Die in dieser Diplomarbeit verwendeten Punktwolken wurden mit dem FARO® Focus<sup>3D</sup> erzeugt (siehe Abschnitt 2.1.1) und anschließend mit der Software SCENE (siehe Abschnitt

2.1.4) verarbeitet. In diesem Abschnitt wird daher auf die Struktur und die Eigenschaften einer solchen Punktfolge eingegangen. Außerdem werden die darauf angewendeten Filter zum Entfernen von Störungen (Rauschen) erläutert.

### 2.2.1. Oberflächen-Punktwellen

Die vom FARO® Focus<sup>3D</sup> erzeugten Punktwellen sind Oberflächen-Punktwellen. Oft werden diese auch  $2\frac{1}{2}$ D-Punktwellen oder sphärische Punktwellen genannt und zeichnen sich dadurch aus, dass die Koordinaten zweier Dimensionen durch eine Rasterebene festgelegt sind. Jeder Rasterzelle wird ein Wert zugeordnet, in diesem Fall eine gemessene Distanz. Auf diese Art entstandene Punktwellen beschreiben dadurch die Oberflächen der in einer Szene vorkommenden Objekte.

Die Zuordnung einer Distanz zu einer Rasterzelle erfolgt über die Winkelstellungen von Spiegel und Laserscanner. Dies ist schematisch in Abbildung 2.8 zu sehen. Aus den Polarkoordinaten  $(d, \theta, \phi)$  werden die kartesischen Koordinaten eines Punktes  $p_i$  ermittelt. Dabei entspricht  $d$  der gemessenen Distanz,  $\theta$  dem horizontalen und  $\phi$  dem vertikalen Winkel. Über diese können auch die mittels Digitalkamera ermittelten Farbinformationen den entsprechenden Punkten zugeordnet werden. Diese Methodik führt dazu, dass die Dichte einer Punktfolge mit der Distanz zum Laserscanner abnimmt (siehe Abbildung 2.8). Der Bereich unter dem Laserscanner kann nicht eingesehen werden, da er durch diesen selbst verdeckt wird.



**Abbildung 2.8.: Oberflächen-Punktfolge** - Links: Der Winkel  $\phi$  entspricht der Winkelstellung des Laserscanner,  $\theta$  des Spiegels. Die Rasterebene ist durch diese beiden Winkel definiert. Die Distanz  $d$  wird vom Laser gemessen. Rechts: Die Dichte einer sphärischen Punktfolge nimmt mit der Entfernung zum Laserscanner ab. Der Bereich unter dem Laserscanner kann nicht eingesehen werden da er durch selbigen verdeckt wird.

### 2.2.2. Filter

Das Akquirieren von Daten ist stets fehlerbehaftet. In der Bildverarbeitung wird dies allgemein als Rauschen bezeichnet und äußert sich in fehlerhaften Koordinaten einzelner

3D-Punkte. Da sich Rauschen stets negativ auf die Verarbeitung der Daten auswirkt, müssen Punktfolgen gefiltert werden.

Die Filterung der hier verwendeten Punktfolgen wird von SCENE (siehe Abschnitt 2.1.4) bewerkstelligt. Dazu werden die im Folgenden Filter angewendet:

- **Dark-Point-Filter** - Die Distanz zu einem gemessenen Punkt wird mittels Phasenverschiebung und Intensität des reflektierten Laserstrahles errechnet (siehe Abschnitt 2.1.1). Ist die Intensität des reflektierten Strahles zu schwach (trifft z.B. auf sehr dunkle, stark absorbierende Objekte), kann keine Zusicherung über die Korrektheit der gemessenen Distanz gegeben werden. In diesem Fall wird der Punkt aus der Punktfolge entfernt.
- **Stray-Point-Filter** - Dieser Filter versucht Rauschen zu erkennen und zu eliminieren. Dazu wird die Umgebung eines jeden Punktes betrachtet. Weichen Distanzwerte einzelner Punkte auf zu geringem Raum voneinander ab, werden diese aus der Punktfolge entfernt. Verursacht werden kann dies z.B. durch Schmutzpartikel in der Luft, welche den Strahl reflektieren bevor dieser das eigentliche Objekt erreicht.

### 2.3. Datenstrukturen

Bei der Verarbeitung großer Punktmenge spielt das Auffinden und der Zugriff auf ein Datum eine entscheidende Rolle. Wie und mit welcher Effizienz diese Operationen ausgeführt werden sind maßgeblich von der Datenstruktur abhängig, in welcher diese gespeichert werden. In diesem Abschnitt soll daher auf einige der grundlegendsten Datenstrukturen im Bereich der 3D-Datenverarbeitung eingegangen werden. Als Programmiersprache wird C++ zugrunde gelegt. Das Augenmerk liegt dabei auf die für diese Diplomarbeit wichtigen Operationen. Diese sind der Zugriff auf ein bestimmtes Element, sowie die Suche nach dem nächsten Nachbarn eines Punktes. Außerdem soll ein Einblick in den Speicherbedarf gegeben werden. Betrachtet werden dabei Punktfolgen mit 3D-Punkten, gegeben durch X-, Y-, und Z-Koordinate.

#### 2.3.1. 1D-Array

Das 1D-Array ist eine der einfachsten Datenstrukturen zum Speichern großer Punktmenge. Da es linear im Speicher liegt, d.h. kein Verlinkungen zwischen den einzelnen Elementen notwendig sind, ist sie sehr speicherplatzeffizient. Für eine Punktfolge mit  $n$  Elementen werden  $n$  Elemente der Größe eines 3D-Punktes benötigt. Erzeugen lässt es sich in  $\mathcal{O}(n)$ .

Zugriffe auf einzelne Elemente erfolgen mittels Index (Speicheradresse), in konstanter Zeit, also  $\mathcal{O}(1)$ . Da 3-D Punkte in einem Array nicht nach ihrer vollständigen räumlichen Lage sortiert vorliegen können, muss bei der Suche nach einem Element im Worst-Case das gesamte Array durchlaufen werden. Dies entspricht einer Komplexität von  $\mathcal{O}(n)$ . Die Suche

nach dem nächsten Nachbarn eines 3D-Punktes ist äquivalent zur Suche eines bestimmten Elementes und liegt ebenfalls in  $\mathcal{O}(n)$ .

### 2.3.2. 2D-Array

Diese Datenstruktur entspricht einer speziell auf  $2^{\frac{1}{2}}$ D-Punktwolken zugeschnittene Speicherstruktur und wird auch als Range-Image bezeichnet. Sie setzt voraus, dass jeder Punkt eindeutig ist, was durch das Erzeugungsprinzip (siehe Abschnitt 2.2.1) einer  $2^{\frac{1}{2}}$ D-Punktwolke gegeben ist. In ihr können 3D-Punkte nach zwei Dimensionen (hier mit X und Y bezeichnet) sortiert abgelegt werden.

Realisiert wird dies in Form eines 2D-Array. Jede vorkommende X-Koordinate erhält eine eigene Zeile, jede vorkommende Y-Koordinate eine eigene Spalte. Die zugehörige Z-Koordinate wird dann an entsprechender Stelle gespeichert. Für  $n$  3D-Punkte werden  $n$  Elemente der Größe einer Z-Koordinate benötigt. Da die Elemente wie in einem 1D-Array linear im Speicher liegen, ist der Speicheraufwand noch geringer und lässt sich ebenfalls in  $\mathcal{O}(n)$  erzeugen. Zugriffe auf Elemente erfolgen per Index, d.h. in konstanter Zeit ( $\mathcal{O}(1)$ ).

Durch die Eindeutigkeit eines 3D-Punktes kann selbst die Suche nach einem bestimmten Element in konstanter Zeit durchgeführt werden. Die Nachbarschaftssuche erfordert ebenfalls konstanten Zeitaufwand. Für den Zugriff auf einen Nachbar muss lediglich der Index der in Suchrichtung liegenden Koordinate erhöht werden.

### 2.3.3. K-dimensionaler-Baum (KD-Tree)

Ein  $k$ -dimensionaler Baum ist ein binärer Suchbaum zum Speichern von Punkten des  $\mathbb{R}^k$ . Dadurch ist es möglich verschiedenste vektorisierbare Elemente zu verwalten [BL97]. Für 3D-Punkte wird demnach ein 3-dimensionaler Baum benötigt. Die Punkte liegen dabei in alle Dimensionen sortiert vor.

Jede Ebene im Baum teilt den Punktraum durch eine Hyperebene entlang einer Raumachse. Sei die Baumwurzel als Ebene 0 definiert, dann teilen die Baumebenen  $\{p \mid p \bmod 3 = 0\}$  den Punktraum in Richtung der X-Achse,  $\{p \mid p \bmod 3 = 1\}$  und  $\{p \mid p \bmod 3 = 2\}$  in Richtung der Y- und Z-Achse.

Die 3D-Punkte werden in den Knoten des Baumes gespeichert. Der Speicheraufwand liegt dadurch in  $\mathcal{O}(n + (n \cdot 2t))$  wobei  $t$  den Speicherplatz eines Zeigers auf einen Kind-Knoten definiert. Der Zugriff auf ein bestimmtes Element entspricht der Suche nach diesem und erfolgt in  $\mathcal{O}(\log(n))$ .

Der Aufbau des  $k$ -dimensionalen Baumes erfordert eine Suche nach der richtigen Position eines jeden Elementes. Dies entspricht einer Komplexität von  $\mathcal{O}(n \log(n))$ . Sei für das Element  $l$  der nächste Nachbar gesucht, dann wird dessen Teilbaum rekursiv bis zum Blatt durchlaufen und als aktueller nächster Nachbar markiert. Beim rekursiven Rücklauf werden sämtliche an den Abstiegspfad angrenzenden Unterbäume im ersten Knoten besucht. Ist

## 2. Grundlagen

---

deren Distanz zu  $l$  geringer als die des aktuellen nächsten Nachbarn, wird dieser aktualisiert. Dies erfordert einen Zeitaufwand von  $\mathcal{O}(\log(n))$ .

Die in dieser Diplomarbeit verwendeten  $k$ -dimensionalen Bäume entstammen der Point Cloud Library [[Lib12i](#)]. Dabei handelt es sich um eine Variante die im Allgemeinen einen balancierten Suchbaum, realisiert durch eine adaptive Wahl der zu bildenden Hyperebene, erzeugt. Operationen auf diesen können dann im Average-Case erfolgen, daher beziehen sich die hier genannten Komplexitätsklassen ebenfalls auf diesen.

Weiterführende Informationen zu KD-Tree's können unter [[NLHo7](#)] und [[BL97](#)] gefunden werden.

### 3. Registrierung von Punktwolken

Um eine vollständige, dreidimensionale Repräsentation einer Szene zu erhalten ist es notwendig, Punktwolken aus verschiedenen Blickwinkeln zu erzeugen. Durch Verdeckung fehlende Bereiche werden dadurch ebenfalls erfasst. Die so erzeugten Punktwolken müssen anschließend in ein gemeinsames Koordinatensystem überführt und zueinander ausgerichtet werden. Dies ist notwendig, um die aufgenommene Szene maßstabsgetreu und der Realität entsprechend repräsentieren zu können.

Beim Überführen der Punktwolken in ein globales Koordinatensystem werden die erfassten Punktkoordinaten aller Punktwolken so transformiert, dass sie demselben Maß entsprechen. Dies bedeutet, dass Skalierungen und Verzerrungen in den Punktkoordinaten angeglichen werden müssen. Da die hier verwendeten Punktwolken bereits demselben Maßstab und Prinzip entsprechend erzeugt wurden (siehe Abschnitt 2.1.1), ist dies jedoch nicht notwendig. Das Überführen in ein gemeinsames Koordinatensystem entspricht dadurch dem Übereinanderlegen des Ursprungs (Scannerposition) beider Punktwolken (siehe Abbildung 3.1).

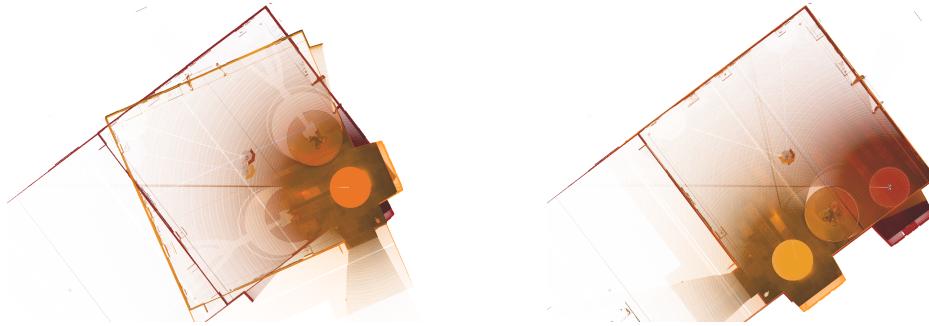
Wenn ein Punkt einer Szene aus unterschiedlichen Positionen aufgenommen wird, tritt dieser in verschiedenen Punktwolken auf. Sind die Punktwolken nicht zueinander ausgerichtet, besitzt er in jeder Punktwolke eine andere Koordinate. Diese Punkte werden als korrespondierend bzw. als Korrespondenzen bezeichnet. Das Auffinden von Korrespondenzen sowie das Verschieben (Translation) und Verdrehen (Rotation) der Punktwolken, sodass korrespondierende Punkte jeder Punktwolke dieselbe Koordinate besitzen, wird als Registrierung bezeichnet. (siehe Abbildung 3.1).

Grundlegend lässt sich die Registrierung auf zwei Arten durchführen: direkt/manuell und automatisiert. Bei der direkten/manuellen Registrierung wird Position und Orientierung des Laserscanners während dem Scanvorgang in einem globalen Koordinatensystem bestimmt [SB05]. Heutige Technologien und Sensoren erlauben es diese Informationen auch automatisch zu erfassen, weshalb man auch von einer direkten Registrierung spricht. Meist wird dabei eine Kombination verschiedener Sensoren wie GPS, Kompass und Inklinometer benutzt. Der Einsatz eines Trägheitsnavigationssystems wie von Talaya et al. [JRE+04] beschrieben, ist dabei ebenso, denkbar wie die Verwendung odometrischer Daten, die bei der Roboternavigation zum Einsatz kommt [Hoc05, Nü02].

Ungenauigkeiten oder gar ausbleibende Signale der Sensoren (wie z.B. GPS in Gebäuden) führen dabei häufig zu großen Abweichungen in der Registrierung und erfordern oft zusätzliche Elektronik. Daher ist diese Art der Registrierung nur für bestimmte Szenarien geeignet.

### 3. Registrierung von Punktwolken

---



**Abbildung 3.1.: Registrierung von Punktwolken** - *Links:* Die Punktwolken liegen in einem gemeinsamen Koordinatensystem vor; der Ursprung beider Punktwolken liegt übereinander (kleiner Kreis Punkte rechts unten). *Rechts:* Die Punktwolken wurden registriert; der Szene entsprechend gleiche Punkte liegen übereinander.

Im Rahmen dieser Diplomarbeit soll die automatisierte, paarweise Registrierung betrachtet werden. Bei dieser werden, anders als bei der globalen Registrierung, nicht beliebig viele, sondern nur Paare von Punktwolken zueinander ausgerichtet [DH01]. Der Ursprung einer Punktwolke (Laserscanner-Position) wird dabei an dem des gemeinsamen Koordinatensystems fixiert. Ausgerichtet wird die verbleibende Punktwolke. Für sie muss die passende Transformation (Translation und Rotation) berechnet werden. Im  $\mathbb{R}^3$  entspricht dies dem Bestimmen von sechs Freiheitsgraden (Translation entlang der X-, Y- und Z-Achse und Rotation um X-, Y- und Z-Achse).

Automatisierte Ansätze zum Lösen des Registrierungsproblems sind in großer Zahl vorhanden. Nachdem das Registrierungsproblem mathematisch definiert wurde, wird daher eine mögliche Klassifikation vorgestellt. Diese gibt einen allgemeinen Überblick und kategorisiert die im Späteren untersuchten Algorithmen (Iterative Closest Point, Normal Distribution Transformation, Point-Feature-Histogram, Spin-Images).

#### 3.1. Mathematische Definition

Gegeben seien zwei, in globalen Koordinaten vorliegende 3D-Punktwolken  $P$  und  $Q$ . Diese repräsentieren dieselbe Szene aus verschiedenen Blickwinkeln.

$$(3.1) \quad \begin{aligned} P &= \{p_i \mid p_i \in \mathbb{R}^3, \quad 0 < i \leq n, \quad n \in \mathbb{N}\} \\ Q &= \{q_j \mid q_j \in \mathbb{R}^3, \quad 0 < j \leq m, \quad m \in \mathbb{N}\} \end{aligned}$$

Sei außerdem eine beliebige Transformation im  $\mathbb{R}^3$  durch die Rotationsmatrix  $R$  und einen Translationsvektor  $\vec{t}$  wie folgt beschrieben.

$$\begin{aligned}
 (3.2) \quad R &= R_x \cdot R_y \cdot R_z \\
 R_x &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix}, \quad \alpha \in \mathbb{R}, \quad 0 \leq \alpha < 360 \\
 R_y &= \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 1 & \cos(\beta) \end{pmatrix}, \quad \beta \in \mathbb{R}, \quad 0 \leq \beta < 360 \\
 R_z &= \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \gamma \in \mathbb{R}, \quad 0 \leq \gamma < 360 \\
 \vec{t} &= \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}, \quad t_x, t_y, t_z \in \mathbb{R}
 \end{aligned}$$

Des Weiteren sei eine Korrespondenz definiert als ein Paar von Punkten beider Punktwolken, deren Punktkoordinaten, der repräsentierten Szene entsprechend, identisch sein sollten.

$$(3.3) \quad c = (c_p, c_q), \quad c_p \in P, c_q \in Q$$

Das Registrierungsproblem ist dann gelöst, wenn  $R$  und  $\vec{t}$  so bestimmt sind, dass für die Menge aller Korrespondenzen  $C$  gilt:

$$(3.4) \quad C = \{(c_{k_p}, c_{k_q}) | c_{k_p} = R \cdot c_{k_q} + \vec{t}, \quad 1 \leq k \leq |C|\}$$

Auf Grund von Ungenauigkeiten in den Messdaten (z.B. Rauschen) kann diese Gleichheit in der Praxis nahezu nie erreicht werden. Daher muss diese Definition um einen Grenzwert  $\varepsilon \in \mathbb{R}$  erweitert werden.

$$(3.5) \quad C = \{(c_{k_p}, c_{k_q}) | \|c_{k_p} - (R \cdot c_{k_q} + \vec{t})\|_2 \leq \varepsilon, \quad 1 \leq k \leq |C|\}$$

Der Abstand zwischen zwei korrespondierenden Punkten ( $\|c_{k_p} - (R \cdot c_{k_q} + \vec{t})\|_2$ ) wird dabei als Spannung bezeichnet.

## 3.2. Klassifikation

Dieser Abschnitt gibt einen Einblick in die Vielfalt der Lösungsvarianten des Registrierungsproblems. Dabei wird dem Leser auch eine Kategorisierung der im späteren Verlauf untersuchten Algorithmen vorgestellt. Gründe der Wahl sind über die in Abschnitt 1.2

### 3. Registrierung von Punktwolken

---

definierten Kriterien sowie die in der Klassifikation beleuchteten Verfahrensweisen zu erschließen. Eine vollständige Betrachtung aller aktuell existierenden Algorithmen würde den Rahmen diese Arbeit sprengen.

Prinzipiell existieren vielerlei mögliche Klassifikationen automatisierter Registrierungs-Algorithmen. Dabei ist die Trennung durch das Aufgabengebiet ebenso berechtigt, wie eine Klassifizierung basierend auf der mathematischen Definition, der zur Berechnung einer Transformation benutzten Methodik. Die hier verwendete Klassifikation basiert auf der von [SMFFo5] und [CFo1]. Sie teilt automatisierte Registrierungs-Algorithmen grundlegend in zwei Klassen:

- **Grobregistrierung** - Diese Klasse beherbergt Verfahren, die nicht in der Lage sind Vorwissen zu verarbeiten. Dadurch sind sie in der Lage, Punktwolken unabhängig von ihrer initialen Ausrichtung zu registrieren. Das Ergebnis ist dabei meist ungenau; die Punktwolken sind nur grob zueinander ausgerichtet.
- **Feinregistrierung** - In dieser Klasse beheimaten sind Algorithmen, die Vorabinformationen bezüglich der Ausrichtung zweier Punktwolken zueinander verwenden. Dabei wird eine grobe Ausrichtung der Punktwolken als initiale Situation vorausgesetzt und verfeinert.

#### 3.2.1. Grobregistrierung

Ziel der Grobregistrierung ist es, Korrespondenzen zwischen zwei Punktwolken zu finden. Dazu werden besondere Merkmale aus beiden Punktwolken extrahiert und miteinander verglichen. Sehr ähnliche Merkmale werden als Korrespondenzen betrachtet. Die Erkennung ähnlicher Merkmale wird dabei als Korrespondenzsuche bezeichnet und weist im Allgemeinen eine Komplexität von  $\mathcal{O}(n^2)$  auf. Dabei bezeichnet  $n$  die Anzahl gefundener Merkmale in einer Punktwolke. Aus ihnen wird eine Transformation berechnet, die alle korrespondierenden Merkmale bestmöglich zueinander ausrichtet. Dies erfolgt mittels Singulärwertzerlegung der Korrelationsmatrix und Schwerpunktdifferenz der Korrespondenzen (siehe Anhang A).

Welche Art von Merkmal dabei gewählt, auf welche Art und Weise die Transformation berechnet und welche Registrierungsstrategie zur Optimierung des Ergebnisses verwendet wird, unterscheidet die einzelnen Verfahren voneinander. Weiter differenziert wird an dieser Stelle über die Art der verwendeten Merkmale. Dies liegt darin begründet, dass Strategie, Transformation und Optimierung oftmals identisch sind.

#### Künstliche Merkmale

Als Merkmale werden bei dieser Methode spezielle Marker eingesetzt. Diese besitzen eine vorgegebene, algorithmisch leicht auffindbare, geometrische oder optische Struktur (siehe Abbildung 3.2). Vor dem Scavorgang werden diese in der aufzunehmenden Szene so platziert, dass aus jedem Blickwinkel ausreichend viele (mindestens drei) davon im Überlapp

sichtbar sind. Nach dem Scavorgang werden die akquirierten Punktwolken auf Marker hin untersucht. Extrahiert wird Art und Position eines Markers. Korrespondierende Marker können dann durch die in Graphen repräsentierten räumlichen Relationen ermittelt werden [Bogo8].

Werden nicht allzu komplexe Marker verwendet, ist eine Extraktion mit geringem Zeitaufwand möglich. Wurden diese außerdem sinnvoll (keine symmetrische Anordnung) platziert, ist eine äußerst zuverlässige Registrierung der Punktwolken möglich. Nachteile dieser Methodik stellten der hohe Zeitaufwand zur Positionierung der Marker, sowie das Verdecken einzelner Bereiche der Szene durch diese dar.



Abbildung 3.2.: Künstliche Marker - Links: Schachbrettmuster. Rechts: Kugelboard.  
Diese Marker werden als leicht auffindbare Korrespondenzpunkte zur Registrierung von der Firma FARO verwendet [FAR12].

### Geometrische Merkmale

Als geometrische Merkmale werden aus Kanten gebildete Eckpunkte [Bero9, Troo8], Rechtecke [Bero9] und Flächen [DB06] verwendet. Eckpunkte werden dabei meist über zwei, bzw. drei sich schneidende Kanten definiert und sind translationsinvariant. Kanten haben den Vorteil, dass lediglich die Rotation um sich selbst und die Verschiebung entlang ihrer Richtung ungeklärt bleibt. Die Länge einer Kante kann jedoch auf Grund des Blickwinkels oft nicht exakt bestimmt werden. Rechtecke hingegen legen sämtliche Rotationen außer die symmetrischen um deren Normale fest. Allgemeine Flächen können das Problem der Symmetrie lösen; die Extraktion und der Vergleich dieser ist jedoch deutlich zeitaufwendiger.

Zur Extraktion und zum Vergleich dieser Merkmale existieren hoch-optimierte Verfahren. Dadurch kann der Zeitaufwand der Merkmalssuche gering gehalten werden. Nachteil ist, dass nicht jede Szene (speziell natürliche Umgebungen) diese geometrischen Merkmale in ihrer Reinform enthält. Oft können dadurch keine Merkmale gefunden werden. Andererseits können in künstlichen Szenen (Bürogebäude, Straßenzüge etc.) sehr viele davon extrahiert werden. Dies stellt dann eine zeitliche Herausforderung an die Korrespondenzsuche dar.

**Punkt-Deskriptor basierte Verfahren** Punktbasierter Merkmale beschreiben generell die Umgebung eines konkreten Punktes. Daher werden diese Merkmale auch als Deskriptoren

### 3. Registrierung von Punktwolken

---

bezeichnet. Deskriptoren sind allgemeine, durch ihre Berechnungsvorschrift definierte punktbasierte Merkmale. Ziel der Deskriptoren ist es, einzelne Punkte einer Punktfolge über eine abstrakte Repräsentation vergleichbar zu beschreiben. Aus diesem Grund werden von Deskriptoren folgende Eigenschaften gefordert [BOo8]:

- **Transformations-Invariant** - Es sei angenommen, dass ein Deskriptor für einen Punkt  $p$  einer Punktfolge bestimmt und anschließend transformiert wird ( $p \rightarrow p'$ ). Dann muss der Deskriptor von  $p'$  gleich bzw. ähnlich dem Deskriptor von  $p$  sein.
- **Vergleichbar** - Zwei auf unterschiedlichen Punktfolgen berechnete Deskriptoren müssen über ein Maß vergleichbar sein, welches deren Ähnlichkeit beschreibt.
- **Effizient** - Die Berechnung eines Deskriptors sollte in angemessener Zeit erfolgen. Dabei sollen nur für den späteren Vergleich notwendige Informationen berechnet werden.
- **Rausch-Unempfindlich** Angemessen verrauschte Daten verändern die vom Deskriptor berechneten Informationen nur so stark, dass identische Punkte vom Ähnlichkeitsmaß auch als solche erkannt werden.

Verschiedene Verfahren unterscheiden sich grundlegend in der Definition ihres Deskriptors. Im Folgenden werden einige der bekanntesten vorgestellt.

Ein sehr einfacher Deskriptor ist der von Oztireli und Basdogan [BOo8] vorgestellte. Dieser beschreibt die Distanz eines Punktes zum Schwerpunkt seiner benachbarten Punkte. Er lässt sich sehr effizient berechnen und ist wenig anfällig für Rauschen. Die Aussagekraft ist jedoch sehr eingeschränkt, da bereits benachbarte Punkte nahezu identische Deskriptoren erzeugen.

Chua [CJ97] führte im Jahr 1997 erstmals die Punktsignatur ein. Für einen Punkt  $p$  wird dabei der Verlauf der Oberfläche  $C$ , auf welcher sich  $p$  befindet, als Deskriptor verwendet. Dabei werden Schnittpunkte einer Kugelumgebung von  $p$  mit der Oberfläche  $C$  ermittelt. Anschließend werden diese auf die Tangentialebene von  $p$  und  $C$  projiziert. Über den Normalenvektor von  $p$  und den von  $p$  entferntesten Punktvektor  $n_1$  wird innerhalb der Kugelumgebung mittels Kreuzprodukt ein Orthonormalsystem definiert. In diesem kann nun jeder sich in  $C$  befindliche Punkt als vorzeichenbehaftete Distanz und Rotation um  $n_1$  repräsentiert werden. Der resultierende Deskriptor entspricht einer 2D-Funktion und wird über den Vergleich von Stützstellen mit anderen auf Ähnlichkeit geprüft. Der Deskriptor reagiert sensibel auf Rauschen und die Berechnung der Schnitte ist zeitintensiv [SMFF05].

Eine Form der statistischen Punkt-Signatur wurde in [WHHo3] vorgestellt. Sogenannte Surflet-Pair-Relation Histogramme beschreiben die Oberflächenbeziehung zwischen zwei benachbarten Punkten über ein Referenzkoordinatensystem.

Dieses Prinzip wurde zu den sogenannten Point Feature Histograms (PFH) [JAo9, RMBBo8, RBMBBo8, Ruso9] ausgebaut. Eine Abwandlung der Point Feature Histograms sind die Fast-Point-Feature-Histograms (FPFH) [RBB09]. Sie gestalten den PFH-Deskriptor weniger rechenaufwendig. Der PFH-Ansatz ist einer der heute am stärksten vertretenen Algorithmen

im Bereich der Objekterkennung und Registrierung ganzer Szenen. Auf Grund seiner scheinbar guten Ergebnisse ist er einer der später untersuchten Algorithmen (siehe 7).

Gegenstand eines etwas anderen Ansatzes zur Berechnung eines Deskriptors, welcher auch als Punktsignatur betrachtet werden kann, sind die sogenannten Spin-Images [FSMo8, JH99, Joh97, CHH99, HH99]. Ursprünglich wurden sie zur Objekterkennung entwickelt. Die dabei erzielten Resultate und die Aussagekraft des Deskriptors sind Anlass, diese beim Einsatz einer Szenen-Registrierung zu untersuchen (siehe 8).

Bei Principal Curvatures [REF+94] werden Punkte über die Oberfläche auf der sie sich befinden beschrieben. Dazu wird maximale und minimale Krümmung sowie die Normale der Oberfläche als Punktsignatur benutzt. Das Berechnen der Krümmung ist dabei sehr aufwendig und anfällig für Rauschen [SMFFo5].

Ein weiterer Ansatz [CYL98] basiert auf der Principal Component Analysis. Mit ihrer Hilfe werden die Orientierungen der Hauptachsen zweier Punktwolken bestimmt und zueinander ausgerichtet. Die Translation wird über den Schwerpunkt der Punktwolken ermittelt. Untersuchungen von [SMFFo5] haben gezeigt, dass dieses Verfahren sehr schnell ist. Für eine erfolgreiche Registrierung müssen sich die Punktwolken jedoch mehr als die Hälfte überlappen. Da diese Annahme in den hier untersuchten Szenarien nicht getroffen werden kann, scheidet dieses Verfahren zur weiteren Untersuchung aus.

Der letzte hier behandelte Deskriptor ist ein, auf den Extended-Gaussian-Images (EGI), basierender. Grundlegendes Prinzip ist dabei, die Normalen aller Punkte einer Punktwolke auf eine Gauß-Kugel aufzutragen. Gleiche Normalen werden an derselben Stelle aufgetragen. Sind viele Normalen sehr ähnlich, erhält die Gauß-Kugel an diesen Stellen besondere Ausprägungen und wird als EGI bezeichnet. EGI's mit ähnlichen Ausprägungen beschreiben dieselben Punktmenge. Rotiert werden müssen die Punktwolken so, dass die Ausprägungen übereinander liegen.

In [Dolo5] werden die Normalen von Punkten und segmentierten Flächen verwendet. Bei [MPIDo6] wird ein EGI für eine gesamte Punktwolke erzeugt. Diese Verfahren ist für den Einsatz in einem globalen Registrierungsvorgang gedacht. Dabei werden die EGI's mit Hilfe von Kugelflächenfunktionen (Sphärische Harmonische) repräsentiert. Mittels Korrelationskoeffizient können diese dann mit geringem Zeitaufwand verglichen werden. Die Translation wird ebenfalls über eine Korrelationsfunktion definiert welche den Überlappungsbereich der bereits rotierten Punktwolken bewertet. Das Verfahren arbeitet sehr schnell und erzielt gute Resultate, erfordert jedoch einen Überlappungsbereich von mindestens 45%.

## Weitere Verfahren

Neben der Verwendung von Deskriptoren existieren noch weitere Verfahren, die nicht näher klassifiziert werden. Hierzu sind im Folgenden einige Repräsentative vorgestellt.

In [TCC98] wird ein Ansatz vorgestellt, welcher mit Hilfe algebraischer Methoden den Oberflächenverlauf einer Punktwolke als Polynome repräsentiert. Diese Repräsentation ist

### 3. Registrierung von Punktwolken

---

wenig anfällig für Rauschen und bietet eine effiziente Repräsentation zur Korrespondenzsuche. Nachteil davon ist jedoch, dass ein Überlappungsbereich von 85% [SMFF05] gefordert wird.

Auch genetische Algorithmen haben längst Einzug gehalten; als Individuum wird in [BS96] dabei ein Vektor von Korrespondenzen definiert. Mit Hilfe der Punktkoordinaten und Orientierungen eines Punktes wird eine Fitnessfunktion aufgestellt. Diese bewertet Winkelabweichungen und Distanzen.

In [LCE06] wird als Individuum ein sechsdimensionaler Vektor als Individuum verwendet. Dieser beschreibt die Rotation und Translation um die Koordinatenachsen. Die Fitnessfunktion basiert auf den Least Trimmed Squares [CSK05]. Generell leiden genetische Algorithmen unter einer gewissen Willkür. So sind im Allgemeinen Aussagen über die Konvergenz und Laufzeit schwer möglich.

Ein anderer, auf RANSAC (Random Sample Consensus) basierender, Ansatz wurde von Chen [CHC] vorgestellt. Dabei werden alle Punktkombinationen, jeweils drei benachbarter Punkte, der Ziel-Punktfolge betrachtet. Beschrieben werden sie über ihre Distanzrelationen zueinander. In der Daten-Punktfolge werden drei Punkte gesucht, welche dieselben Distanzrelationen aufweisen. Diese sechs Punkte definieren dann drei Korrespondenzen, die zur Berechnung einer eindeutigen Transformation ausreichend sind. Wiederholt wird dies für alle Punktkombinationen und resultierenden Korrespondenz Triplets. Als finale Transformation wird aus der Menge aller Transformationen diejenige gewählt, die am häufigsten berechnet wurde. Das Ergebnis der Registrierung ist beeindruckend, ebenso der exorbitante Rechenaufwand bei großen Punktfolgen. Dieser kann durch eine zufällige Wahl von Triplets minimiert werden, führt dann allerdings zu einer weniger stabilen Variante.

#### 3.2.2. Feinregistrierung

Die Feinregistrierung versucht eine bereits grobe Ausrichtung zweier Punktfolgen zu optimieren. Im Allgemeinen geschieht dies in einem iterativen Prozess. Dabei wird in jedem Schritt angestrebt eine Fehlerfunktion zu minimieren. Das Registrierungsproblem kann dann als Optimierungsproblem betrachtet werden. Trotzdem bedienen sich einige der hier angesiedelten Algorithmen ebenfalls der Suche nach Korrespondenzen. Diese werden zur Definition der Fehlerfunktion verwendet.

Eine Unterscheidung der Verfahren ist oft über deren Fehlerfunktion, Definition als Optimierungs- oder Korrespondenzproblem sowie der Methodik zur Bestimmung einer Transformation realisiert. Hier soll die zugrundeliegenden Idee im Vordergrund stehen, da eine Veränderung der vorher genannten Unterscheidungsmöglichkeiten meist nur zu einer Variante und keinem neuen Verfahren führt.

Der wohl bekannteste hier vertretene Algorithmus ist der ICP (Iterative Closest Point), wie er von Besl und McKay [BM92] eingeführt wurde. Dieser versucht den Abstand zwischen Korrespondenzen zu minimieren. Von ihm existieren unzählige Varianten, die auf Punktfolgen geringer Dichte bzw. Punktmenge sehr gut untersucht wurden. Die dabei erzielten Ergebnisse waren ausschlaggebendes Argument, diesen Algorithmus in die Untersuchungen

der vorliegenden Arbeit aufzunehmen. Eine genauere Beschreibung wird daher in Abschnitt 9 vorgenommen.

Ebenfalls den Abstand zwischen Korrespondenzen minimierend, arbeitet der von Chow et al [CTL04] entworfene genetische Algorithmus. Dabei wird als Individuum ein sechsdimensionalen Vektor definiert, dessen Einträge die drei Rotationswinkel und Verschiebungen in Richtung der Koordinatenachsen repräsentieren. Mittels Mutation und Crossover wird dann der Transformationsraum nach der besten Lösung durchsucht. Als Fehlerfunktion wird der Median der Abstände zwischen Punkt-zu-Punkt-Korrespondenzen verwendet. Da lokale Minima der Fehlerfunktion mit Hilfe genetischer Algorithmen umgangen werden können, kann über die Konvergenzgeschwindigkeit nur wenig ausgesagt werden.

Der letzte hier angesprochene Algorithmus benutzt die durch [BS03] eingeführte Normalverteilung von Punkten und wird daher auch NDT (Normal Distribution Transformation) genannt. Dieses Konzept wurde von Magnusson in [Mag09, MLD07] für 3D-Punktwolken erweitert. Die dabei erzielten Ergebnisse sind sehr gut und fordern, dass auch dieser Algorithmus im weiteren Verlauf genauer zu untersuchen <sup>10</sup> auf Seite 73.



## 4. Fehlermaße

Um das Ergebnis einer Registrierung evaluieren zu können sind Fehlermaße notwendig. Diese müssen unabhängig vom verwendeten Algorithmus vergleichbar sein. Sinnvoll erscheint es dabei, den Fehler über den Unterschied zwischen der korrekten und berechneten Transformation zu ermitteln.

Mit Hilfe homogener Transformationsmatrizen wird im ersten Teil dieses Kapitels ein dafür geeignetes Fehlermaß vorgestellt. Dieses erfasst den verursachten Gesamtfehler einer Registrierung. Eine gesonderte Betrachtung des durch die Rotation bzw. Translation verursachten Fehlers ist dabei nicht möglich. Sie gibt jedoch Aufschluss über die Stärken und Schwächen eines Algorithmus. Daher werden zwei weitere Fehlermaße eingeführt, die eine Aussage bezüglich des Rotations- und Translationsfehlers ermöglichen.

Die korrekte Transformation sei dabei durch die Rotationsmatrix  $R_c$  und den Translationsvektor  $\vec{t}_c$  gegeben, die berechnete analog dazu, durch  $R_e$  und  $\vec{t}_e$ . Definiert sind diese wie in Abschnitt 3.1.

### 4.1. Gesamtfehler

Eine gemeinsame Darstellung von Rotation und Translation ist durch homogene Transformationsmatrizen ( $\mathbb{R}^{4 \times 4}$ ) gegeben. Diese Darstellung ermöglicht es, den Unterschied zweier Transformationen über Matrizen-Normen zu beschreiben.

Verwendet wird die Frobeniusnorm. Sie ist auf dem Körper der reellen Zahlen über eine Vektornorm definiert. Dabei wird eine  $m \times n$  Matrix ( $m, n \in \mathbb{N}$ ) als  $m + n$  dimensionaler Vektor aufgefasst. Entsprechend der euklidischen Vektornorm lässt sich die Frobeniusnorm für eine Matrix  $A \in \mathbb{R}^{m \times n}$  definieren als:

$$(4.1) \quad \|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}, \quad a_{ij} \in \mathbb{R}$$

Sei die korrekte  $(R_c, \vec{t}_c)$  und die berechnete Transformation  $(R_e, \vec{t}_e)$  in homogenen Transformationsmatrizen ohne Skalierung und Verzerrung durch  $H_c^{4 \times 4}$  und  $H_e^{4 \times 4}$  gegeben, dann ist der Gesamtfehler  $E_{Gesamt}$  der Registrierung über deren Differenzmatrix  $H_d = H_c - H_e$  definiert als:

$$(4.2) \quad e_{Gesamt} = \|H_d\|_F = \sqrt{\sum_{i=1}^4 \sum_{j=1}^4 (h_{d_{ij}})^2}, \quad i, j \in \mathbb{N}$$

Dieses Maß beschreibt den Unterschied zwischen einer berechneten und einer korrekten Transformation. Dadurch steht es in direktem Zusammenhang zur Güte eines Registrierungsergebnisses. Translation und Rotation werden dabei gemeinsam betrachtet. Unterschiedliche Abweichungen in Translation und Rotation können dadurch zu einem identischen Fehler führen.

## 4.2. Translationsfehler

Um das Ergebnis eines Registrierungs-Algorithmus richtig zu deuten muss festgestellt werden können, weshalb sich ein Punkt einer Punktwolke nicht an seiner korrekten Position befindet. Eine Ursache kann dabei eine fehlerhafte Translation sein.

Der Einfluss der Translation  $\vec{t} \in \mathbb{R}^3$  bei der Transformation eines Punktes  $p \in \mathbb{R}^3$  ist durch  $f_t(p) = p + \vec{t}$  gegeben. Wie weit ein Punkt  $p$  dabei durch die berechnete Translation fälschlicherweise verschoben wurde, kann durch den Differenzvektor  $\vec{e}_{Translation}$  der resultierenden Punktpositionen zueinander wie folgt gemessen werden:

$$(4.3) \quad \begin{aligned} \vec{e}_{Translation} &= f_{t_c} - f_{t_e} \\ &= p + t_c - p + t_e \\ &= t_c - t_e \end{aligned}$$

Der Translationsfehler ist dabei unabhängig von der Punktposition eines Punktes  $p$ . Das bedeutet, dass der Translationsfehler auf alle Punkte in selbem Maße wirkt und repräsentativ für den verursachten Fehler innerhalb einer gesamten Punktwolke steht. Als vergleichbares Maß kann dann die euklidische Norm von  $\vec{e}_{Translation}$  gebildet werden:

$$(4.4) \quad e_{Translation} = \|\vec{e}_{Translation}\|_2$$

## 4.3. Rotationsfehler

Neben dem Translationsfehler hat auch die fehlerhafte Rotation einer Punktwolke Einfluss auf die Abweichung der darin enthaltenen Punkte zu deren korrekten Position.

Analog zum Translationsfehler bestünde ein Ansatz um diesen Fehler zu erfassen darin, den Abstand  $\vec{e}_{Rotation}$  zwischen dem korrekt und fehlerhaft rotierten Punkt zu verwenden. Sei die Rotation  $R \in \mathbb{R}^3$  des Punktes  $p \in \mathbb{R}^3$  einer Punktwolke gegeben durch  $f_R(p) = R \cdot p$ , dann lässt sich dieser beschreiben als:

$$(4.5) \quad \begin{aligned} \vec{e} &= f_{R_c}(p) - f_{R_e}(p) \\ &= R_c \cdot p - R_e \cdot p \\ &= (R_c - R_e) \cdot p \end{aligned}$$

Problem dabei ist, dass der Einfluss einer Rotation abhängig von der Lage eines Punktes ist. Dadurch ist keine repräsentative Aussage für eine gesamte Punktwolke möglich. Ein Fehler mit mehr Aussagekraft wird durch die LUB-Norm (lowest-upper-bound) beschrieben:

$$(4.6) \quad \|A\|_L = \max_{x \leq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|, \quad x \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}$$

Der Fehler unter der Matrix-Abbildung  $A$  lässt sich dabei abschätzen als:

$$(4.7) \quad \|Ax\| \leq \|A\|_L \cdot \|x\|$$

Mit zunehmender Norm von  $x$  wird dieser also größer. Berechnen lässt er sich mittels Singulärwertzerlegung der Fehlermatrix  $R_d = R_c - R_e$ :

$$(4.8) \quad \begin{aligned} R_d &= UDV^T \text{ mit} \\ UU^T &= \mathbf{1} \in \mathbb{R}^{3 \times 3} \\ V^TV &= \mathbf{1} \in \mathbb{R}^{3 \times 3} \\ D &= \text{diag}(\sigma_1, \sigma_2, \sigma_3) \in \mathbb{R}^{3 \times 3} \end{aligned}$$

Dabei ist das Bild der Einheitskugel  $\mathbb{S}^2$  unter der Fehlermatrix  $R_d$  gegeben durch:

$$(4.9) \quad R_d \cdot \mathbb{S}^2 \stackrel{(4.9)}{=} UDV^T \mathbb{S}^2 \stackrel{(V^T \mathbb{S}^2 = \mathbb{S}^2)}{=} UDS^2$$

Dies entspricht einem Ellipsoid mit Halbachsen  $\sigma_1, \sigma_2, \sigma_3$  und den Spaltenvektoren von  $U$  als Halbachsenvektoren. Hiermit lässt sich die LUB-Norm berechnen als:

$$(4.10) \quad e_{\text{Rotation}} = \|R_d\|_L = \max(\sigma_1, \sigma_2, \sigma_3)$$

Dies bedeutet, dass ein Punkt  $p$  der fälschlicherweise mittels  $f_{R_e}(p)$  statt  $f_{R_c}(p)$  rotiert wird, pro Abstand 1 zum Ursprung einen maximalen Fehler von  $e_{\text{Rotation}}$  erfährt. Eine genauere Aufschlüsselung des Fehlers in Richtung der Koordinatenachsen der Punktwolke ist durch eine Projektion des Ellipsoides auf diese möglich. Der daraus resultierende Vektor ist gegeben durch:

$$(4.11) \quad \vec{e}_{\text{Rotation}} = (e_x, e_y, e_z) \in \mathbb{R}^3$$



## 5. Testumgebung

DIESES Kapitel beschreibt den Rahmen, in dem die Analysen der einzelnen Algorithmen durchgeführt wurden. Dadurch wird die Möglichkeit gegeben Testergebnisse zu reproduzieren und nachzuvollziehen.

Zur Analyse der Algorithmen wurde eine in die Software SCENE (siehe Abschnitt 2.1.4) integrierte Testumgebung implementiert. Dies ermöglicht, die mit dem Laserscanner Focus<sup>3D</sup> (siehe Abschnitt 2.1) erhobenen Testszenarien (Punktwolken, Farbbilder und Sensorinformationen) zu verwalten und gemäß Abschnitt 2.2.2 zu filtern.

Die Testumgebung wurde in C++ implementiert. Alle ausgeführten Tests und die damit verbundenen Laufzeitanalysen beziehen sich auf ein mit Microsoft® Windows 7 - 64 Bit betriebenes System. Die darin enthaltenen Komponenten sind ein Intel® Core i7 Prozessor mit jeweils 2.2 GHz pro Kern (4 Kerne insgesamt) sowie 8 GB DDR3 Arbeitsspeicher mit einer Taktfrequenz von 1330 MHz.

Nachdem der Funktionsumfang der Testumgebung erläutert und auf die Implementierung bzw. interne Funktionsweise eingegangen wurde, werden die zur Analyse herangezogenen Testszenarien vorgestellt.

### 5.1. Funktionsumfang

Die Testumgebung wird über eine Liste von Konfigurationsdateien gesteuert, die den Funktionsumfang widerspiegeln. Jede Konfigurationsdatei besteht dabei aus zwei unterschiedlichen Bereichen und entspricht einem Testverfahren.

- **Verwaltung** - Hier werden die im Wesentlichen zur Verwaltung notwendigen Informationen definiert. Dazu gehören Ausgabedateien wie Ereignisprotokolle, Analysedateien der Registrierungsergebnisse und Informationen über einzelne Repräsentationen einer Punktwolke (z.B. Deskriptoren (siehe Abschnitt 3.2.1)). An dieser Stelle wird auch die Sub-Sampling-Rate des Uniformen-Sub-Sampling festgelegt. (siehe Abschnitt 6.1). Außerdem kann definiert werden, ob Algorithmen sukzessive aufeinander angewendet werden sollen oder nicht.
- **Algorithmen** - Dieser Teil besteht aus einer Liste der zu prozessierenden Algorithmen. Dabei ist für jeden Algorithmus ein Parametersatz zu definieren. Dies kann auf zwei Arten erfolgen.

## 5. Testumgebung

---

- **Normal** - Es wird ein einfaches Parameterset definiert, mit welchem der zugehörige Algorithmus prozessiert wird.
- **Automatisch** - Es wird ein Parameterset definiert, bei welchem für jeden Parameter ein Start- und Endwert sowie eine Schrittweite festgelegt wird. Die Registrierung wird dann mit allen Kombinationen möglicher Parameter einzeln ausgeführt.

Hier wird auch das räumliche Sub-Sampling definiert, welches mehr Funktionen als nur das Reduzieren von Daten zur Aufgabe haben kann (siehe Abschnitt 7).

Ist ein Testverfahren abgeschlossen werden die erzeugten bzw. transformierten Punktwolken in einer von SCENE gegebenen Projektstruktur separat gespeichert. Sie ermöglicht auch die in Dateien exportierte Registrierungsergebnisse zu verlinken. Ermöglicht wird dadurch eine spätere Betrachtung aller notwendigen Daten.

### 5.2. Funktionsweise

Die Funktionsweise der Testumgebung lässt sich in zwei grundlegende Arbeitsschritte gliedern. Diese werden für jeden Testlauf und somit für jede Konfigurationsdatei durchgeführt.

- **Initialisierung** - In diesem Schritt wird die übermittelte Konfigurationsdatei eingelesen und falls erforderlich automatische Parametersets für Parameteranalysen generiert. Anschließend werden die Punktwolken in bereits gefilterter Form geladen. Die Ursprünge der Punktwolken werden dabei übereinander gelegt; es ist keine Information bezüglich der Transformationen vorhanden. Eine Liste der zu prozessierenden Algorithmen wird angelegt.
- **Ausführung** - Hier wird die Liste der angelegten Algorithmen durchlaufen und nacheinander ausgeführt. Für jeden Algorithmus werden dabei die folgenden Teilschritte durchlaufen:
  1. Übermitteln der Punktwolken (von vorherigem Algorithmus falls sukzessiv)
  2. Ausführung des Algorithmus (Registrierung oder Sampling der Punktwolken)
  3. Speichern der generierten bzw. transformierten Punktwolken in SCENE
  4. Berechnung des Registrierungsfehler
  5. Exportieren und Verlinken der Ergebnisse

Im Wesentlichen wurden dafür fünf, sehr eng miteinander kooperierende, Klassen entworfen und in C++ implementiert. Augenmerk wurde dabei auf die Trennung von Verwaltung und Datenhaltung sowie Performanz und Speicherbedarf gelegt.

- **SCENE-Punktwolke** - Bei dieser Klasse handelt es sich um eine Repräsentation, der in SCENE gespeicherten Punktwolke. Über sie können Punktdaten von SCENE geladen, gespeichert, gefiltert und verändert werden. Dadurch bildet sie im Allgemeinen die Schnittstelle zwischen SCENE und der Testumgebung.
- **PCL-Punktwolke** - Einige der implementierten Algorithmen entstammen der PCL (Point Cloud Library) [Lib12a]. Sie arbeiten auf einer eigenen Speicherstruktur, welche durch diese Klasse repräsentiert wird. Diese Speicherstruktur ist in der Lage verschiedene Repräsentationen einer Punktwolke gleichzeitig zu speichern, den Zugriff auf diese zu realisieren und bei Bedarf in Dateien zu exportieren. Dazu gehören z.B. die Orientierungen von Punkten oder Deskriptoren (siehe Abschnitt 3.2.1). Auch bewerkstelligt sie die Konvertierung einer SCENE-Punktwolke in das PCL-Format und umgekehrt. Eine PCL-Punktwolke kann mit einer SCENE-Punktwolke verlinkt werden. Dadurch können Daten bei Bedarf geladen, entladen und gespeichert werden. Dies spart Speicherplatz und erlaubt es, sämtliche Zugriffe auf Punktwolken über diese Klasse zu bewerkstelligen.
- **Attribut-Container** - Der Attribut-Container ist eine Klasse, die in der Lage ist beliebige Datentypen zu verwalten. Diese können über einen eindeutigen Bezeichner gespeichert und ausgelesen werden. Ebenfalls bietet er die Möglichkeit alle gespeicherten Informationen als String serialisiert auszulesen. Über ihn können Parameter, Ergebnisse und Zustände kompakt ausgetauscht werden.
- **Algorithmen-Basisklasse** - Diese Klasse ist abstrakt und dient als normierte Schnittstelle zwischen den Algorithmen und einer beliebigen anderen Klasse. Von ihr muss ein Algorithmus abgeleitet werden, um in der Testumgebung ausgeführt werden zu können. Sie hält Instanzen von PCL-Punktwolken und Attribut-Containern. Außerdem stellt sie Funktionalitäten zum Zugriff auf diese bereit.
- **Management** - Die Management-Klasse ist für die Verwaltung und Koordination aller ablaufenden Prozesse zuständig. Dazu gehören die Analyse der Konfigurationsdateien, die Autogenerierung von Parametersets sowie die Beschaffung der Punktwolken. Außerdem ist sie für das Ausführen der einzelnen Algorithmen, die Fehlerberechnung und Aufbereitung bzw. Export der Ergebnisse zuständig.

### 5.3. Testszenarien

Zur Analyse der Registrierungsalgorithmen wurden Punktwolken aus drei verschiedenen Umgebungen mit Hilfe des Laserscanner Focus<sup>3D</sup> generiert. Die Testszenarien wurden so gewählt, dass typische Situationen aus allen Anwendungsbereichen repräsentiert werden (siehe Abbildung 5.1). Sie werden wie folgt unterschieden:

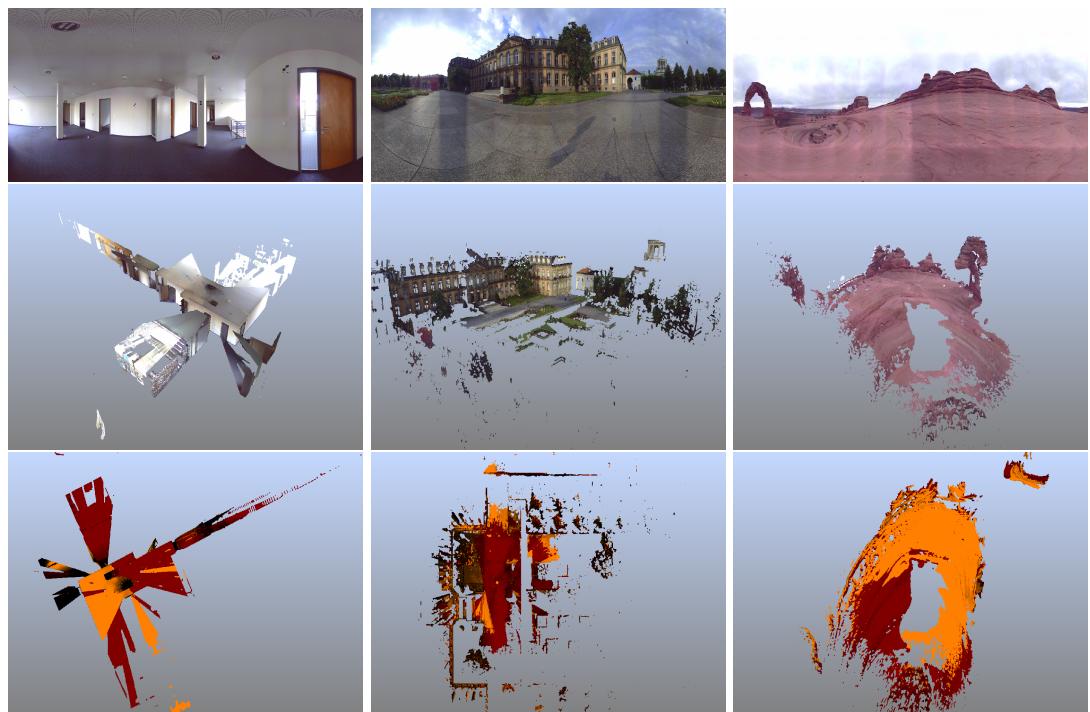
- **Indoor-Szenario** - Indoor-Szenarien zeigen sich häufig symmetrisch weil überwiegend Flächen, Kanten und Ecken vorhanden sind. Ihre Ausdehnung (5 – 20 Meter) ist im Allgemeinen gering. Sensorinformationen wie Kompass oder Inklinometer sind oft fehlerbehaftet, odometrische Daten jedoch können sehr genau erhoben werden. Die

## 5. Testumgebung

---

geringe Ausdehnung führt zu einem meist großen Überlappungsbereich. Dieser ist auf Grund von Verdeckungen jedoch nicht durchgängig.

- **Outdoor-Szenario** - Outdoor-Szenarien zeichnen sich durch überwiegend runde Oberflächen und sehr schwache Kanten aus. Ihre Ausdehnung entspricht meist der maximalen Reichweite des Laserscanner (120 m). Zur Bestimmung von Position und Lage können vielerlei Sensoren verwendet werden. Das trägt zu einer meist guten Bestimmung einer groben Ausrichtung bei. Der Überlappungsbereich ist dabei häufig durchgängig, die Größe stark von der Distanz der Laserscanner-Positionen abhängig.
- **Hybrid-Szenario** - Hybride Szenarien sind eine Mischung aus Indoor- und Outdoor-Szenarien. Sie beinhalten Flächen, Kanten, Ecken und Rundungen. Ihre Ausdehnung sowie deren Überlappungsbereich entspricht der eines Outdoor-Szenarios.



**Abbildung 5.1.: Testszenarien** - Verschiedene zur Analyse der Registrierungs-Algorithmen verwendete Testszenarien. Von links nach rechts: Indoor-Szene, Hybrid-Szene, Outdoor-Szene. Von oben nach unten: Planare Ansicht, 3D-Ansicht, Korrespondenz-Ansicht.

Die Punktwolken wurden manuell registriert. Eine der Punktwolken, im Folgenden als Zielpunktwolke bezeichnet, wurde dabei mit der Position des Laserscanners am Ursprung des gemeinsamen Koordinatensystems fixiert. Die Andere, im Folgenden als Datenpunktwolke bezeichnet, wurde transformiert. Die korrekte Transformation wird zur Berechnung des Registrierungsfehlers benutzt (siehe Kapitel 4) und vor der Prozessierung eines Algorithmus entfernt. Dadurch liegen beide Laserscanner-Positionen im Ursprung. Die Aufgabe

		Indoor-Szenario	Hybrid-Szenario	Outdoor-Szenario
Translation	<b>x (m)</b>	2.96	-1.29	-14.50
	<b>y (m)</b>	-1.33	7.18	7.33
	<b>z (m)</b>	0.00	-0.14	4.51
<b>Rotation</b>	<b>z (°)</b>	242.23	358.08	346.60

**Tabelle 5.1.: Korrekte Transformationen der Testszenarien** - Aufgeführt sind die zum korrekten Ausrichten der Datenpunktewolke notwendigen Translationen und Rotationen der einzelnen Testszenarien.

eines Registrierungsalgorithmus in der Testumgebung kann dann als das Wiederfinden der korrekten Transformation für die Datenpunktewolke bezeichnet werden.

Einige der Registrierungsalgorithmen sind in der Lage Vorabinformationen (Initiale Transformation) zu verarbeiten. Die von einem Inklinometer erhaltenen Daten sind äußerst zuverlässig und präzise. Sie werden direkt auf die Punktdaten angewendet und sind dadurch implizit vorhanden. Die verbleibende Rotation um die Z-Achse ( $z \in \mathbb{R}, 0 \leq z \leq 360$ ) sowie die Translation ( $\vec{t} = t_x, t_y, t_z \in \mathbb{R}^3$ ) werden simuliert. Dadurch ist ein szenenübergreifender Vergleich der Registrierungsergebnisse möglich. Ein initiales Parameterset  $\Gamma_{\vec{\xi}}^{\psi}$  ist dann wie folgt gegeben:

$$(5.1) \quad \Gamma_{\vec{\xi}}^{\psi} = \begin{cases} z + \psi, & \text{mit } (z + \psi) \in [0, 360], \psi \in \mathbb{R} \\ \vec{t} + \vec{\xi}, & \text{mit } \vec{\xi} \in \mathbb{R}^3 \end{cases}$$

Die Transformationen zur korrekten Registrierung der Datenpunktewolke an die Zielpunktewolke der einzelnen Testszenarien, sind in Tabelle 5.1 aufgeführt.



## 6. Sub-Sampling

**H**OCHAUFGELÖSTE Punktwolken, wie die hier betrachteten, enthalten bis zu 700 Millionen Punkte. Diese Datenmenge stellt selbst für heutige Prozessoren und parallelisierte Verarbeitungsschritte eine Herausforderung dar. Speziell bei der Registrierung ist diese Masse an Punkten oft hinderlich und in der Verarbeitung sehr zeitaufwändig.

Sub-Sampling beschreibt im allgemeinen Sinne die Reduktion von Daten. In diesem Fall ist damit das Verringern der Anzahl von Punkten in einer Punktwolke gemeint und wird als eine Veränderung der Auflösung der Umgebung bezeichnet. Dies wird realisiert, indem von einer Punktwolke  $P$  eine Teilmenge  $P'$  bestimmt wird, sodass  $P' \subseteq P \wedge P' \neq \emptyset$  gilt. Definiert seien diese dabei wie in Gleichung 3.2.

Bei den hier verwendeten sphärischen Punktwolken (siehe Abschnitt 2.2.1) lassen sich im Allgemeinen zwei Klassen des Sub-Sampling definieren [MMRM10]. Diese unterscheiden sich in der Berücksichtigung der Entfernungsinformation.

In den folgenden Abschnitten wird ein Einblick in die für diese Diplomarbeit relevanten Sub-Sampling Verfahren gegeben. Behandelt werden dabei das zufällige-, diskrete- und räumliche Sub-Sampling.

### 6.1. Entfernungsunabhängiges Sub-Sampling

Wie durch die Bezeichnung suggeriert, handelt es sich bei dieser Art des Sub-Sampling um Verfahren, welche die räumliche Lage der Punkte einer Punktwolke nicht berücksichtigen. Nach welchem Muster die zur Repräsentation der Punktwolke herangezogenen Punkte ausgewählt werden, unterscheidet die verschiedenen Verfahren.

- **Diskret** - Diskret bedeutet in diesem Zusammenhang, dass Punkte in einer diskreten Schrittweite ausgewählt werden. Als Reihenfolge wird dabei Zeile für Zeile des  $2^{\frac{1}{2}}D$  Rasters, beginnend mit der dem Laserscanner am nächsten liegenden, verwendet. Soll die Punktwolke  $P$  mit  $n \in \mathbb{N}$  Punkten auf  $m \in \mathbb{N}$  Punkte reduziert werden, so wird jeder  $\lfloor m/n \rfloor$ -te Punkt in  $P'$  übernommen. Ist  $n \bmod m \neq 0$ , dann gilt  $\lfloor m/n \rfloor \cdot n < m$ . Dies bedeutet, dass nicht  $m$  Intervalle selber Größe möglich sind. Ausgeglichen werden kann dies, indem das letzte Element pauschal zu  $P'$  hinzugefügt oder ignoriert wird.

Diskretes Sub-Sampling kann schnell und speichereffizient realisiert werden. Außerdem bleibt die Eigenschaft der Dichteverteilung einer Punktwolke (siehe Abbildung 6.1) erhalten. Sich in unmittelbarer Nähe des Laserscanner befindende Bereiche sind nach dem Sampling noch immer dichter repräsentiert als ferne. Der Zeitaufwand ist hierbei

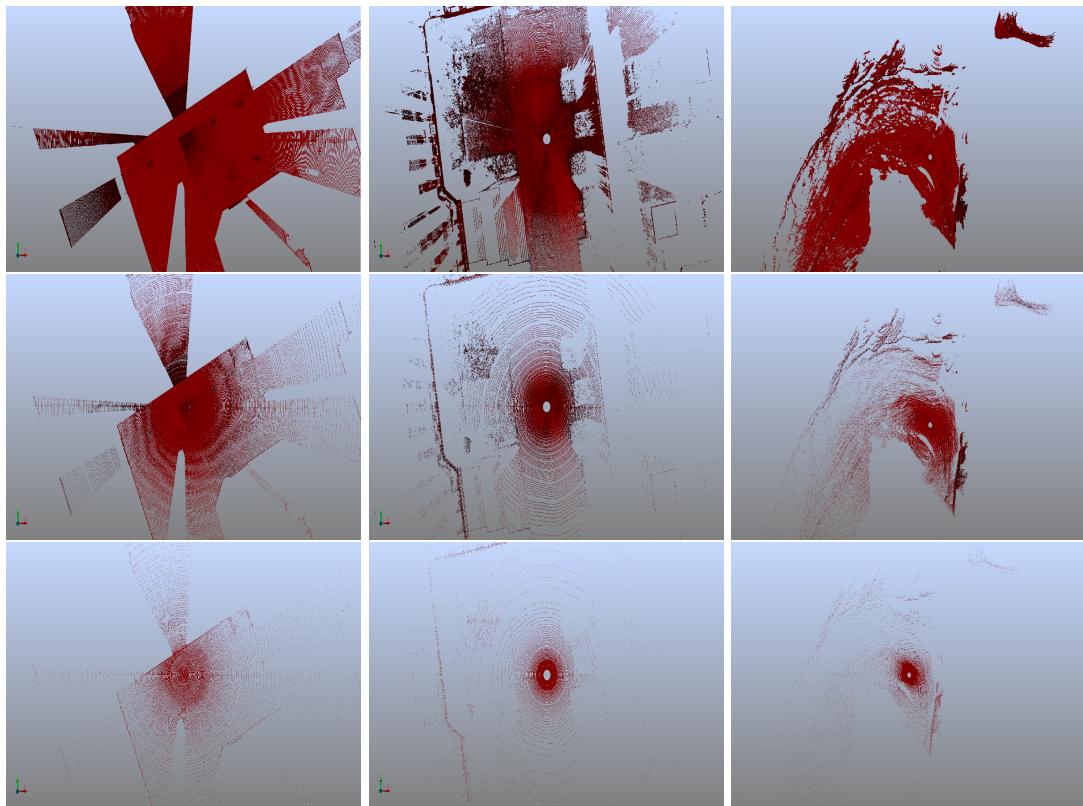
## 6. Sub-Sampling

---

zu vernachlässigen, da lediglich  $P$  durchlaufen werden muss und die entsprechenden Punkte nach  $P'$  kopiert werden.

Zusätzlicher Zeitaufwand erfordert diese Art des Sub-Sampling nicht, da bereits beim Laden der Punktwolken entschieden werden kann, welcher Punkt in  $P'$  enthalten sein soll. Dem Zeitvorteil entgegen wirkt jedoch, wie bei allen entfernungsunabhängigen Sub-Sampling Verfahren, die resultierende Repräsentation der gesamten Szene.

Ein dem diskreten Sub-Sampling entsprechendes Verfahren wird von der Software SCENE [FAR12] bereitgestellt. Dieses wird zum Sub-Sampling der in Abschnitt 5.3 vorgestellten Punktwolken verwendet. Als repräsentative Auflösungen wurden diese auf 1K, 10K, 100K, 1M, 10M (K=Tausend, M=Millionen) Punkte reduziert. Optimierungen der verwendeten Variante führen dazu, dass keine Reduktion auf eine exakte Punktmenge erfolgen kann. Die verschiedenen Auflösungen enthalten dadurch Abweichungen in der Punktmenge von  $\pm 2\%$ .



**Abbildung 6.1: Diskretes Sub-Sampling** - Verschiedene Szenen in unterschiedlichen Auflösungsstufen mittels diskretem Sub-Sampling bearbeitet. Von links nach rechts: Indoor-Szene, Hybrid-Szene, Outdoor-Szene. Von oben nach unten: Volle Auflösung, 1M, 100K Punkte. Eine hohe Punktdichte ist im Nahbereich des Laserscanners stets erhalten.

## 6.2. Entfernungsabhängiges Sub-Sampling

Beim entfernungsabhängigen, auch räumliches oder uniformes Sub-Sampling genannt, werden Punkte ihrer räumlichen Lage entsprechend betrachtet. Dazu wird über eine Punktwolke  $P$ , ausgehend vom Ursprung (Position des Laserscanners), ein Raster erzeugt, welches diese in einzelne Zellen und somit Teilpunktewolken  $P_i$  teilt. Meist (so auch bei dem hier verwendeten Verfahren) wird dazu eine Voxel-Grid benutzt, dessen Zellen aus Quadern bestehen. Jede Teilpunktewolke soll nach dem Sub-Sampling durch nur einen Punkt in  $P'$  repräsentiert werden. Die Wahl des Repräsentanten kann dabei auf beliebige Art und Weise getroffen werden. Sinnvoll erscheint dabei meist der Schwerpunkt ( $S$ ) einer Teilpunktewolke.

Sei  $k \in \mathbb{N}$  die Anzahl der Punkte einer Teilpunktewolke und seien diese, wie in Gleichung (3.2) definiert, dann berechnet sich dieser wie folgt:

$$(6.1) \quad S = \frac{1}{k} \sum_{i=1}^k P_i,$$

Im Allgemeinen gilt dabei  $S \notin P$ .  $P'$  entspricht dadurch einer Approximation der in  $P$  enthaltenen Daten.

Die Dichte der resultierenden Punktwolke  $P'$  ist weitgehend gleich verteilt. Lediglich an den Rändern der Zellen können Ballungen auftreten. Damit bleibt die Dichteigenschaft der sphärischen Punktwolke nicht erhalten. Sie wird ersetzt durch eine im Wesentlichen der Größe einer Raster-Zelle entsprechenden. (siehe Abbildung 6.3). Dadurch ist auf jeder Auflösung stets die gesamte Szene repräsentiert. Wie detailliert einzelne in der Szene vorkommenden Objekte dabei vertreten sind, hängt von der Anzahl der Punkte ab. Im Allgemeinen kann jedoch davon ausgegangen werden, dass jedes Objekt relativ zu seiner Größe mit ähnlich vielen Punkten und somit in selbem Maße detailliert repräsentiert ist.

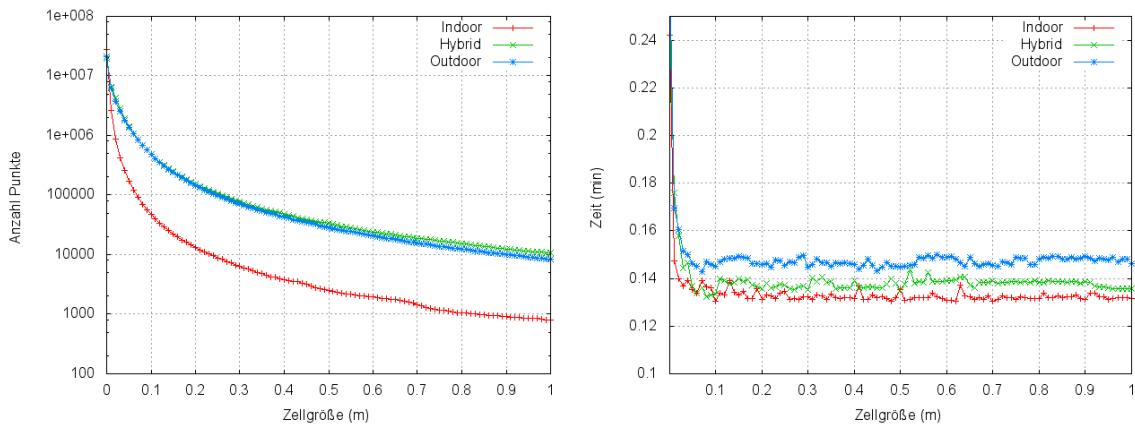
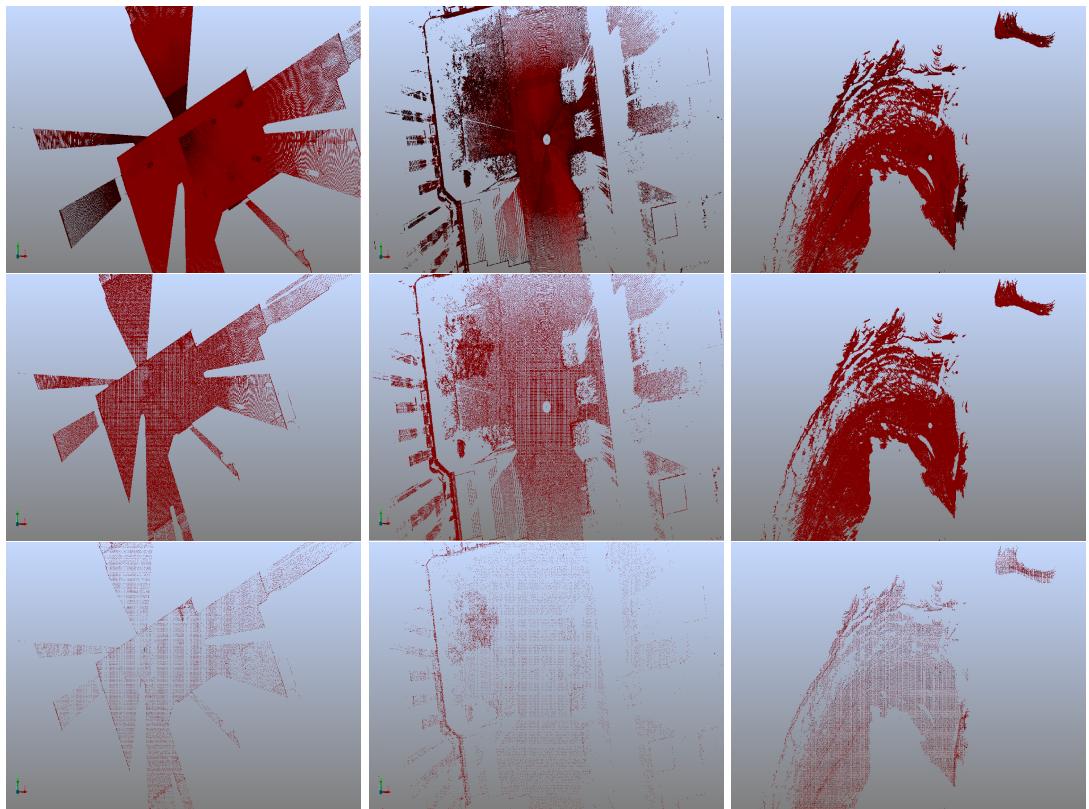


Abbildung 6.2.: Räumliches Sub-Sampling Zeit-Punkt-Zellgrößen-Vergleich - *Oben:* Die Sampling-Rate ist im wesentlichen von der Ausdehnung einer Szene bestimmt. *Unten:* Aus der Wahl des Schwerpunktes als Repräsentant einer Zelle, resultiert ein konstanter Zeitaufwand.

## 6. Sub-Sampling

---



**Abbildung 6.3.: Räumliches Sub-Sampling** - Verschiedene Szenen in unterschiedlichen Auflösungsstufen, mittels räumlichem Sub-Sampling bearbeitet. Von links nach rechts: Indoor-Szene, Hybrid-Szene, Outdoor-Szene. Von oben nach unten: Volle Auflösung, 1M, 100K. Eine gleichmäßige Punktdichte ist zu sehen.

Die Anzahl verbleibender Punkte bei variierender Zellgröße in  $P'$  ist in erster Linie von der Ausdehnung einer Punktwolke  $P$  abhängig. Ausgedehnte Punktwolken, wie sie meist in Outdoor- und Hybrid-Szenen entstehen, erfahren bei steigender Zellgröße eine geringere Datenreduktion, als Indoor-Szenen (siehe Abschnitt 5.3). Diese zeichnen sich durch eine geringe Ausdehnung und eine gleichmäßige räumliche Dichte aus (siehe Abbildung 6.2).

Der Zeitaufwand dieses Verfahrens ist im Allgemeinen durch das Berechnen des Repräsentanten in jeder Rasterzelle bestimmt. Im Falle des Schwerpunktes ist dieser konstant und dadurch von der Anzahl der Punkte in einer Punktwolke abhängig. Mit sinkender Zellgröße steigt die Anzahl der Zellen. Der zur Verwaltung der Zellen notwendige Mehraufwand wirkt sich ebenfalls auf die Laufzeit aus (siehe Abbildung 6.2).

Das hier beschriebene Verfahren des entfernungsabhängigen Sub-Sampling wird im Späteren als Vorverarbeitung zur Registrierung der in Abschnitt 5.3 vorgestellten Punktwolken verwendet. Als repräsentative Auflösungen wurden diese auf 1K, 10K, 100K, 1M, 10M (K=Tausend, M=Millionen) Punkte reduziert. Da keine exakte Reduktionsrate bestimmt werden kann, enthalten die verschiedenen Auflösungen Abweichungen von  $\pm 2\%$  in der tatsächlichen Punktmenge.

## 6.2. Entfernungsabhängiges Sub-Sampling

---

Der verwendete Programmcode entstammt der Point Cloud Library [[Lib12j](#)] dessen Korrektheit vorausgesetzt wird. Zur Nachbarschaftssuche wird dabei ein KD-Tree verwendet.



## 7. Point Feature Histogram (PFH)

Der Point Feature Histogram (PFH) Ansatz entstammt der Idee einer statistischen Beschreibung von Punktumgebungen. Wie bereits in Abschnitt 3.2.1 erwähnt, wurde diese im Jahre 2003 erstmals von [WHHo3] unter dem Namen „Surflet-Pair-Relation Histograms“ (SPRH) eingeführt. Verwendet wurden diese zur Objekterkennung. Dabei werden SPRH's für ein Objekt erzeugt und in einer Datenbank abgelegt. Anschließend werden die für eine Punktwolke erzeugten SPRH's mit dieser Datenbank abgeglichen. Rusu et al. nutzten diese in [RMBBo8] zur Segmentierung geometrischer Strukturen. Vorteilhaft war dabei, dass Histogramme bestimmter geometrischer Strukturen, einzigartige Ausprägungen aufweisen. Erweitert und ausführlicher untersucht wurden sie dann in [RBMBBo8, RBB09, Ruso9]. Dabei wurde auch der Begriff Point Feature Histograms geprägt sowie deren Verwendung zur Registrierung von 3D-Szenen. Anzusiedeln ist der PFH-Ansatz in der Klasse der deskriptorisierten Grobregistrierung. Daher ist er nicht in der Lage Informationen einer initialen Ausrichtung zu verarbeiten.

In diesem Kapitel wird das Funktionsprinzip des PFH-Algorithmus erläutert. In Abschnitt Erweiterungen werden Erweiterungen vorgestellt, die Einfluss auf Qualität und Laufzeit nehmen. Zuletzt wird eine Variante vorgestellt die dann in Kapitel 11 auf hochauflösten Punktwolken verwendet wird.

### 7.1. Funktionsprinzip

Soll eine Datenpunktwolke  $Q$  an eine Zielpunktwolke  $P$ , definiert wie in Gleichung 3.2 registriert werden, dann ist die grundlegende Idee des PFH-Algorithmus für jeden Punkt der beiden Punktwolken ein PFH seiner Umgebung zu erstellen. Anschließend werden die PFH aus  $P$  mit denen aus  $Q$  verglichen. Ähnliche PFH weisen auf gleiche Punkte hin und werden als korrespondierend betrachtet. Aus ihrer Punktkoordinaten wird dann die Transformation berechnet, die alle korrespondierenden Punkte bestmöglich zueinander ausrichtet (siehe Anhang A).

Das PFH eines Punktes  $p_q \in P$  beschreibt die Oberfläche  $C_q \subset P$  auf der sich dieser befindet. Definiert ist  $C_q$  über eine Kugelumgebung mit Radius  $k \in \mathbb{R}$  und Mittelpunkt  $p_q$ . Alle sich in dieser Kugelumgebung befindenden Punkte  $p_{k_i} \in P, 1 \leq i \leq |P|$  bilden  $C_q$ . Im PFH wird  $C_q$  über die Distanz- und Orientierungsverhältnisse der sich auf  $C_q$  befindenden Punkte repräsentiert. (siehe Abbildung 7.1). Diese werden für jedes sich in  $C_q$  befindende Punktpaar berechnet. Die vollständige Berechnung eines PFH liegt dadurch in der Komplexitätsklasse

## 7. Point Feature Histogram (PFH)

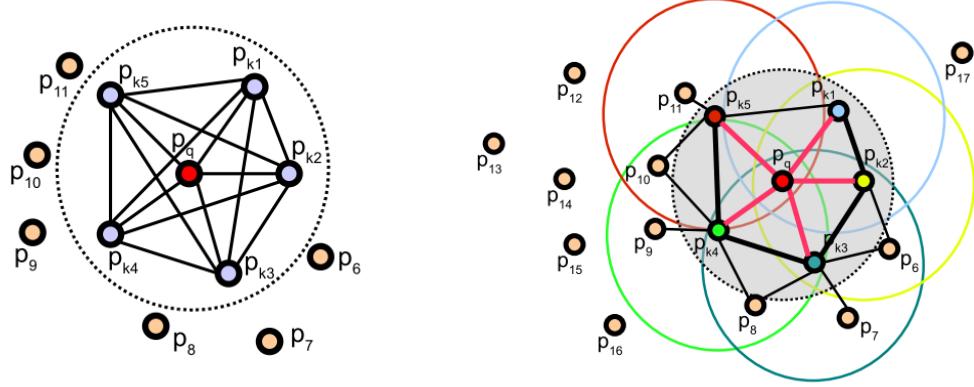


Abbildung 7.1.: Point Feature Histogramm Deskriptor - Links: Das für den Ausgangspunkt ( $p_q$ ) berechnete PFH beschreibt die Oberfläche einer spezifizierten Nachbarschaft (gestrichelter Kreis). Dazu werden Distanz- und Orientierungsverhältnisse zwischen allen darin enthaltenen Punktpaaren ( $p_{ki}, i = [1, 5] \in \mathbb{N}$ ) erfasst [Rus09]. Rechts: Erweiterte Nachbarschaft des Ausgangspunktes ( $p_q$ ), um die Kugelumgebungen (weiß hinterlegte Kreise) seiner Nachbarn ( $p_{ki}, i = [1, 5] \in \mathbb{N}$ ), zur Berechnung des FPFH.

$\mathcal{O}(|C_p|^2)$ . Für eine Punktwolke mit  $n \in \mathbb{N}$  Punkten ergibt sich eine Gesamt-Komplexität von  $\mathcal{O}(n|C_p|^2)$ .

Das Distanz- und Orientierungsverhältnis eines Punktpaars wird im PFH in kodierter Form festgehalten. Primäres Ziel davon ist, ein rotationsinvariantes Bezugssystem [WHHo3, RMBBo8] für diese zu erzeugen. Bewerkstelligt wird dies mit dem sogenannte Darboux-Frame (siehe Abbildung 7.2) [RMBBo8]. Seien  $p_s, p_t \in P$  als zwei sich in der Nachbarschaft von  $p_q$  befindenden Punkte definiert, weiter seien ihre zugehörigen Normalen durch  $n_s$  und  $n_t$  gegeben und sind diese einheitlich zum Standpunkt des Laserscanners ausgerichtet, so berechnet sich der Darboux-Frame mit Ursprung in  $p_s$  wie folgt:

$$(7.1) \quad \begin{aligned} u &= n_s \\ v &= v \times \frac{p_t - p_s}{\|p_t - p_s\|_2} \\ w &= u \times v \end{aligned}$$

Basierend auf diesem Bezugssystem wird das Distanz- und Orientierungsverhältnis eines Punktpaars durch das folgenden Quadrupel kodiert:

$$(7.2) \quad \begin{aligned} d &= \|p_t - p_s\|_2 \\ \alpha &= v \cdot n_t \\ \phi &= u \cdot \frac{(p_t - p_s)}{d} \\ \theta &= \arctan w \cdot n_t, u \cdot n_t \end{aligned}$$

Die ursprüngliche Information zweier Punkte und ihrer Normalen (Distanz und Orientierung) ist dann durch vier, statt zwölf Parameter repräsentiert.

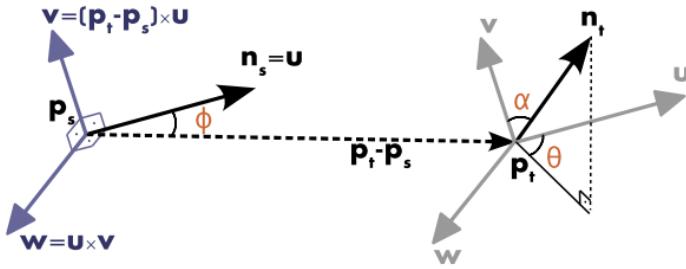


Abbildung 7.2.: Darboux Frame - Geometrische Veranschaulichung des Darboux-Frame ( $u, v, w$ ) sowie das berechnete Quadrupel  $(p_t - p_s, \alpha, \phi, \theta)$  für ein Punktpaar  $(p_t, p_s)$  und dessen Normalen  $n_t$  und  $n_s$ . [Rus09].

Das, für jedes in  $C_p$  vorkommende Punktpaar, berechnete Quadrupel muss nun im PFH vermerkt werden. Dies erfordert eine Diskretisierung der im Quadrupel kontinuierlich vorliegenden Parameter. Dazu wird der Wertebereich jedes Parameters in uniforme Klassen aufgeteilt. Sei die Anzahl der Klassen pro Parameter  $b$ , ergibt sich ein PFH der Größe  $b^4$  (bei zwölf Parametern  $b^{12}$ ). Jeder Parameterwert kann dadurch einer Klasse zugewiesen werden. In ihm werden dann die Häufigkeiten der vorkommenden Parameterwerte prozentual zu  $|C_p|$  festgehalten.

Zum Vergleich der so generierten PFH existieren vielerlei Metriken. Ihre Beschreibung ist im folgenden Abschnitt (7.2) zu finden.

## 7.2. Erweiterungen

In diesem Abschnitt werden Erweiterungen des PFH-Algorithmus vorgestellt. Diese zielen auf das Umgehen spezieller Probleme sowie eine Optimierung des Standardansatzes (siehe Abschnitt 7.1) ab. Im folgenden sind einige dieser, kategorisiert nach ihrem Wirkungsbereich, erläutert. Eine umfassende Betrachtung aller vorhandenen Erweiterungen und Varianten führt an dieser Stelle zu weit.

### 7.2.1. Punktwahl

Viele der berechneten PFH einer Punktwolke, speziell benachbarter Punkte, sind sehr ähnlich. Dadurch sind bestimmte Bereiche überrepräsentiert. Die Folge ist, dass sehr häufig falsche Korrespondenzen gebildet werden.

Eine Lösung für dieses Problem ist die Betrachtung nur einer besonders aussagekräftigen Teilmenge der Punktwolke. Sie kann dabei zufällig, über räumliches Sub-Sampling oder Keypoint-Detektoren, (wie z.B. 3D-SIFT [SCM10]) bestimmt werden. Ein Nachteil dieser Ansätze ist, dass Teilmengen in beiden Punktwolken bestimmt werden können, die keinen gemeinsamen Überlapp mehr bilden.

Eine etwas andere Variante (Persistente PFH) versucht sehr ähnliche PFH zu verwerfen [RBMB08]. Hierzu wird für jeden Punkt einer Punktwolke ein PFH berechnet. Aus allen PFH wird dann ein Durchschnittshistogramm  $PFH_\mu$  bestimmt. Anschließend wird für jedes PFH die Distanz zu  $PFH_\mu$  berechnet. Aus dieser lässt sich dann eine Standardabweichung  $\sigma$  zum Durchschnittshistogramm ermitteln. Verworfen werden nun alle PFH, deren Distanz zu  $PFH_\mu$  kleiner als  $\alpha \cdot \sigma$  ist. PFH deren Distanz größer ist, werden als einzigartig betrachtet. Optional wird dies zusätzlich über verschiedene Berechnungs-Radien prozessiert. Ist ein PFH über mehrere Radien hinweg einzigartig, wird es als persistent betrachtet. Persistente PFH sind besonders aussagekräftig, da sie im Allgemeinen robust sind und sich voneinander unterscheiden.

Wird zu einem persistenten PFH ein sehr ähnliches gefunden, kann mit hoher Wahrscheinlichkeit von einer korrekten Korrespondenz ausgegangen werden. Der Nachteil dieser Erweiterung steckt in der Laufzeit. Sowohl der Vergleich zum Durchschnittshistogramm als auch die Berechnung der PFH über mehrere Skalen hinweg ist sehr rechenaufwändig.

### 7.2.2. PFH Größe

Die Größe eines PFH ist im wesentlichen von der Anzahl (Quadrupel) und der Diskretisierungsrate ( $b$ ) der Parameter abhängig. Grundlegend kann gesagt werden, dass ein PFH, je größer es ist, die dadurch beschriebene Oberfläche umso genauer repräsentiert. Gleichzeitig gilt aber auch, dass je größer das PFH ist, desto zeitaufwändiger der spätere Vergleich in der Korrespondenzsuche ist.

Rusu [Rus09] kam zu dem Ergebnis, dass die Distanzinformation zwischen zwei Punkten einer  $2\frac{1}{2}$ D-Punktwolke nur sehr geringfügige Aussagekraft über die beschriebene Oberfläche besitzt. Dies basiert auf der Tatsache, dass die Abstände benachbarter Punkte durch das Aufnahmeprinzip (siehe Abschnitt 2.2.1) bedingt, nur sehr gering und meist uniform sind. Diese Erkenntnis erlaubt eine Reduktion des Quadrupels ( $\langle d, \alpha, \phi, \theta \rangle$ ) auf ein Triplet der Form  $\langle \alpha, \phi, \theta \rangle$ . Die Größe des PFH reduziert sich dadurch auf  $b^3$ . Sie ermöglicht es die verbleibenden, aussagekräftigeren Informationen genauer abzubilden (erhöhen von  $b$ ).

### 7.2.3. Einflussbereich

Der PFH-Deskriptor betrachtet jedes mögliche Punktpaar der zu beschreibenden Oberfläche  $C_p$ . Seine Berechnungskomplexität liegt dadurch in  $\mathcal{O}(|C_p|^2)$ . Sollen grobe Strukturen als Merkmale verwendet werden, muss der Radius der Kugelumgebung erhöht werden. Dadurch wird eine größere Oberfläche betrachtet,  $|C_p|$  wächst und mit ihr auch die Rechenzeit.

Der in [RBB09] vorgestellte FPFH (Fast Point Feature Histogram) Deskriptor reduziert die quadratische Komplexität des PFH-Deskriptor. Sei  $p \in P$  dabei der Punkt für den ein FPFH berechnet werden soll. In der Menge  $K$  sind alle die Punkte enthalten, die sich in der Kugelumgebung mit Radius  $r$  und Mittelpunkt  $p$  befinden. Ein SPFH (Simplified Point Feature Histogram) wird dann für  $p$  berechnet, indem das Triplet  $\langle \alpha, \phi, \theta \rangle$  zwischen

$p$  und jedem Punkt in  $K$  berechnet wird. Anschließend wird die Nachbarschaft von  $p$  um die Kugelumgebungen der in  $K$  enthaltenen Punkte, erweitert (siehe Abbildung 7.1). Das resultierende FPFH für  $p$  wird dann wie folgt berechnet:

$$(7.3) \quad FPFH(p) = SPFH(p) + \frac{1}{|K|} \sum_{k=1}^{|K|} \frac{1}{\omega_k} \cdot SPFH(k)$$

Dabei ist  $\omega_k$  eine von der Distanz zwischen  $p$  und  $k \in K$  abhängige Gewichtung, die den Einfluss der erweiterten Nachbarschaften bestimmt. Sei  $C_p$  die gesamte, betrachtete Punktmenge bei der Berechnung des FPFH für  $p$ , dann ist die resultierende Komplexität des FPFH-Deskriptor  $\mathcal{O}(|C_p|)$ .

#### 7.2.4. Metrik

Zur Identifikation korrespondierender PFH müssen diese miteinander verglichen und deren Ähnlichkeit evaluiert werden. Sowohl Geschwindigkeit als auch die Aussagekraft sind dabei beeinflussbare Größen.

Hierzu wurden in [WHH03, Rus09] verschiedene Metriken vorgestellt. Dabei bezeichnet  $k$  die Klassenanzahl zweier PFH ( $H$  und  $H'$ ),  $H(i)$  und  $H'(i)$ , die für die Klasse  $1 \leq i \leq d$  berechneten Werte (siehe Gleichung 7.5).

Untersuchungen von [WHH03] und [Rus09] zufolge liefert die Kullback-Leibler Divergenz bei höchstem Zeitaufwand die besten Ergebnisse. Die Unterschiede zum Chi-Quadrat-Test und der Bhattacharyya Distanz sind dabei gering. Qualitativ am schlechtesten schneiden die restlichen Metriken ab, sind jedoch deutlich schneller. Prinzipiell kann gesagt werden, dass sich die Qualität konträr zur Laufzeit verhält. Dabei sei beachtet, dass sich abhängig von der Szene unterschiedliche Metriken profilieren können.

$$\begin{aligned}
 (7.4) \quad d_{\cap}(H, H') &= \sum_{i=1}^k \min(H(i), H'(i)) \quad \text{(Intersection)} \\
 d_{L_1}(H, H') &= \sum_{i=1}^k (H(i) - H'(i)) \quad \text{(Manhattan Distanz)} \\
 d_{L_2}(H, H') &= \sqrt{\sum_{i=1}^k (H(i) - H'(i))^2} \quad \text{(Euklidische Distanz)} \\
 d_{JM}(H, H') &= \sqrt{\sum_{i=1}^k \left( \sqrt{H(i)} - \sqrt{H'(i)} \right)^2} \quad \text{(Jeffries-Matusita)} \\
 d_{\epsilon}(H, H') &= \sum_{i=1}^k (H(i) - H'(i))^2 \quad \text{(Summed Squared Distanz)} \\
 d_B(H, H') &= \ln \sum_{i=1}^k \sqrt{(H(i) - H'(i))} \quad \text{(Bhattacharyya Distanz)} \\
 d_{\chi_1^2}(H, H') &= \sum_{i=1}^k \frac{(H(i) - H'(i))^2}{H(i)} \quad \text{(Chi-Quadrat Test, 1.Form)} \\
 d_{\chi_2^2}(H, H') &= \sum_{i=1}^k \frac{(H(i) - H'(i))^2}{H(i) - H'(i)} \quad \text{(Chi-Quadrat Test, 2.Form)} \\
 d_{\kappa(H, H')} &= \sum_{i=1}^k (H(i) - H'(i)) \ln \frac{H(i)}{H'(i)} \quad \text{(Kullback-Leibler)}
 \end{aligned}$$

### 7.3. Analysierte Variante

Dieser Abschnitt erläutert die zur Analyse implementierte Variante des PFH-Algorithmus. Der Fokus wurde dabei auf die Laufzeit gelegt. Zum Einsatz kommt daher ein Persistenz-Test der nur eine Skala betrachtet. Im Folgenden sind die einzelnen Schritte des Algorithmus vorgestellt.

Im ersten Schritt wird für jeden Punkt beider Punktwolken die Normale bestimmt. Das Einflussgebiet für diese ist über den Radius einer Kugelumgebung definiert. Für die darin enthaltenen Punkte wird eine Principal Component Analysis der Kovarianzmatrix durchgeführt (siehe Kapitel B). Anschließend werden alle Normalen in Richtung des Laserscanner-Standpunktes ausgerichtet. Verwendet wurde dabei das in der Point Cloud Library verfügbare Modul [Lib12h]. Die dabei verwendete Datenstruktur entspricht einem KD-Tree (siehe Abschnitt 2.3.3) um notwendige Suchoperationen performant durchführen zu können.

Basierend auf den Normalen wird für jeden Punkt beider Punktwolken ein FPFH [Lib12e] berechnet. PFH sind auf Grund des zu hohen Rechenaufwandes (Tests ergaben bei diskret re-

duzierten Punktdaten von 100K und einem Radius von 0,25 Meter, Laufzeiten von bis zu 20 Minuten) nicht performant. Dabei wird das Triplett  $\langle \alpha, \phi, \theta \rangle$  verwendet. Der Wertebereich jedes Parameters ist dabei auf 11 Klassen festgelegt. Diese werden konateniert, sodass ein Histogramm der Größe 33 entsteht. Es wird dann auf Persistenz bzw. Einzigartigkeit geprüft, indem ein Vergleich mit dem Durchschnittshistogramm durchgeführt wird (Standardabweichung durch Normalverteilung approximiert [Rus09]). Dabei wird, auf Grund ihrer hohen Aussagekraft, als Metrik die Kullback-Leibler-Divergenz verwendet. Sie soll einen Ausgleich zu einem nicht über mehrere Skalen hinweg, durchgeföhrten Test der Einzigartigkeit bilden. Verwendet wurde dabei die bereits vorhandene Implementierung der Point Cloud Library [Lib12g, Lib12e].

Die so entstandenen FPFH werden anschließend auf Korrespondenzen untersucht. Zum Vergleich der FPFH wird dabei die Merkmalsdistanz verwendet. Da ein FPFH vektorisiert werden kann, lässt es sich in einem KD-Tree organisieren. Das, bezogen auf die Merkmalsdistanz, ähnliche FPFH wird dann über eine zeiteffiziente Suche des nächsten Nachbarn bestimmt. Verwendet wurde dabei die der Point Cloud Library entstammende Variante [Lib12c]

Die resultierenden Korrespondenzen müssen anschließend auf Eindeutigkeit geprüft werden. Das ist notwendig, da ein FPFH nächster Nachbar zu beliebig vielen anderen sein kann. Tritt dieser Fall ein, wird nur jene Korrespondenz verwendet, welche die höchste Ähnlichkeit aufweist. Zusätzlich werden alle Korrespondenzen verworfen, deren Ähnlichkeitsmaß (Merkmalsdistanz) zueinander einen spezifizierten Grenzwert überschreitet. Entfernt werden so Korrespondenzen, deren korrespondierenden FPFH sehr unähnlich sind.

Abschließend wird über die Punktinformationen der resultierenden Korrespondenzen die Transformation bestimmt, die alle korrespondierenden Punkte bestmöglich zueinander ausrichtet. Dies erfolgt mit Hilfe der Singulärwertzerlegung (siehe Anhang A) [Lib12d].

Eine Korrektheit externer Implementierungen wird dabei vorausgesetzt.



## 8. Spin-Images (SI)

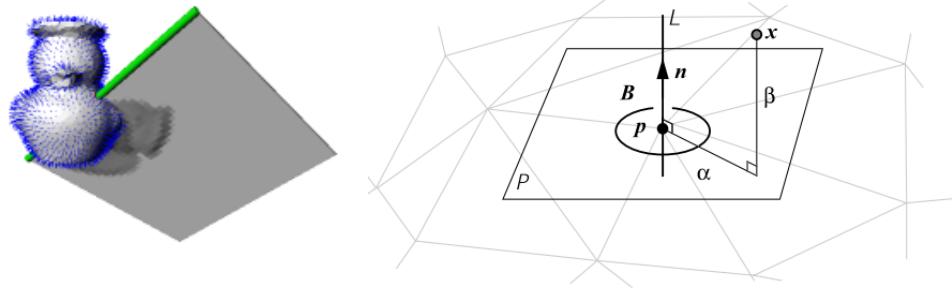
Das Prinzip der Spin-Images (SI) wurde 1997 von Johnson [Joh97] publiziert. Entwickelt wurden sie zur Objekterkennung. Die Idee dabei ist, von den zu erkennenden Objekten eine SI Repräsentation anzufertigen und in einer Datenbank zu speichern. Zur Laufzeit werden dann die aus einer Szene erzeugten SI mit denen der Datenbank abgeglichen. Gibt es Übereinstimmungen, wurde ein Objekt erkannt. Zusätzlich kann dadurch auch die Ausrichtung eines erkannten Objektes bestimmt werden. Diesem Prinzip entsprechend sollen im folgenden die SI für zwei Punktwolken erzeugt und miteinander verglichen werden, um Korrespondenzen zu identifizieren. SI können keine Lage und Positionsinformationen berücksichtigen, sind über einen Deskriptor definiert und gehören dadurch zur Klasse der deskriptorbasierten, grobregistrierenden Algorithmen (siehe Abschnitt 3.2.1).

Im Laufe dieses Kapitels wird das Funktionsprinzip der SI nach [Joh97] erläutert und auf Erweiterungen dieser eingegangen. Abschließend wird die zur Analyse herangezogene Variante vorgestellt.

### 8.1. Funktionsprinzip

SI sind Deskriptoren, die für einzelne Punkte der Punktwolken berechnet werden. Für diese beschreiben sie ihre darunterliegende Oberfläche. Hierzu wird für jeden Punkt eine Transformations-invariante Rotationsachse definiert. Wird um diese rotiert, werden alle sich in einem definierten Einzugsgebiet befindlichen Punkte besucht. Diese werden dann in einem 2D-Bildarray festgehalten. Anschaulich kann das mit einem um eine Achse rotierenden Blatt verglichen werden, auf welches die durchlaufenen 3D-Punkte aufgezeichnet werden (siehe Abbildung 8.1). Bereiche an denen viele Punkte aufgetragen werden sind dunkler repräsentiert als andere.

Zur Erzeugung eines SI muss für jeden Punkt der Punktwolken eine Normale über ein definiertes Einflussgebiet (Kugelumgebung mit Radius  $r$ ) berechnet werden. Die Orientierung der Normale muss für alle einheitlich erfolgen, um eine Invarianz gegenüber Transformationen zu gewährleisten. Für Punktwolken wird dafür im Allgemeinen der Laserscanner-Standort verwendet. Sei  $p$  ein Punkt einer Punktwolke, für den ein SI berechnet werden soll und  $n$  beschrieben als die zugehörige Normale sowie  $P$  die zur Bestimmung der Normalen berechnete Tangentialebene, auf welcher sich  $p$  befindet, dann ist eine transformationsinvariante Basis durch  $(p, n)$  definiert. Die Rotationsachse  $L$  wird als eine zu  $n$  parallel und durch  $p$  verlaufende Linie definiert. Ein in der Umgebung von  $p$  liegender Punkt wird dann über eine radiale und axiale Distanz  $\alpha \in \mathbb{R}$  und  $\beta \in \mathbb{R}$  in zylindrischen Koordinaten beschrieben. Eine



**Abbildung 8.1.: Spin Image Deskriptor -** Links: Ein SI wird als eine Abtragung aller 3D-Punkte betrachtet, die bei einer Ebenen-Rotation um eine definierte Achse besucht werden [JH99]. Rechts: SI Parameter für einen Punkt  $p$ . Nachbarpunkte werden über die Basis  $(p, n_p)$  in zylindrischen Koordinaten  $(\alpha, \beta)$  dargestellt [Joh97].

Rotation von  $360^\circ$  um  $L$  definiert diese zylindrische Umgebung von  $p$ . Alle darin enthaltenen Punkte werden wie folgt in das 2D-SI  $S_p$  abgebildet:

$$(8.1) S_p : \mathbb{R}^3 \mapsto \mathbb{R}^2, \quad S_p(x) \mapsto (\alpha, \beta) = \left( \sqrt{\|x - p\|_2 - (n \cdot (x - p))^2}, n \cdot (x - p) \right)$$

Da  $\alpha$  und  $\beta$  auch negativ sein können, hat das theoretische SI eine Größe von  $2\alpha \times 2\beta$ . Zur tatsächlichen Repräsentation in einem 2D-Bildarray muss der Wertebereich von  $\alpha$  und  $\beta$  diskretisiert werden. Zur Vereinfachung wird dabei ein quadratisches SI der Größe  $W \times W$  mit  $W \in \mathbb{N}$  verwendet. Sei  $b$  die Schrittweite definiert als  $W = b \cdot i + 0$  und  $W = b \cdot j + 0$ , wobei  $i \in \mathbb{N}$  die Anzahl der Zeilen des SI und  $j \in \mathbb{N}$  die Spalten bezeichnet, dann wird der Punkt  $x$  gemäß  $S_p(x)$  an Position  $(i, j)$  im diskreten SI abgebildet:

$$(8.2) \quad i = \left\lfloor \frac{\frac{W}{2} - \beta}{b} \right\rfloor, \quad j = \left\lfloor \frac{\alpha}{b} \right\rfloor$$

An dieser Position wird dann die Anzahl der Punkte, anteilig zur Gesamtzahl projizierter Punkte, vermerkt.

Zum Vergleich der SI wird standardmäßig der in Bildvergleichen häufig verwendete lineare Korrelationskoeffizient [Joh97] eingesetzt. Seien  $S_p$  und  $S_q$  zwei diskrete SI, mit jeweils  $n = i \cdot j$  Einträgen, dann ist dieser definiert als:

$$(8.3) \quad R(S_p, S_q) = \frac{\sum_{i,j}^n S_p(i,j)S_q(i,j) - \sum_{i,j}^n S_p(i,j) \sum_{i,j}^n S_q(i,j)}{\sqrt{\left( \sum_{i,j}^n S_p(i,j)^2 - \left( \sum_{i,j}^n S_p(i,j) \right)^2 \right) \left( \sum_{i,j}^n S_q(i,j)^2 - \left( \sum_{i,j}^n S_q(i,j) \right)^2 \right)}}$$

Er liefert Werte im Intervall  $[-1, 1]$ . Dabei entspricht 1 einer großen,  $-1$  keiner Ähnlichkeit.

## 8.2. Erweiterungen

Dieser Abschnitt behandelt Probleme, die bei der Erzeugung und dem Vergleich von SI auftreten können. Parallel dazu wird auch eine Lösung präsentiert. Eine ausführliche Betrachtung findet dabei nicht statt.

Kleine Änderungen der Punktdaten (Rauschen) haben einen direkten Einfluss auf das resultierende SI [Joh97] eines Punktes. Dadurch können für korrespondierende Punkte unterschiedliche SI entstehen. Um dies zu umgehen wird ein Punkt  $x$  nur dann auf das SI abgebildet, wenn dessen Normale  $n_x$  zu der von  $p$  ( $n_p$ ) in folgender Weise kompatibel ist:

$$(8.4) \quad \cos^{-1}(n_p \cdot n_x) < A_p$$

$A_p$  wird dabei als Support-Winkel bezeichnet.

Verursacht durch das beschriebene Kompatibilitätskriterium kann ein Spin Image an einer Positionen  $(i, j)$  fälschlicherweise keine Einträge enthalten. Zur Lösung dieses Problems wird zwischen jeweils vier benachbarten SI Positionen bilinear interpoliert [FSMo8, Joh97]. Die Gewichtungen werden dabei wie folgt berechnet:

$$(8.5) \quad w_a = \alpha - ib, \quad w_b = \beta + \frac{W}{2} - jb$$

Die resultierenden Einträge eines SI  $S_p$  an Position  $(i, j)$  sind dann:

$$(8.6) \quad \begin{aligned} S_p(i, j+1) &= S_p(i, j+1) + a(b - w_b) \\ S_p(i+1, j) &= S_p(i+1, j) + w_b(b - w_a) \\ S_p(i+1, j+1) &= S_p(i+1, j+1) + (w_a w_b) \\ S_p(i, j) &= S_p(i, j) + (b - w_a)(b - w_b) \end{aligned}$$

Dadurch entstehen weichere Verläufe innerhalb des SI, die den Prozess der Korrespondenzfindung verbessern. Allerdings werden dadurch auch feine Details ausgeblendet.

Auf Grund nicht vollständig überlappender Punktwolken weisen korrespondierende Punkte weiterhin unterschiedliche SI auf. Meist enthalten diese sogar eine unterschiedliche Anzahl abgebildeter Punkte. Da der Korrelationskoeffizient darauf sehr sensibel reagiert, wird in [Joh97] ein weiteres Ähnlichkeitskriterium eingeführt. Dieses ist eine Mischung aus Korrelationskoeffizient und Normalverteilung. Es berücksichtigt eine unterschiedliche Verteilung von Punkten. Definiert ist dieses wie folgt:

$$(8.7) \quad C(S_p, S_q) = (\arctan(R(S_p, S_q)))^2 - \lambda \frac{1}{n-3}$$

Dabei sind  $S_p$  und  $S_q$  SI der Punkte  $p$  und  $q$ ,  $n$  bezeichnet die Größe der Spin -Images.  $R$  entspricht dem Korrelationskoeffizienten und  $\lambda$  gewichtet den Einfluss von  $R$ . Das erlaubt ein Erkennen korrespondierender Punkte mit leicht unterschiedlichen SI.

### 8.3. Analysierte Variante

Zur Analyse wurde eine Variante eingesetzt, die aus Modulen der Point Cloud Library [[Lib12a](#)] zusammengesetzt wurde.

Für jeden Punkt beider Punktwolken wird die Normale bestimmt [[Lib12h](#)]. Das erfolgt mit Hilfe der PCA (siehe Kapitel B). Der Einzugsbereich wird dabei über den Radius einer Kugelumgebung definiert. Anschließend wird für jeden Punkt ein SI, wie in Abschnitt 8.1 vorgestellt, bestimmt. Dabei wird ein Support-Winkel verwendet, um Ausreißer zu eliminieren. Außerdem wird eine bilineare Interpolation (siehe Abschnitt 8.2) durchgeführt. Die Größe des Spin-Images ist auf  $8 \times 8$  festgelegt.

Um die Korrespondenzsuche in einem zeitlichen Rahmen zu halten und möglichst aussagekräftige SI zu extrahieren, wird auch hier der in Abschnitt 7.2.1 eingeführte Persistenz-Test durchgeführt. Möglich wird dies durch die Auffassung eines SI als Vektor der Länge  $i \cdot j$ . Als Vergleichsmetrik wird ebenfalls die Kullback-Leibler-Divergenz verwendet.

Zur Bestimmung der Korrespondenzen wird wie bei der analysierten Variante der PFH die Merkmalsdistanz verwendet. Das erfolgt ebenfalls unter der Betrachtung eines SI als Vektor. Die Suche nach Korrespondenzen wird dadurch auf die des nächsten Nachbarn reduziert. Mit Hilfe eines KD-Tree [[Lib12i](#)] (siehe Abschnitt 2.3.3) ist diese Suche effizient möglich.

Die so erhaltenen Korrespondenzen werden analog zu der in Abschnitt 7.3 vorgestellten Lösung auf Eindeutigkeit geprüft. Ein Verwerfung von Korrespondenzen bezüglich eines spezifizierten Grenzwertes erfolgt ebenfalls über die Merkmalsdistanz, welche durch die Auffassung eines SI als Vektor, im KD-Tree effizient ermittelt werden kann.

Abschließend wird aus den Punktkoordinaten der korrespondierenden SI mit Hilfe der Singulärwertzerlegung (siehe Anhang A) eine Transformation bestimmt [[Lib12d](#)].

Die Korrektheit der externen Implementierungen wird vorausgesetzt.

## 9. Iterative Closest Point (ICP)

DER 1992 von Besl und McKay [BM92] vorgestellte Iterative Closest Point (ICP) Ansatz ist einer der berühmtesten Algorithmen im Bereich der 3D-Registrierung (Klassischer ICP). Er gehört, wie in Abschnitt 3.2.2 erläutert, zur Klasse der Feinregistrierung. Durch seine Fähigkeit beliebige geometrische Daten (Punkte, Liniensegmente, Kurven, Oberflächen (Meshes)) verarbeiten zu können, hat er Einzug in nahezu jeden Anwendungsbereich der Registrierung von 3D-Daten erhalten. Besonders etabliert hat er sich in der Robotik. Aufgaben wie die Selbstlokalisierung können mit Hilfe odometrischer Daten und Objekterkennung in Echtzeit bewältigt werden [RL01]. Die dabei verarbeiteten Datenmengen beschränken sich jedoch meist auf 1000 – 10.000 Datenpunkte.

Dieses Kapitel stellt das Funktionsprinzip des ICP-Algorithmus vor. Anschließend wird ein Einblick in vorhandene Varianten sowie der in dieser Diplomarbeit untersuchten Variante (siehe Kapitel 11) gegeben.

### 9.1. Funktionsprinzip

Der ICP-Algorithmus versucht in einem iterativen Prozess eine Datenpunktfolke ( $Q$ ) zu einer Zielpunktfolke ( $P$ ) auszurichten. Dabei wird vorausgesetzt, dass  $Q$  durch eine initiale Transformation bereits grob an  $P$  registriert wurde. Für jeden Punkt  $p_i \in P$ , ( $1 \leq i \leq |P|$ ) wird dabei der nächste Nachbar  $nn_{p_i} \in Q$  ermittelt. Seien  $P$  und  $Q$  wie in Gleichung (3.2) definiert, dann lässt sich dieser beschreiben als:

$$(9.1) \quad \begin{aligned} nn_{p_i} &= q_j \in Q \quad \wedge \quad \forall q_l \in Q : \|p_i - q_j\|_2 \leq \|p_i - q_l\|_2, \quad j, l = 1 \dots |Q| \\ \|p - q\|_2 &= \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2}, \quad p \in P, q \in Q \end{aligned}$$

Diese bilden Korrespondenzen, aus denen jene Transformation berechnet wird, die alle Korrespondenzpunkte bestmöglich zueinander ausrichtet (siehe Anhang A). Ein Anwenden dieser Transformation auf  $Q$  entspricht dann einer Minimierung der quadratischen Distanzen zwischen den Korrespondenzpunkten. Betrachtet man die Menge aller Korrespondenzen, lässt sich dies als Summe der quadratischen Distanzen ausdrücken. Sei durch  $R$  und  $T$  eine Transformation wie in Gleichung (3.3) definiert, dann lässt sich diese wie folgt beschreiben:

$$(9.2) \quad E(R, T) = \sum_{i=1}^{|C|} \sum_{j=1}^{|C|} w_{ij} \|p_i - (R \cdot q_j - T + q_j)\|^2$$

Dabei bezeichnet  $C = \{(p_i, q_j) | p_i \in P, q_j \in Q, 1 \leq i \leq N_p, 1 \leq j \leq N_q\}$  die Korrespondenz-Menge (siehe Abschnitt 3.1). Die Gewichtung  $w_{ij}$  ist wie folgt belegt:  $w_{ij} = 1$  wenn  $(p_i, q_j) \in K$ , ansonsten  $w_{ij} = 0$ .

Im Sinne des ICP-Algorithmus wird  $E(R, T)$  als Fehler betrachtet und beschreibt, wie weit zwei Punktwolken von ihrer korrekten Ausrichtung entfernt sind. Über mehrere Iterationen des beschriebenen Prozesses hinweg, wird dieser Fehler minimiert. Dadurch definiert der ICP-Algorithmus das Registrierungsproblem als eine Mischung aus Korrespondenz- und Optimierungsproblem.

## 9.2. Varianten

Der vielseitige Einsatzbereich des ICP hat zu einer ungeheuren Vielfalt an Varianten geführt. In [RL01] wurde eine Kategorisierung der vorhandenen Varianten über mögliche Modifikationen einzelner Arbeitsschritte des ICP vorgenommen. Dabei wurden die sechs Teilschritte Punktwahl, Korrespondenzfindung, Korrespondenzgewichtung, Korrespondenzverwerfung, Fehlermaß und Minimierung betrachtet.

Neben der Kategorisierung verschiedener Varianten, wurden diese auch genauer untersucht und miteinander verglichen. Basierend auf den von Rusinkiewicz et al. [RL01] durchgeführten Untersuchungen soll im Folgenden ein Überblick der einzelnen Arbeitsschritte, existierenden Varianten und deren Verhalten gegeben werden.

### 9.2.1. Punktwahl

Bei der Punktwahl wird grundlegend in der gesamten oder nur teilweisen Verwendung der involvierten Punktwolken unterschieden. Ziel davon soll eine Reduktion der Datenmenge sein.

Die vorgestellten Algorithmen erzeugen Teilmengen dabei mittels uniformem, random und einem auf besonderen Intensitäts- und Farbwerten basierten Sub-Sampling. Rusinkiewicz et al. selbst führen in ihrem Vergleich eine neue Sub-Sampling-Strategie ein, bei der nur die Punkte einer Punktwolke verwendet werden, deren Normalen sehr unterschiedlich sind (Normal-Space-Sampling). Ziel davon ist es, aussagekräftige Repräsentanten der Punktwolken zu erhalten. Generell können die Teilmengen von beiden oder nur einer Punktwolke sowie in jeder oder nur der ersten Iteration (Random-Sampling) erzeugt werden.

Auf das Konvergenzverhalten und die Qualität der Ergebnisse haben verschiedene Sub-Sampling-Strategien bei rauschfreien Daten keinen Einfluss. Ein beidseitiges Sub-Sampling erzielt geringfügig höhere Genauigkeiten als das einseitige. Konkrete Laufzeiten werden an dieser Stelle nicht genannt. Als Abschätzung kann dafür das Konvergenzverhalten im Falle einer einmaligen Sub-Sampling-Strategie herangezogen werden.

### 9.2.2. Korrespondenzfindung

Die Suche von Korrespondenzen und somit des nächsten Nachbarn ist der zeitaufwändigste Arbeitsschritt des ICP. In [NLH07, FBF77, EMSS96] werden KD-Trees (siehe Abschnitt 2.3.3) und Cached-KD-Trees beschrieben. Diese erlauben eine effiziente Suche des nächsten Nachbarn in  $\mathcal{O}(\log(n))$  anstelle von  $\mathcal{O}(n)$  (siehe Abschnitt 2.3.3). Wird eine vektorisierbare Repräsentation von korrespondierenden Elementen verwendet, kann diese Suchvariante mit jeder beliebigen, im Folgenden betrachteten Modifikation kombiniert werden. Unterschieden werden diese in der Art der Korrespondenzen und der Auffindung dieser verwendeten Technik.

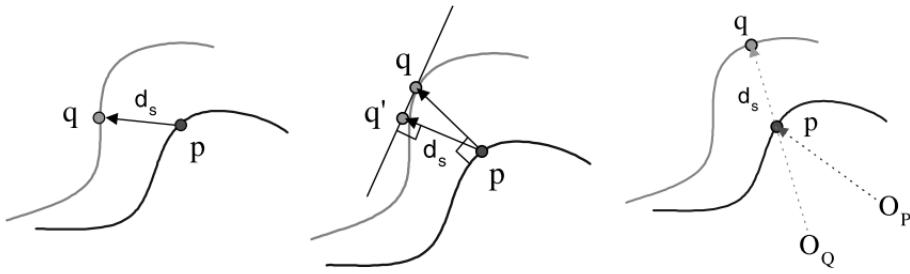
Während der klassische ICP Punktkorrespondenzen (Point-to-Point) (siehe Abbildung 9.1) basierend auf Positionsdaten ermittelt, verwenden Varianten [JK97] zusätzlich Farbinformationen der Punkte. Ergebnissen zufolge kann dadurch die Genauigkeit und Anzahl benötigter Iterationen reduziert werden. Als Normal-Shooting (Point-to-Plane) (siehe Abbildung 9.1) bezeichneten Rusinkiewicz et al. eine Variante, die der Normalen eines Datenpunktes (Punkt aus Datenpunktewolke) zur Oberfläche der Zielpunktewolke folgt und den dort getroffenen Punkt als korrespondierend betrachtet. Ein Ausbilden falscher Korrespondenzen aufgrund von Ausreißern oder versetzten Oberflächenverläufen die sich als geringste Punkt-zu-Punkt-Distanz anbieten würden, soll dadurch verhindert werden Reverse-Calibration projiziert (Point-to-Projektion) (siehe Abbildung 9.1) einen Zielpunkt über den Ursprung der Datenpunktewolke auf selbige. Dadurch ist keine Suche des nächsten Nachbarn notwendig. Für all diese Ansätze kann die Bestimmung des nächsten Nachbarn durch Informationen wie Normalen, Farbe oder Intensität eines Punktes optimiert werden. Diese Informationen werden auch Kompatibilitäten genannt. ICP-Varianten, die diese Informationen nutzen, tragen daher auch den Namen „Iterative Compatible Point“.

Rusinkiewicz et al. kamen zu dem Ergebnis, dass der Point-to-Point Ansatz kombiniert mit einer Normalenkompatibilität (Punkte mit Winkeldifferenz der Normalen  $< 45^\circ$ ), das qualitativ beste Ergebnis bei bestem Konvergenzverhalten erzielt. Der Zeitaufwand ist dabei deutlich geringer als bei Point-to-Plane mit Normalenkompatibilität. Sowohl Qualität als auch Konvergenzverhalten sind dabei jedoch besser als beim Point-to-Projektion Ansatz. Dieser schneidet diesbezüglich am schlechtesten ab, ist aber auch am schnellsten. In den Untersuchungen wurden rauschfreie Daten verwendet. Die Sensibilität der Normalen auf Rauschen führt im Allgemeinen zu schlechten Ergebnissen. Generell kann aber gesagt werden, dass diese Modifikationen das Bilden schlechter Korrespondenzen unterbinden.

### 9.2.3. Korrespondenzgewichtung

Mit Hilfe einer Gewichtung soll in diesem Schritt der Einfluss einzelner, als besonders gut erachteter Korrespondenzen auf das Fehlermaß verstärkt werden.

Als Variationen werden die konstante (alle Korrespondenzen gleich gewichtet), die entfernungsabhängige (je kleiner die Distanz, desto höher die Gewichtung), die kompatibilitätsabhängige (je ähnlicher die Normalenrichtungen oder Farbwerte desto höher die



**Abbildung 9.1.: Korrespondenz-Findung** - *Links: Point-to-Point Korrespondenz; der nächste Nachbar (minimales Distanz  $d_s$ ) zu  $p$  ist  $q$ . Mitte: Point-to-Plane Korrespondenz;  $q$  ist Schnittpunkt des Normalen-Strahles von  $p$  und korrespondiert.  $q'$  entspricht der Projektion von  $p$  auf die Tangentialebene deren Berührpunkt  $q$  ist. Rechts: Punkt  $p$  wird vom Ursprung der Datenpunktewolke aus ( $O_Q$ ) auf die Zielpunktewolke vorwärts-projiziert. Mittels Ursprung der Zielpunktewolke ( $O_P$ ) wird  $p$  vor der Projektion ermittelt. [PS03]*

Gewichtung) und eine auf dem Rauschverhalten des Laserscanner basierte Gewichtung vorgestellt. Bei Letzterer wird jedem Punkt, basierend auf einem Rauschmodell des verwendeten Entfernungssensors, eine Wahrscheinlichkeit zugeordnet. Diese beschreibt, ob und wie stark ein Punkt fehlerbehaftet ist (ein hohe Wahrscheinlichkeit entspricht einer geringer Gewichtung).

Die durchgeföhrten Untersuchungen haben gezeigt, dass die entfernungsabhängige und kompatibilitätsabhängige Gewichtung prinzipiell die Anzahl der Iterationen minimieren können, dies aber stark datenabhängig (Rauschen etc.) ist. Außerdem entspricht die entfernungsabhängige Gewichtung der im folgenden Abschnitt vorgestellten Grenzwertverwerfung. Abschließend wird festgestellt, dass der generelle Einfluss von Gewichtungen auf das Endergebnis gering und somit zu vernachlässigen ist.

#### 9.2.4. Korrespondenzverwerfung

In diesem Schritt wird versucht einem definierten Qualitätsmaß entsprechend schlechte Korrespondenzen zu erkennen und zu eliminieren. Dies soll Qualität und Korrespondenzverhalten des Algorithmus verbessern.

Eine vorgestellte Modifikation benutzt dazu die Entfernung der Korrespondenzpunkte. Dabei werden alle Korrespondenzen verworfen, deren Punkt-zu-Punkt-Distanz einen spezifizierten Grenzwert überschreitet.

Ein etwas variabler Grenzwert wird von einer anderen Variante verwendet. Dabei werden die  $n\%$  schlechtesten Korrespondenzen, bezüglich ihrer Punkt-zu-Punkt-Distanz verworfen. Ein weiterer Ansatz vergleicht die Punkt-zu-Punkt-Distanz benachbarter Korrespondenzen und verwirft Ausreißer.

Als ähnliche Variante wurde auch das Verwerfen von Korrespondenzen, deren Punkt-zu-Punkt-Distanz größer als das Vielfache der Standardabweichung aller Korrespondenzen ist, vorgestellt. Einem generellen Problem folgend entfernt die letzte betrachtete Modifikation

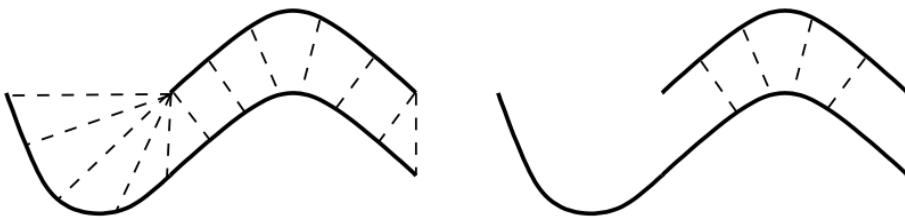


Abbildung 9.2.: Korrespondenzverwerfung - *Links:* Randbereiche der Oberflächen führen häufig zur Bildung falscher Korrespondenzen. *Rechts:* Die Korrespondenzen an Randbereichen wurden verworfen.

Korrespondenzen, deren Punkte auf dem Rand einer Oberfläche liegen (siehe Abbildung 9.2).

Rusinkiewicz et al. kamen zu dem Schluss, dass Stabilität und Genauigkeit durch das Entfernen von ausreißenden Korrespondenzen verbessert werden kann. Die Konvergenzgeschwindigkeit jedoch bleibt davon nahezu unbeeinflusst.

### 9.2.5. Fehlermaß und Minimierung

In diesem Abschnitt werden zwei Arbeitsschritte auf Grund ihres engen Zusammenhangs gemeinsam betrachtet. Dabei kann das Fehlermaß verändert und abhängig davon auf verschiedene Art und Weise minimiert werden. Der Minimierungsschritt beinhaltet dabei auch die Berechnung der Transformation.

Der klassische ICP verwendet als Fehlermaß die Summe quadratischer Distanzen der gefundenen Korrespondenzen. Minimiert werden kann dieses Fehlermaß mittels Singulärwertzerlegung (siehe Anhang A), orthonormalen Matrizen, Quaternionen und Dual-Zahl Quaternionen. Diese wurden in [ELF97] genauer auf Präzision und Stabilität hin mit dem Ergebnis untersucht, dass ihre Unterschiede nur sehr gering sind.

Ein weitere von Rusinkiewicz et al. erwähnte Variante erweitert die Metrik der Summe quadratischer Fehler auf zusätzliche Farbinformationen.

Der in Abschnitt 9.2.2 vorgestellte Point-to-Plane Ansatz wurde mit einer eigenen Metrik eingeführt. Diese beschreibt die Summe der quadratischen Abstände eines jeden Zielpunktes, zu der Tangentialebene, deren Berührpunkt dem korrespondierenden Datenpunkt entspricht. Dieses Fehlermaß erfordert, auf Grund seiner nicht-linearen Form, ein Lösungsverfahren, wie z.B. das Levenberg-Marquardt-, das erweiterte Gauß-Seidel- oder das Newton-Raphson-Verfahren.

Betrachtete man die Minimierung über mehrere Iterationen hinweg, können verschiedene Strategien zum Minimieren des gesamten Fehlers verwendet werden. Ein recht geradliniges Verfahren bestimmt dabei in jeder Iteration eine Menge von Korrespondenzen und eine Transformation, die ihren Abstand zueinander minimiert.

Der klassische ICP verwendet neben diesem eine zusätzliche Extrapolation im Transformationsraum. Über den Transformations- und Fehlerverlauf bisheriger Iterationen wird dabei

approximiert und extrapoliert. Ausreißende Transformationen werden dann geringfügig korrigiert, wodurch das Konvergenzverhalten verbessert wird.

Eine Brute-Force ähnliche Methode wurde ebenfalls untersucht. In dieser wird der ICP mit verschiedenen initialen Transformationen prozessiert. Als endgültige Transformation wird die gewählt, die den geringsten Fehler verursacht. Auf diese Art und Weise können lokale Minima der Fehlerfunktion übersprungen werden. Dem selben Ziel folgend wurde ein Simulated-Annealing Ansatz vorgestellt. Er wurde jedoch auf Grund seines hohen Rechenaufwandes nicht weiter untersucht.

Der zuletzt betrachtete Ansatz verwendet in jeder Iteration eine zufällige Teilmenge von Korrespondenzen. Minimiert wird mit den Korrespondenzen, deren Fehler am geringsten ist. Der Ansatz erfordert einen hohen Zeitaufwand und enthält Zufallselemente, dessen Ausgang schwer zu bestimmen ist.

### 9.3. Analysierte Variante

Die zur Analyse verwendete Variante soll aussagekräftige Ergebnisse zu den in Abschnitt 1.2 genannten Kriterien (Konvergenz, Effizienz, Stabilität und Reproduzierbarkeit) liefern. Basierend auf den Ergebnissen von [RL01] wurde daher entschieden, die in der Point Cloud Library vorhandene Variante des ICP [Lib12f] in leicht veränderter Form zu verwenden.

Gemäß der im vorherigen Abschnitt beschriebenen Kategorisierung ist sie wie folgt zusammengestellt:

- **Punktwahl** - In jeder Iteration werden die gesamten Punktwolken verwendet. Eine Reduktion der Daten kann im Vorfeld mit Hilfe der in Kapitel 6 vorgestellten Verfahren gewährleistet werden.
- **Korrespondenzfindung** - Als Korrespondenzen werden Punkt-zu-Punkt Korrespondenzen verwendet. Farb- oder Normaleninformationen werden auf Grund des zusätzlichen Rechenaufwandes nicht betrachtet. Die Suche des nächsten Nachbarn wird mittels KD-Tree (siehe Abschnitt 2.3.3) durchgeführt.
- **Korrespondenzgewichtung** - Alle Korrespondenzen werden gleich gewichtet (konstante Gewichtung), da der Einfluss von Gewichtungen sehr gering ist und teilweise über die Korrespondenzverwerfung gesteuert werden kann.
- **Korrespondenzverwerfung** - Zur Verwerfung schlechter Korrespondenzen wird die entfernungsabhängige Verwerfung verwendet. Dabei werden korrespondierenden Punkten, deren Distanz zueinander einen spezifizierten Grenzwert übersteigt, verworfen. Dieser wird im folgenden als Korrespondenzgrenzwert bezeichnet. An dieser Stelle wurde die in der Point Cloud Library vorhandene Variante modifiziert. Diese verwendet zusätzlich die Verwerfung ausreißender Korrespondenzen, basierend auf einem RANSAC-Ansatz. Aufgrund des dadurch eingeführten Zufallselementes und der damit verbundenen Verletzung der Reproduzierbarkeit wurde sie entfernt.

- **Fehlermaß und Minimierung** - Basierend auf der Wahl der Korrespondenzfindung, wird die Summe der quadratischen Distanzen (siehe Gleichung (9.2)) verwendet. Mittels Singulärwertzerlegung wird minimiert (siehe Anhang A).

Das Verfahren konvergiert, wenn die folgenden drei Bedingungen (Abbruchbedingungen) erfüllt sind:

1. **Transformationsgrenzwert** - Die Differenz der berechneten Transformationen zwischen aktueller und vorheriger Iteration ist geringer, als ein spezifizierter Grenzwert.
2. **Fehlergrenzwert** - Der berechnete Fehler (Summe der quadratischen Distanzen) ist geringer als ein spezifizierter Grenzwert.
3. **Iterationen** - Eine spezifizierte Anzahl von Iterationen wurde ausgeführt.



# 10. Normal Distribution Transformation (NDT)

Der Normal Distribution Transformation Ansatz (NDT) wurde erstmals von Biber und Strasser [BS03] im Jahre 2003 vorgestellt. Ursprünglich war dieser für zweidimensionale Punktdaten konzipiert (2D-NDT). Ripperda [Rip05] konzipierte 2005 eine aus 2D-Schnitten beschriebene 3D-Variante. 2007 erweiterte Magnusson [MLD07] diesen für den Einsatz in autonomen Minenfahrzeugen vollständig auf den dreidimensionalen Raum (3D-NDT). In seiner zwei Jahre später erschienenen Dissertation [Mag09], wurde dieser genauer untersucht. Durch die Fähigkeit initiale Transformationen verarbeiten zu können, gliedert er sich in die Klasse der feinregistrierenden Algorithmen ein. Sein Haupteinsatzgebiet wird dabei in der Navigation autonomer Roboter und dem Erstellen von Umgebungskarten aus dichten 3D-Punktwolken gesehen.

Im Laufe dieses Kapitels wird die Funktionsweise des 3D-NDT vorgestellt. Außerdem wird ein Überblick möglicher Erweiterungen sowie der in Kapitel 11 untersuchten Variante gegeben.

## 10.1. Funktionsprinzip

Dreh- und Angelpunkt des NDT liegt in seiner Repräsentation der Zielpunktwolke  $P$  (siehe Gleichung (3.2)). Diese wird wie beim räumlichen Sub-Sampling (siehe Abschnitt 6.2) in ein Gitter zerlegt, das aus uniformen Zellen besteht. Für jede Zelle  $B$  wird dann der Schwerpunkt  $q$  und die Kovarianzmatrix  $C$  der darin enthaltenen Punkte  $b_i$  gebildet.

$$(10.1) \quad \begin{aligned} q &= \frac{1}{|B|} \sum_{i=1}^{|B|} b_i, \quad B \subseteq P \\ C &= \frac{1}{|B|} \sum_{i=1}^{|B|} (b_i - q)(b_i - q)^T \end{aligned}$$

Diese modellieren die Normalverteilung  $N(q, C)$  der in Zelle  $B$  vorhandenen Punkte. Um eine aussagekräftige Kovarianzmatrix zu erhalten sind dazu mindestens fünf Punkte notwendig [MLD07]. Weiter kann darauf basierend eine Wahrscheinlichkeitsdichtefunktion (PDF)  $p(x)$  definiert werden:

$$(10.2) \quad p(x) = \frac{1}{c} \exp \left( -\frac{(x - q)^T C^{-1} (x - q)}{2} \right), \quad x \in \mathbb{R}^3$$

Diese beschreibt die Wahrscheinlichkeit, dass in Zelle  $B$  ein Punkt an Position  $x$  vorkommt. Die Konstante  $c$  dient zur Normalisierung und kann in diesem Fall als 1 festgelegt werden [MLD07].

Sei  $T(e, q_i)$  eine Transformationsfunktion, die einen Punkt  $q_i$  aus der Daten-Punktwolke  $Q$  (die an  $P$  registriert werden soll) gemäß einer Pose  $e$  im  $\mathbb{R}^3$  transformiert. Die Pose  $e$  ist dabei durch eine Translation und Rotation, wie in Gleichung (3.3) gegeben. Mit eulerschen Winkeln ist diese dann beschrieben als:

$$(10.3) \quad e = (\psi, \theta, \phi, t_x, t_y, t_z), \quad 0 \leq \psi, \theta, \phi \leq 360^\circ, \quad t_x, t_y, t_z \in \mathbb{R}^3$$

Eine Bewertungsfunktion  $s(e)$ , die eine Güte der Ausrichtung von  $Q$  zu  $P$  liefert, wird dann definiert als:

$$(10.4) \quad s(e) = - \sum_{i=1}^{|Q|} p(T(e, q_i))$$

Dies entspricht der negierten Summe der Wahrscheinlichkeiten, mit denen Punkte aus  $Q$ , unter der Pose  $e$ , zu denen in  $P$  korrelieren.

Da die Fehlerfunktion über eine PDF definiert ist, existieren für diese stetige Ableitungen erster und zweiter Ordnung. Diese erlauben, das Registrierungsproblem als nichtlineares Optimierungsproblem der Form

$$(10.5) \quad H\Delta e = -g$$

mit Hessematrix  $H$  und Gradienten  $g$  von  $s(e)$  zu definieren. Gelöst werden kann es mit Hilfe numerischer Optimierungsverfahren. Wird dabei das iterative Newton-Verfahren verwendet, wird die Transformationsänderung  $\Delta e$  in jeder Iteration zu der bisherigen hinzugefügt. Für Punkte aus  $Q$  gilt dann in jeder Iteration:

$$(10.6) \quad q'_i = T(e + \Delta e, q_i) - q$$

## 10.2. Varianten

Einige Bereiche des NDT lassen sich in Hinblick auf Qualität, Laufzeit und Einsatzbereich optimieren. Ein Überblick bisheriger Modifikationen wird im folgenden gegeben. Der Anspruch auf Vollständigkeit wird und soll dabei nicht erhoben werden.

Wie bereits erwähnt stellten Ripperda und Brenner [Rip05] ebenfalls eine 3D-Variante des NDT vor. Dabei wird eine 3D-Punktwolke in 2D-Scheiben zerlegt. Auf diesen kann dann der von Biber und Strasser [BS03] vorgestellte Algorithmus ausgeführt werden. Nachteil dieses Verfahrens ist, dass nur auf einer Ebene (Scheibe) registriert werden kann. Dies bedeutet, dass die Punktwolken auf der selben Höhe erzeugt worden sein müssten, was selten der Fall ist.

In [KTMAo8] wurde eine Erweiterung vorgestellt, die ebenfalls auf dem 2D-NDT basiert. Bei dieser wird die Kovarianzmatrix einer Zelle in jeder Iteration leicht abgeschwächt. Dadurch entsteht eine „verblasste“ Repräsentation der Punktwolke. Kaminade et al. erzielten dadurch auf zweidimensionalen Daten eine höhere Genauigkeit als mit der von Biber und Strasser [BS03] präsentierten Variante.

Magnusson [Mag09] beschreibt mehrere Erweiterungen seines eingeführten 3D-NDT. Diese behandeln im Allgemeinen die Wahl der Zellgrößen, was als Art der Diskretisierung einer Punktwolke interpretiert werden kann. Dabei wird eine auf einem Octree (wie KD-Tree (siehe Abschnitt 2.3.3) aber mit acht statt zwei Kindern je Knoten) basierende Methode beschrieben. Diese verfeinert durch Zerlegung in kleinere Zellen Bereiche einer Punktwolke, wenn die Varianz der Normalverteilung einen spezifizierten Grenzwert übersteigt. Ziel dabei ist es, markante Bereiche genauer zu untersuchen. Der Octree dient dabei zur Beschleunigung der Punkt-Zell-Vergleiche und Verwaltung der unterschiedlichen Zellhierarchien.

Als weitere adaptive Variante wird eine Segmentierung der Punktwolke vorgestellt. Jedes erkannte Segment wird dabei als eine Zelle repräsentiert.

Ein anderer Ansatz verringert die Zellgröße von Iteration zu Iteration. Dies hat den Effekt einer Grob-zu-Fein-Registrierung.

Die trilineare Interpolation ist eine 3D-Erweiterung der in [BS03] bereits vorgestellten Glättung der Normalverteilungsfunktion. Diese wird dabei für eine Zelle in Abhängigkeit der benachbarten acht berechnet. Dadurch entsteht ein achtfacher Berechnungsaufwand, wie Magnusson selbst berichtet.

Takeuchi [TTo6] präsentierte eine Modifikation, deren Besonderheit in der Wahl der Zellgrößen liegt. Dabei wird die Zellgröße mit der Entfernung zum Laserscanner erhöht. Dadurch werden entfernte rauschbehaftete Bereiche weniger detailliert repräsentiert als nahe. Die auf einer Indoor-Szene durchgeführten Untersuchungen erbrachten gute Ergebnisse.

### 10.3. Analysierte Variante

Die zur Analyse verwendete Variante entstammt der Point Cloud Library [Lib12b]. Diese entspricht der in [Mag09] vorgestellten Variante. Die Korrektheit der dortigen Implementierung wird vorausgesetzt.

Als PDF wird dabei eine Mischung aus Normalverteilung und uniformer Verteilung von Punktdaten verwendet. Diese ist definiert als [Mag09]:

$$(10.7) \quad p(x) = c_1 \exp \left( -\frac{(x - q)^T C^{-1} (x - q)}{2} \right) + c_2 p_0$$

Dabei beschreibt  $p_0$  die Rate erwarteter Ausreißer in den Punktdaten. Die Konstanten  $c_1$  und  $c_2$  werden so gewählt, dass die Summe aller Wahrscheinlichkeiten 1 ergibt. Dadurch kann

der nicht unerhebliche Einfluss von Ausreißern auf die Bewertungsfunktion limitiert werden. In der verwendeten Implementierung werden die Konstanten wie folgt belegt:

$$(10.8) \quad \begin{aligned} c_1 &= 10 \cdot (1 - p_0) \\ c_2 &= \frac{1}{(\text{Zellgröße})^3} \end{aligned}$$

Die so definierte PDF führt zu nicht trivialen ersten und zweiten Ableitungen der zu optimierenden Fehlerfunktion. Daher wird diese über eine Gaußsche Normalverteilung approximiert [Mago9]. Der Einfluss eines Punktes  $x$  auf die Bewertungsfunktion ist dann gegeben durch:

$$(10.9) \quad \begin{aligned} p(x) &= -d_1 \exp \left( -\frac{d_2}{2} (x - q)^T C^{-1} (x - q) \right) \\ d_1 &= -\log(c_1 - c_2) - \log(c_2) \\ d_2 &= -2 \log \left( \frac{(-\log(c_1 \exp(-\frac{1}{2}) + c_2) + \log(c_2))}{d_1} \right) \end{aligned}$$

Zur Repräsentation der Zellen wird ein Voxel-Grid, wie bereits in Abschnitt 6.2 erwähnt, mit konstanten, gleichseitigen Quadern verwendet. Zur Optimierung wird das iterative Newton-Verfahren unter Verwendung der Moré-Thuente Variante [MTMp92] eingesetzt.

Als Konvergenzkriterium werden die folgenden Bedingungen verwendet:

1. **Tranformationsgrenzwert** Die berechnete Transformationsänderung ist geringer als ein spezifizierter Grenzwert.
2. **Iterationen** Eine spezifizierte Anzahl an Iterationen wurde ausgeführt.

# 11. Analyse

**ZIEL** dieses Kapitels ist es, die Eignung punktbasierter Registrierungsalgorithmen auf hochauflösten Punktwolken zu klären. Analysiert werden dazu die bereits vorgestellten Varianten der Registrierungsalgorithmen PFH (siehe Abschnitt 7.3), SI (siehe Abschnitt 8.3), ICP (siehe Abschnitt 9.3) und NDT (siehe Abschnitt 10.3).

Verwendet wird der in Kapitel 5 beschriebene Aufbau. Dabei werden die drei vorgestellten Testszenarien 5.3 in den diskreten und räumlichen Auflösungen 1K, 10K, 100K, 1M, 10M (K=Tausend, M=Millonen) (siehe Kapitel 6) registriert.

Die folgenden vier Unterkapitel befassen sich mit den Analysen der jeweiligen Algorithmen. Dabei werden zuerst die einzelnen Parameter studiert. Auf Grund sehr langer Laufzeiten wird dabei eine Auflösung von 100K verwendet. Dies entspricht beim räumlichen Sub-Sampling einer Zellgröße von 0,06 m für das Indoor-Szenario bzw. 0,25 m für Hybrid- und Outdoor-Szenario. Anschließend werden die in 1.2 erwähnten Anforderungen der Konvergenz, Effizienz und Stabilität untersucht. Die Reproduzierbarkeit muss nicht weiter untersucht werden, da diese bereits durch die Wahl der Varianten gewährleistet ist.

## 11.1. Point Feature Histogram

Der PFH Algorithmus ist nicht in der Lage Vorabinformationen zu verarbeiten. Eine initiale Ausrichtung der Punktwolken ist daher durch das Platzieren der Punktwolken mit der Position des Laserscanners auf dem Ursprung des gemeinsamen Koordinatensystems (siehe Abschnitt 5.3) gegeben. Die initialen Fehler dazu sind in Anhang C einzusehen.

Die hier analysierte Variante des PFH Algorithmus kann im Wesentlichen über den Normalenradius, den Merkmalsradius, die Abweichung vom Durchschnittshistogramm und den Korrespondenzgrenzwert gesteuert werden. Mit Hilfe dieser Parameter sollen möglichst aussagekräftige FPFH aus Ziel- und Datenpunktfolke extrahiert werden. Davon wird sich eine Berechnung möglichst weniger aber dafür korrekter Korrespondenzen erhofft.

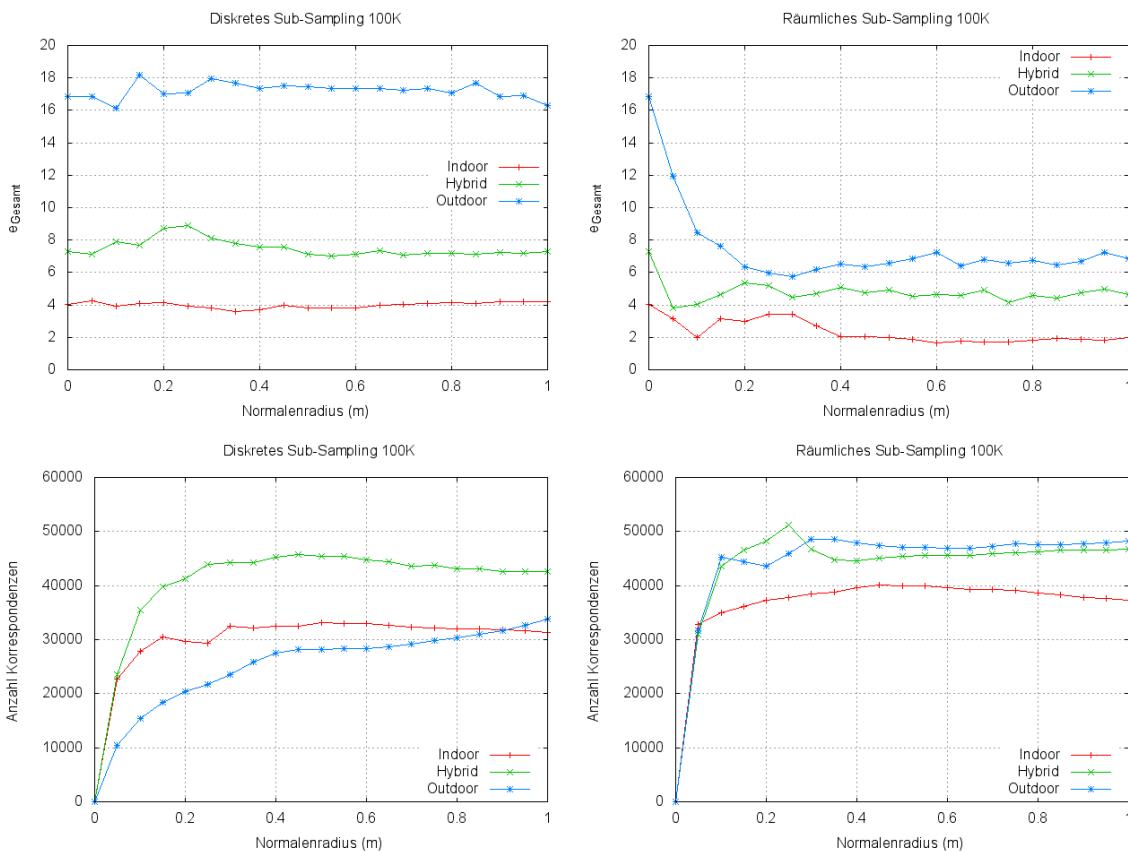
### 11.1.1. Normalenradius

Die Suche nach besonders aussagekräftigen FPFH kann auch als Suche nach möglichst unterschiedlichen FPFH betrachtet werden. Die Normale eines Punktes bildet hierfür die Grundlage und sollte daher so berechnet werden, dass die tatsächliche Orientierung eines

## 11. Analyse

Punktes möglichst genau repräsentiert wird. Ist dies nicht der Fall kann die anschließend grobe Klassifizierung im Histogramm zu ähnlichen oder gar gleichen FPFH führen.

Zur Untersuchung des Normalenradius wurde ein Merkmalsradius gewählt, der auch bei einer räumlichen Reduktion der Punktdaten ausreichend viele Nachbarn eines Punktes betrachtet. Eine Verwerfung der Merkmale durch Prüfen der Einzigartigkeit sowie eine Verwerfung von Korrespondenzen über einen Korrespondenzgrenzwert erfolgte dabei nicht. Lediglich die Eindeutigkeit der Korrespondenzen wurde abgefragt. Die resultierende Anzahl an Korrespondenzen gibt dadurch Aufschluss über die Ähnlichkeit der FPFH. Eine geringe Anzahl von Korrespondenzen bedeutet, dass viele FPFH der Zielpunktwolke zu einem FPFH der Datenpunktwolke (ebenso umgekehrt) korrespondieren, verworfen werden und ähnlich sind. Bei einer hohen Anzahl resultierender Korrespondenzen sind die FPFH entsprechend unterschiedlich.



**Abbildung 11.1.: FPFH Normalenradius** - Eine geringe Anzahl von Korrespondenzen indiziert ähnliche FPFH, eine sehr hohe entsprechend unterschiedliche. Bei räumlichem Sub-Sampling verbessert sich mit einer steigenden Anzahl unterschiedlicher FPFH das Registrierungsergebnis.

Basierend auf den Ergebnissen in Abbildung 11.1 kann gesagt werden, dass eine Wahl des Normalenradius größer als der Merkmalsradius keine Verbesserung des Registrierungsergebnisses erzielt. Eine zu kleine Wahl führt allerdings dazu, dass sehr ähnliche FPFH entstehen.

Besonders auffällig ist dies bei diskret reduzierten Punktwolken. Ursache davon ist vermutlich die sehr dichte Repräsentation der dem Laserscanner naheliegenden Bereiche. Diese zeichnen sich in Hybrid- und Outdoor-Szenario durch überwiegend flachen Untergrund, im Indoor-Szenario durch flache Wände, Decken und Böden aus. Punkte, die auf einer Fläche liegen, erhalten bei kleinen Radien eine nahezu identische Normale, woraus auch identische FPFH folgen. Bei größeren Radien fließen auch Bereiche außerhalb der Flächen in die Berechnung der Normalen ein, was unterschiedlichere Normalen und FPFH erzeugt. Die hohe Punktdichte führt dazu, dass benachbarte Punkte selbst bei größeren Radien sehr ähnliche Normalen aufweisen. Dies erklärt auch, dass auf räumlich reduzierten Punktdaten generell mehr Korrespondenzen gebildet werden.

Auf räumlich reduzierten Daten verbessert sich das Registrierungsergebnis bei sinkender Ähnlichkeit der FPFH. Die vollständige, aber weniger detaillierte Repräsentation einer Szene führt vermutlich dazu, dass sich für die in ihr enthaltenen Objekte ausgeprägtere Merkmale ausbilden lassen.

Für diskretes Sub-Sampling lässt sich kein Zusammenhang zwischen der Ähnlichkeit der FPFH und dem Registrierungsergebnis herstellen. Auch die Wahl des Normalenradius scheint bei dem hier verwendeten Merkmalsradius keinen wesentlichen Einfluss auf dieses zu nehmen. Die allgemein niedrige Anzahl von Korrespondenzen weist darauf hin, dass die dortigen FPFH unterschiedlich genug sind um eindeutige Korrespondenzen ausbilden zu können, aber nicht in einem Maß, welches eine korrekte Zuordnung der Korrespondenzpunkte erlaubt.

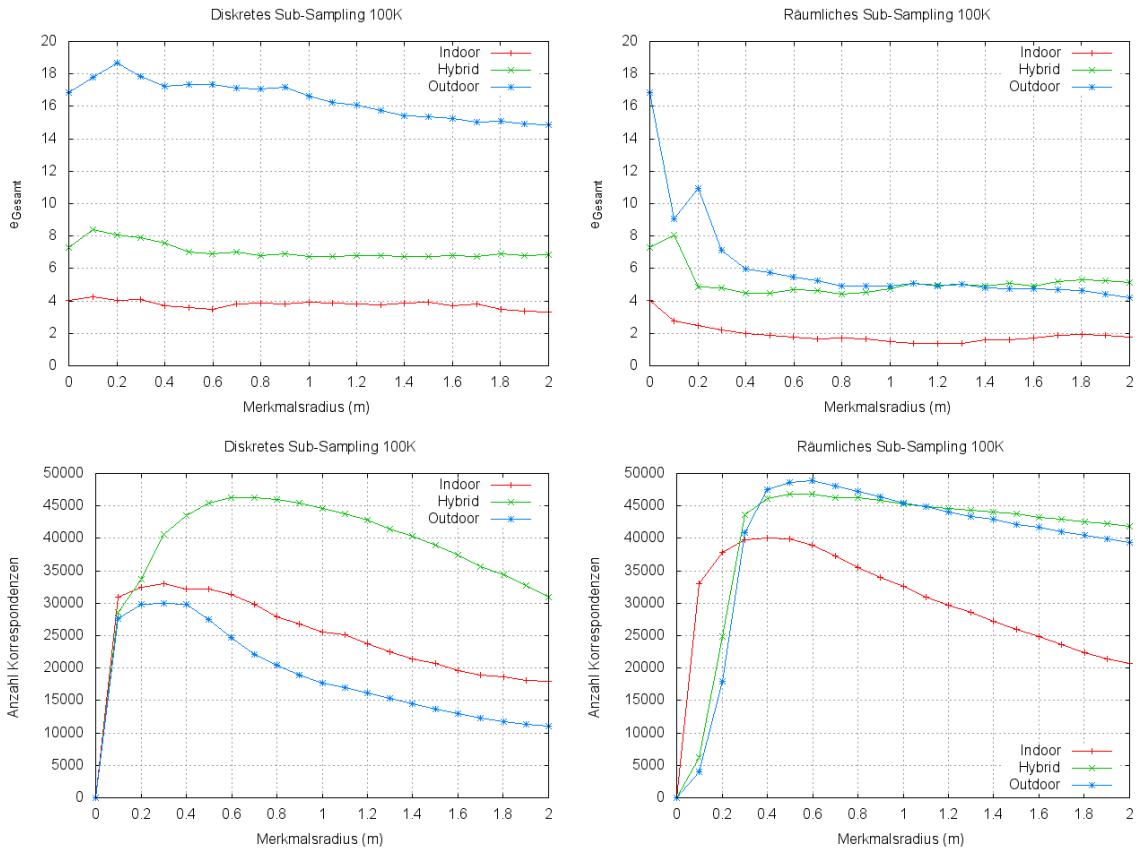
### 11.1.2. Merkmalsradius

Dieser Parameter bestimmt die auf ein FPFH Einfluss nehmende Umgebung eines Punktes. Bildlich gesprochen bedeutet dies, dass ein großer Wahl des Merkmalsradius zu einer Repräsentation größer Merkmale führt. Analog zum Normalenradius wurde auch bei den Untersuchungen zum Merkmalsradius keine Prüfung der Einzigartigkeit und Verwerfung von Korrespondenzen bezüglich eines Korrespondenzgrenzwertes vorgenommen. Durch die Eindeutigkeitsprüfung der Korrespondenzen ist deren Anzahl auch hier ein Indikator für die Ähnlichkeit der FPFH.

Die Ergebnisse in Abbildung 11.2 zeigen, dass sich bei einem steigenden Merkmalsradius auf diskret reduzierten Punktdaten eine nur unwesentliche Verbesserung des Registrierungsergebnisses einstellt. Am meisten profitiert das Outdoor-Szenario von sehr großen Radien. Die Verschiedenheit der FPFH steigt bei allen Szenarien mit dem Merkmalsradius bis zu einer wesentlichen Überschreitung des Normalenradius und sinkt anschließend wieder. Sehr viele Korrespondenzen werden dabei in der Hybrid-Szene berechnet.

Bei räumlich reduzierten Punktwolken sinkt der Registrierungsfehler mit steigendem Merkmalsradius bis der Normalenradius überschritten wird; danach verläuft er konstant. Die Anzahl der Korrespondenzen ist auf räumlich reduzierten Punktdaten deutlich höher und führt zu dem Schluss, dass mehr voneinander verschiedene FPFH ausgebildet werden konnten.

## 11. Analyse



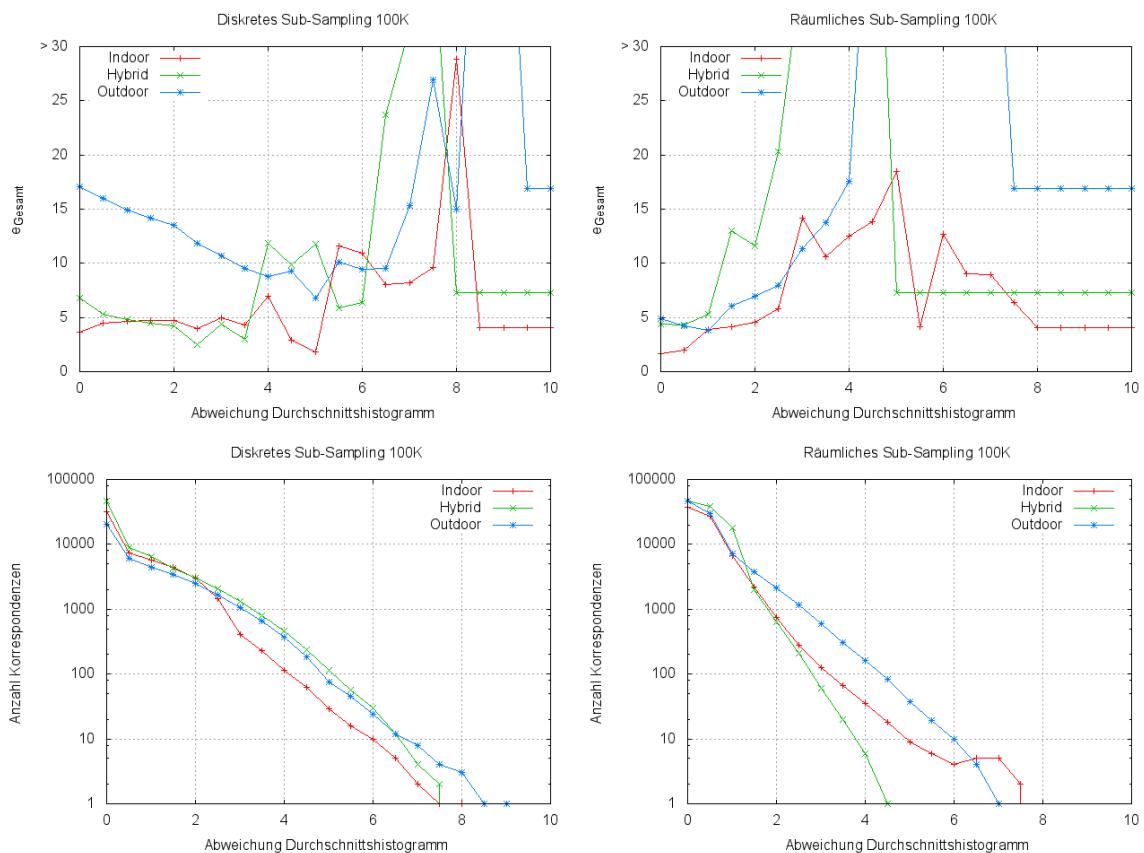
**Abbildung 11.2.: FPFH Merkmalsradius** - Eine geringe Anzahl von Korrespondenzen indiziert ähnliche FPFH, eine sehr hohe entsprechend unterschiedliche. Der Merkmalsradius sollte, um möglichst viele verschiedene FPFH zu erhalten, mindestens so groß wie der Normalenradius gewählt werden.

Der Einfluss des Merkmalsradius auf das Registrierungsergebnis bei einem diskreten Sub-Sampling lässt vermuten, dass größere als die hier untersuchten Merkmalsradien bessere Ergebnisse erzielen. Aus Effizienzgründen stellt dies jedoch keine Option dar und wird daher nicht weiter untersucht. Der hohe Einfluss sehr dicht repräsentierter Flächen auf die FPFH benachbarten Punkte scheint ein Problem darzustellen. Thematisiert wurde dies bereits für den Normalenradius (siehe Abschnitt 11.1.1) und lässt sich auf den Merkmalsradius übertragen. Indiz dafür ist die geringe Anzahl berechneter Korrespondenzen für das Indoor- und Outdoor-Szenario sowie die damit verbundene Ähnlichkeit der FPFH. Im Vergleich zur Hybrid-Szene enthalten sie sehr ausgedehnte und dem Laserscanner nahe Flächen. Ein räumliches Sub-Sampling wirkt diesem Problem entgegen.

### 11.1.3. Abweichung Durchschnittshistogramm

Dieser Parameter entscheidet über die Anzahl und Ähnlichkeit der zur Korrespondenzsuche zugelassenen FPFH. Das Ziel ist es, aus den berechneten FPFH für Ziel- und Datenpunkt-

wolke die unterschiedlichsten herauszufiltern. Die geforderte Abweichung wird über das Vielfache ( $\alpha$ ) der Standardabweichung bestimmt. Hohe Werte für  $\alpha$  fordern eine große Abweichung vom Durchschnittshistogramm, niedrige eine entsprechend kleine. Zur Untersuchung wurde keine Verwerfung der Korrespondenzen durch einen spezifizierten Korrespondenzgrenzwert vorgenommen. Eine hohe Anzahl von Korrespondenzen kann dadurch als ein Vorkommen vieler verschiedener Strukturen in einer Szene betrachtet werden. Analog dazu sind viele ähnliche Strukturen vorhanden, wenn die Anzahl der verbleibenden Korrespondenzen gering ist. Die Ergebnisse der Untersuchungen sind in Abbildung 11.3 einzusehen.



**Abbildung 11.3.: FPFH Abweichung Durchschnittshistogramm** - Eine Verbesserung des Registrierungsergebnisses durch die Verwerfung von FPFH, welche dem Durchschnittshistogramm zu ähnlich sind, erfolgt im Wesentlichen bei diskret reduzierten Punktwolken.

Allgemein ist zu sehen, dass eine zu hohe Wahl von  $\alpha$  dazu führt, dass keine Korrespondenzen gebildet werden können. Resultat ist der globale Fehler als Registrierungsergebnis. Während sich mit steigendem  $\alpha$  das Registrierungsergebnis auf räumlich reduzierten Punktwolken zunehmend verschlechtert, verbessert sich dieses bei diskret reduzierten Punktwolken. Abhängig von der Szene existiert dafür jedoch eine Obergrenze, ab welcher sich dieses über den initialen Fehler hinweg verschlechtert. Betrachtet man die Anzahl der verbleibenden Korrespondenzen ist zu sehen, dass diese mit steigendem  $\alpha$  sinkt. Die Reduktionsrate ist bei diskret reduzierten Punktwolken jedoch geringer als dies bei räumlichem Sub-Sampling zu

beobachten ist. Zusätzlich sind bei räumlich reduzierten Punktdaten Unterschiede in den einzelnen Szenarien zu verzeichnen.

Eine Erklärung dafür liefert die Betrachtung des Durchschnittshistogramms. Wie bereits im vorherigen Abschnitt festgestellt führt diskretes Sub-Sampling zu überwiegend ähnlichen FPFH. Das daraus resultierende Durchschnittshistogramm wird aller Wahrscheinlichkeit nach ähnliche Ausprägungen wie die überwiegend vorkommenden FPFH aufweisen. Für ein geringes  $\alpha$  werden dadurch bereits sehr viele FPFH verworfen. Die verbleibenden FPFH besitzen besondere Ausprägungen und erlauben eine korrekte Korrespondenzfindung wodurch sich das Registrierungsergebnis verbessert. Räumliches Sub-Sampling hingegen repräsentiert die einzelnen Strukturen der gesamten Szene weniger dicht, wodurch abhängig von der Szenerie, mehr unterschiedliche FPFH entstehen. Dies führt zu einem Durchschnittshistogramm, welches keine besonderen Ausprägungen enthält. Bereits für sehr kleine Werte von  $\alpha$  werden dadurch markante FPFH verworfen. Wird ein FPFH der Zielpunktwolke verworfen, dessen korrespondierendes FPFH der Datenpunktwolke aber nicht, entstehen falsche Korrespondenzen und beeinflussen das Registrierungsergebnis negativ.

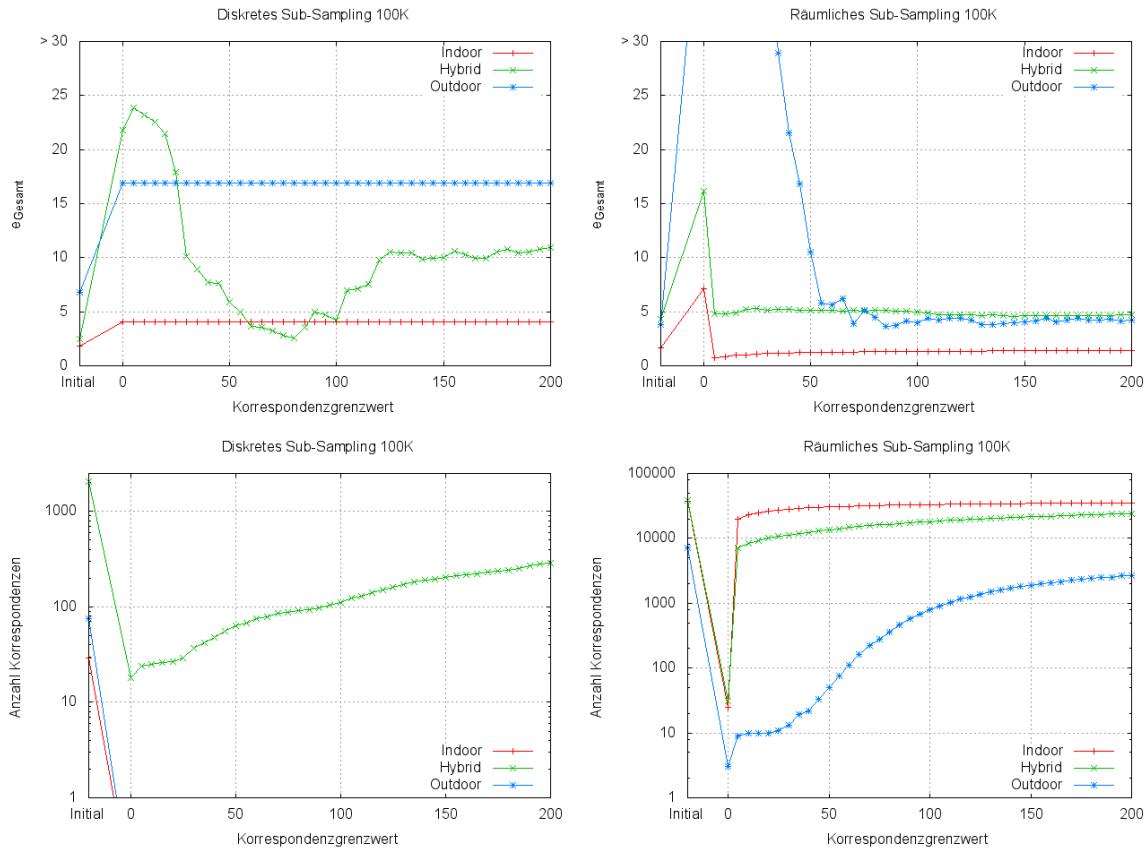
### 11.1.4. Korrespondenzgrenzwert

Die Verwerfung erzeugter Korrespondenzen, deren korrespondierenden FPFH einen spezifizierten Grenzwert überschreiten, hat eine Identifizierung falscher Korrespondenzen zur Aufgabe. Dabei wird davon ausgegangen, dass sich die Güte einer Korrespondenz über die Unterschiedlichkeit der korrespondierenden FPFH beschreiben lässt. Grundlage für die hier durchgeführten Untersuchungen sind die aus den vorherigen Abschnitten individuell für Szene und Sampling-Strategie ermittelten Parameterwerte. Die Registrierungsergebnisse der einzelnen Szenarien sind in Abbildung 11.4 festgehalten.

Erwartungsgemäß steigt mit dem Korrespondenzgrenzwert auch die Anzahl der verbleibenden Korrespondenzen. Die Stärke des Anstiegs gibt Auskunft darüber, in welchem Verhältnis sehr ähnliche bzw. sehr unterschiedliche, korrespondierende FPFH in der initialen Korrespondenzmenge enthalten sind. Ein linearer Verlauf deutet demnach auf eine Gleichverteilung dieser hin. Steigt die Anzahl der Korrespondenzen logarithmisch, sind die korrespondierenden FPFH sehr unterschiedlich, entsprechend ähnlich bei einem exponentiellen Verlauf. Anhängig ist dies von der jeweiligen Szene und Sub-Sampling-Strategie, was sich mit den Beobachtungen im vorherigen Abschnitt (11.1.3 Abweichung Durchschnittshistogramm) deckt. In welcher Größenordnung sich die Ähnlichkeit der FPFH dabei definiert, ist ebenfalls von der Sub-Sampling-Strategie und Szene abhängig. Auf räumlich reduzierten Punktwolken bleibt die Anzahl der Korrespondenzen selbst bei einer Änderung des Korrespondenzgrenzwertes um 10000 konstant. Das Hybrid-Szenario hingegen zeigt deutliche Reaktionen bei einer Änderung von dessen um 5. Diese Tatsache macht die Bestimmung eines Korrespondenzgrenzwertes äußerst problematisch.

Betrachtet man das Registrierungsergebnis wird deutlich, dass durch die Verwendung eines Korrespondenzgrenzwertes hier keine Verbesserung erzielt werden kann. Die initiale Korrespondenzmenge lieferte für alle Szenarien und Sub-Sampling-Strategien ein besseres

## 11.1. Point Feature Histogram



**Abbildung 11.4.: FPFH Korrespondenzgrenzwert** - „Initial“ entspricht einem unendlichen Korrespondenzgrenzwert (keine Korrespondenzverwerfung). Die Korrespondenzverwerfung führt in keinem Szenario zu einer Verbesserung des Registrierungsergebnisses. Bei einer diskreten Reduktion der Indoor- und Outdoor-Szene ist erst ab einem Grenzwert von 1000 die minimal sinnvolle Anzahl von drei Korrespondenzen verfügbar.

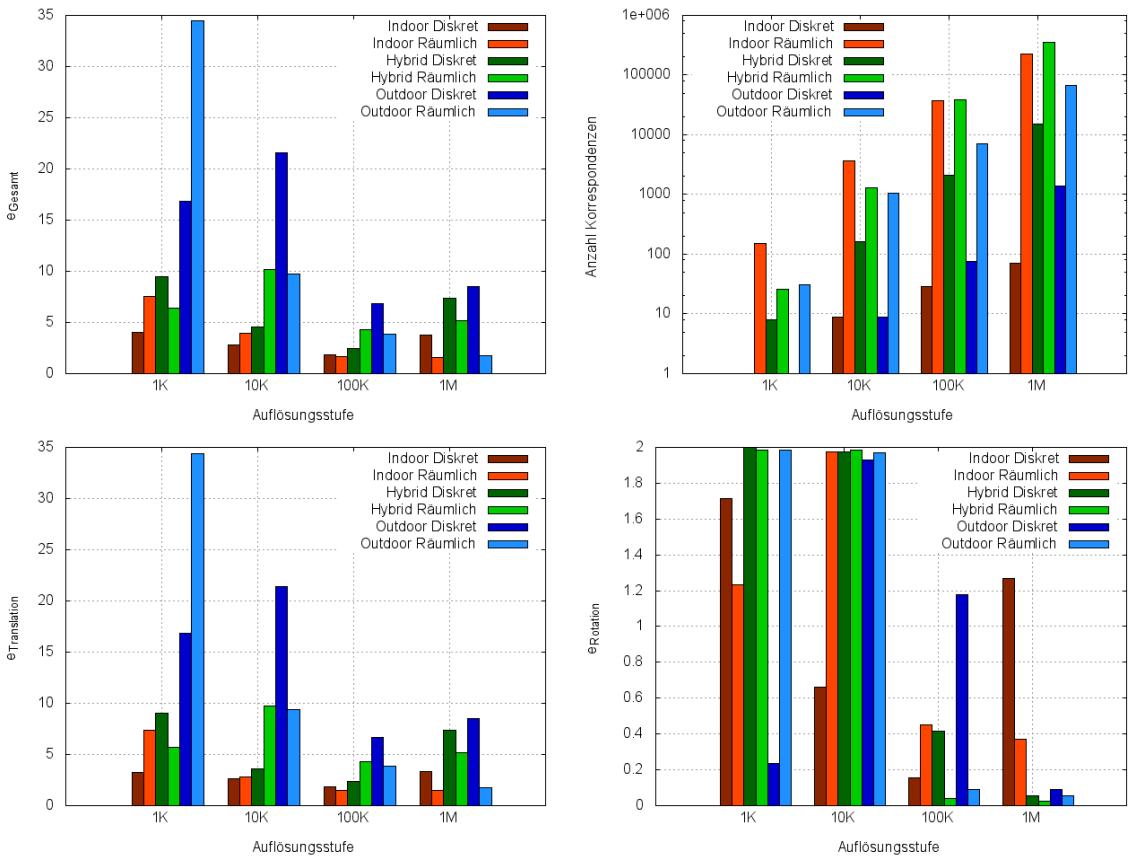
Ergebnis. Die Verwerfung von Korrespondenzen, über die Forderung einer spezifizierten Ähnlichkeitsbeziehung der korrespondierenden FPFH, scheint in diesem Registrierungsschema einen eher negativen Einfluss auf die Qualität der Registrierung zu haben. In den nachfolgenden Untersuchungen wird daher auf ein Verwerfung der Korrespondenzen über den hier präsentierten Korrespondenzgrenzwerte verzichtet.

### 11.1.5. Konvergenz

Mit Hilfe der vorhergehenden Parameterstudie wurden für die unterschiedlichen Szenen und Sub-Sampling-Strategien Parametersets definiert, die bei der verwendeten Auflösung von 100K scheinbar gute Registrierungsergebnisse erzielen. Diese wurden anschließend zur Registrierung auf verschiedenen Auflösungen verwendet. Da der PFH-Algorithmus keine initiale Ausrichtung verarbeitet, werden die globalen Fehler (siehe Anhang C) als

## 11. Analyse

Konvergenzkriterien im Sinne der definierten Anforderungen verwendet. Die Ergebnisse sind in Abbildung 11.5 präsentiert.



**Abbildung 11.5.: FPFH Konvergenz** - Abgebildet ist der Gesamt-, Translations- und Rotationsfehler nach einer Registrierung der unterschiedlichen Szenarien auf verschiedenen, räumlichen und diskreten Auflösungen. Außerdem ist die Anzahl der verwendeten Korrespondenzen abgebildet. Die Ausgangssituation ist durch die globale Ausrichtung gegeben. Der damit verbundene initiale Gesamtfehler liegt für die Indoor-Szenerie bei 4.05, für die hybride Szene bei 7.30 und für das Outdoor-Szenario bei 16.87 (siehe Anhang C).

Für die Auflösungen 100K und 1M konnten alle Szenarien erfolgreich registriert werden. Dabei erzielen räumliche reduzierte Punktwolken besonders bei höheren Auflösungen ein besseres Ergebnis. Die geringe Anzahl der Korrespondenzen auf diskret reduzierten Punktdaten lässt den Schluss zu, dass die hohe Punktdichte zu einer Vielzahl ähnlicher FPFH führt. Je geringer die Anzahl vorhandener Korrespondenzen, desto größer ist der Einfluss falscher Korrespondenzpaare auf das Registrierungsergebnis. Ein Ausbilden ausschließlich korrekter Korrespondenzen wird bei hohen Auflösungen als nahezu unmöglich erachtet.

Bei einer diskreten Auflösung von 1K konnten für Indoor- und Outdoor-Szenario keine Korrespondenzen ermittelt werden. Die Ursache ist, dass alle FPFH aufgrund zu hoher Ähnlichkeit durch die Abweichung vom Durchschnittshistogramm verworfen wurden. Eine räumliche Auflösung von 1K erlaubt das Bilden von unterschiedlichen FPFH. Eine erfolgrei-

che Registrierung ist dennoch nicht möglich. Dies ist auch bei einer diskreten und räumlichen Auflösung von 10K zu beobachten. Auffällig sind dabei die sehr hohen Rotationsfehler und im Allgemeinen eher geringen Translationsfehler. Eine Betrachtung der Korrespondenzpaare zeigt, dass FPFH markanter Bereiche korrekt zugeordnet wurden. FPFH der in allen Szenarien sehr ausgedehnten Böden führen jedoch zu falschen Korrespondenzen zwischen dem Bodenbereich der Ziel- und Datenpunktfolge. Dadurch erfolgt ein im Wesentlichen richtiges Übereinanderschieben des Bodens. Anteilig sind die Korrespondenzen der Bodenbereiche den Korrespondenzpaaren markanter Bereiche überlegen und verhindern dadurch ein Auflösen der Rotation.

Aufgrund zu langer Laufzeiten (siehe nächster Abschnitt 11.1.6) wurde eine Auflösung von 10M nicht untersucht. Vermutlich würde das Registrierungsergebnis für diskret reduzierte Punktdaten erneut schlechter werden, während es sich beim räumlichen Sub-Sampling weitgehend konstant verhielte.

Das genaueste Registrierungsergebnis wurde für das räumlich reduzierte Indoor-Szenario, bei einer Auflösung von 1M erzielt. Dabei betrug  $\vec{e}_{Translation} = (0.99, -1.38, -0.68)$  m und  $\vec{e}_{Rotation} = (0.00, 0.07, 0.02)$  m.

### 11.1.6. Effizienz

Die in Abbildung 11.6 gemessenen Laufzeiten beziehen sich auf die im vorherigen Abschnitt (Konvergenz 11.1.5) durchgeführten Untersuchungen.

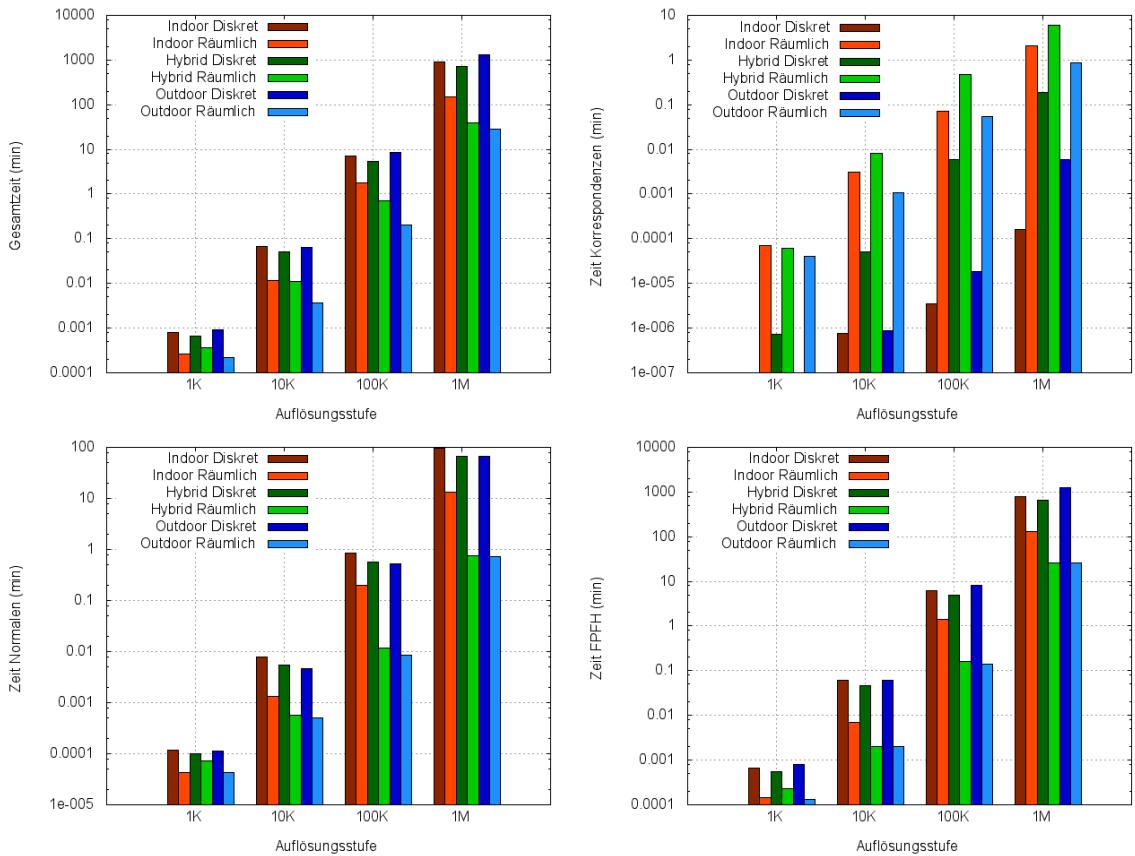
Festzustellen ist, dass diskret reduzierte Punktwolken für alle Auflösungen deutlich längere Laufzeiten zur Berechnung der Normalen und FPFH aufweisen als räumlich reduzierte. Begründet liegt dies in der höheren Punktdichte diskret reduzierter Punktwolken. Sie führt zu einer höheren Anzahl von Punkten in einer definierten Kugelumgebung jedes einzelnen Punktes, als dies bei räumlich reduzierten Punktdaten der Fall ist.

Wie bereits bei der Untersuchung der Parameter festgestellt, führt ein räumliches Sub-Sampling zu einer größeren Anzahl von Korrespondenzen. Dementsprechend ist der Zeitaufwand zur Korrespondenzfindung deutlich höher als bei diskretem Sub-Sampling. Der Einfluss auf die Gesamlaufzeiten ist speziell bei niedrigen Auflösungen enorm.

Unterschiedliche Laufzeiten sind ebenfalls zwischen den verschiedenen Szenarien zu erkennen. Rückzuführen ist dies ebenfalls auf die Punktdichte, welche mitunter auch von der Beschaffenheit einer Szene abhängig ist. In der Theorie ist die Zeitkomplexität für das Berechnen der Normalen und FPFH linear in der Anzahl der Punkte. Große Radien, eine hohe Punktdichte und die logarithmische Suche der in einer Kugelumgebung enthaltenen Punkte auf KD-Trees, führen jedoch zu sehr hohen Konstanten und einem nahezu quadratischen Laufzeitverhalten für diskret reduzierte Punktdaten.

Für eine erfolgreich Registrierung muss mindestens eine Auflösung von 100K verwendet werden. Eine mittlere Berechnungszeit, wie für das räumlich reduzierte Hybrid-Szenario, liegt bei  $\approx 0.01$  Minuten für die Normalen,  $\approx 0.1$  Minuten für die FPFH und  $\approx 1$  Minute für das Berechnen und Verwerfen der Korrespondenzen.

## 11. Analyse



**Abbildung 11.6.: FPFH Effizienz** - Abgebildet sind die Laufzeiten zur Berechnung der Normalen, der FPFH, der Korrespondenzen sowie die Gesamlaufzeit der verschiedenen Szenarien auf unterschiedlichen, diskreten und räumlichen Auflösungen. Eine diskrete Reduktion der Punktwolken führt durch die hohe Punktdichte zu einer zeitintensiven Berechnung der Normalen und FPFH.

### 11.1.7. Stabilität

Wie die Parameterstudie (siehe Abschnitte 11.1.1, 11.1.2, 11.1.3, 11.1.4) bereits anzeigt, musste für jede Szene und Sub-Sampling-Strategie ein individuelles Parameterset definiert werden. Diese sind außerdem wenig tolerant gegenüber Abweichungen von den optimalen Parametern. Mit den individuellen Parametersets war es gemäß den Untersuchungen in Abbildung 11.5 möglich, Punktwolken verschiedener Auflösungen korrekt zu registrieren.

Für sehr geringe Auflösungen konnte keine korrekte Registrierung erfolgen (siehe Abbildung 11.5). Bei steigender diskreter Auflösung verschlechterte sich das Registrierungsergebnis. Offensichtlich sollte die Szenerie weder zu detailliert noch zu schwach repräsentiert sein. Ein stabiles Verhalten kann lediglich für räumlich reduzierte Punktwolken auf den Auflösungen 100K und 1M verzeichnet werden.

Bei höheren Auflösungen war es möglich alle drei Szenarien korrekt zu registrieren (siehe Abbildung 11.5). Die höchsten Fehlerschwankungen traten dabei für die Outdoor-Szene

auf. Sie ist durch sehr ähnliche Strukturen definiert, was einer Erzeugung markanter FPFH entgegenwirkt. Laufzeiten lassen sich nicht unbedingt voraussagen, da eine Abhängigkeit dieser von der Punktdichte und somit von der Szene selbst existiert (siehe vorheriger Abschnitt 11.1.6).

## 11.2. Spin-Images

Der SI Algorithmus ist ebenso wie der PFH-Algorithmus nicht in der Lage Vorabinformationen zu verarbeiten. Eine initiale Ausrichtung der Punktwolken ist daher durch das Platzieren der Punktwolken mit der Position des Laserscanners auf dem Ursprung des gemeinsamen Koordinatensystems (siehe Abschnitt 5.3) gegeben. Die resultierenden „globalen“ Fehler dazu sind in Anhang C einzusehen.

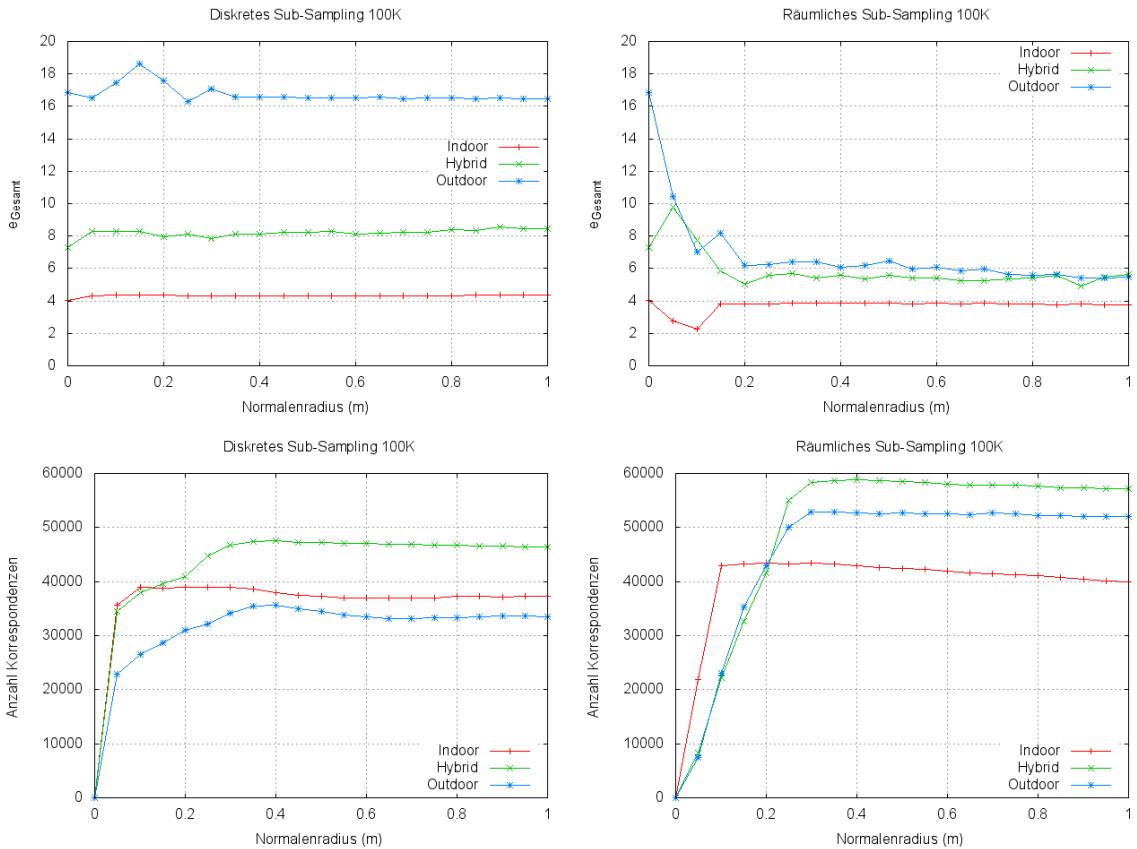
Wesentlich Kontrollelemente des Algorithmus sind der Normalenradius, der Suchradius, der Support-Winkel, die Abweichung vom Durchschnittshistogramm sowie die Verwerfung von Korrespondenzen bezüglich eines spezifizierten Grenzwertes. Diese Parameter sollen so gewählt werden, dass möglichst aussagekräftige SI aus Ziel- und Datenpunktfolke extrahiert werden. Erhofft wird sich dadurch eine Berechnung möglichst weniger, aber dafür korrekter Korrespondenzen.

### 11.2.1. Normalenradius

Wie bereits bei der Untersuchung des Normalenradius der FPFH (siehe Abschnitt 11.1.1) wurde der Merkmalsradius so gewählt, dass unabhängig von der Sub-Sampling-Strategie ausreichend viele Punkte in der Umgebung eines Punktes zur Berechnung eines aussagekräftigen SI zu finden sind. Ebenfalls wurde keine Verwerfung der SI durch die Abweichung vom Durchschnittshistogramm vorgenommen. Durch die Prüfung der Korrespondenzen auf Eindeutigkeit gibt die Anzahl dieser Auskunft über die Ähnlichkeit der SI. Eine geringe Anzahl entspricht dabei vielen ähnlichen SI, eine hohe weist auf unterschiedliche hin. Festgehalten sind die Ergebnisse in Abbildung 11.7.

Festzustellen ist, dass die Unterschiedlichkeit der SI mit steigendem Normalenradius zunimmt, ab einem gewissen Normalenradius jedoch gesättigt verläuft. Der Normalenradius ab welchem eine Sättigung eintritt scheint dabei unabhängig von der Sub-Sampling-Strategie. Ein Zusammenhang zum Merkmalsradius kann ebenfalls nicht festgestellt werden. Diskret reduzierte Punktdaten weisen in allen Szenarien eine geringere Anzahl von Korrespondenzen auf als dies für räumliches Sub-Sampling der Fall ist. Auch steigt die Anzahl der Korrespondenzen auf räumliche reduzierten Punktwolken linear, während eine diskrete Reduktion zu einem logarithmischen Verlauf bis zur Sättigung führt. Dieses Verhalten wurde bereits beim Normalenradius der FPFH festgestellt (siehe Abschnitt 11.1.1) und lässt sich offensichtlich auf dieselbe Ursache zurückführen, nämlich dass eine höhere Punktdichte größere Räden zur Berechnung verschiedener Normalen erfordert.

## 11. Analyse



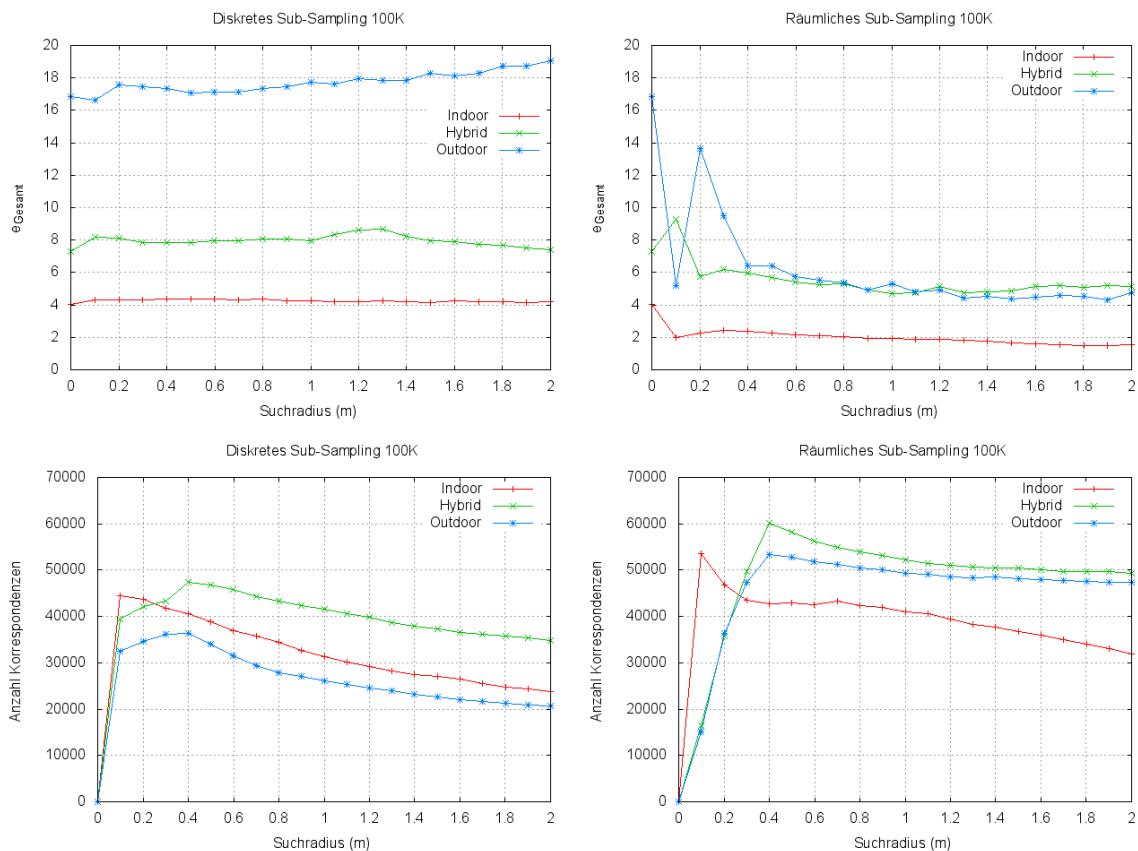
**Abbildung 11.7.: SI Normalenradius** - Eine geringe Anzahl von Korrespondenzen indiziert ähnliche SI, eine sehr hohe entsprechend unterschiedliche. Bei einem räumlichem Sub-Sampling verbessert sich mit einer steigenden Anzahl unterschiedlicher SI auch das Registrierungsergebnis.

Da über die Normale eines Punktes die Rotationsachse seines SI definiert wird scheint es vernünftig, dass benachbarte Punkte mit sehr ähnlichen Normalen, nahezu identische Bereiche im SI klassifizieren, wodurch diese nur geringe bzw. keine Unterschiede aufweisen. Die Indoor-Szene zeigt außerdem, dass zu große Radien ebenfalls ähnliche SI erzeugen können, da der Einfluss großer Strukturen, welche durch viele Punkte repräsentiert werden (wie z.B. Flächen), die Normale eines Punktes verfälschen können.

Betrachtet man den Einfluss des Normalenradius auf das Registrierungsergebnis, ist lediglich für die Outdoor-Szene eine klare Reaktion zu verbuchen. Bei ihr bleibt das Registrierungsergebnis ab einem Normalenradius stabil, welcher in etwa der maximalen Anzahl von Korrespondenzen, also vielen verschiedenen SI entspricht. Dieses Verhalten ist auch auf räumlich reduzierten Punktdaten zu beobachten. Bei ihnen verbessert sich das Registrierungsergebnis mit steigender Anzahl verschiedener SI. Lediglich die Indoor-Szene zeigt trotz konstanter Anzahl von Korrespondenzen eine Verschlechterung. Die Ursache wird in der bereits erwähnten Verfälschung von Normalen bei zu großen Radien vermutet.

### 11.2.2. Suchradius

Um Aufschluss über die Ähnlichkeit der berechneten SI zu erhalten, wurde auch für die Untersuchungen zum Suchradius keine Prüfung der SI auf Einzigartigkeit durchgeführt. Die Verwerfung nicht eindeutiger Korrespondenzen erlaubt daher, wie im vorherigen Abschnitt (11.2.1 Normalenradius), eine Aussage über die Ähnlichkeit der SI. Die erzielten Ergebnisse sind in Abbildung 11.8 präsentiert. Dabei wurden die verschiedenen Szenarien auf unterschiedlichen räumlichen und diskreten Auflösungen bei einem variierenden Suchradius registriert.



**Abbildung 11.8.: SI Suchradius** - Eine geringe Anzahl von Korrespondenzen indiziert ähnliche FPFH, eine sehr hohe entsprechend unterschiedliche. Auf räumlich reduzierten Punktfolgen verbessert sich das Registrierungsergebnis mit steigendem Suchradius, bis die Verschiedenheit der SI zu sinken beginnt.

Unabhängig von Sub-Sampling-Strategie und Szenario ist zu beobachten, dass die Verschiedenheit der SI mit dem Suchradius bis zu einem Maximum steigt. Eine weitere Erhöhung des Suchradius führt zu einer Abnahme der Korrespondenzanzahl. Erklären lässt sich dieses Verhalten dadurch, dass der Suchradius im Wesentlichen die Größe des Bereiches festlegt, aus welchem Punkte in das SI projiziert werden. Werden sehr große Bereiche betrachtet, wird die Klassifizierung der einzelnen Punkte sehr grob, wodurch ähnliche SI entstehen. Bei

einer kleinen Wahl des Suchradius werden zu kleine Bereiche betrachtet um die eigentliche Form der betrachteten Oberfläche noch abbilden zu können.

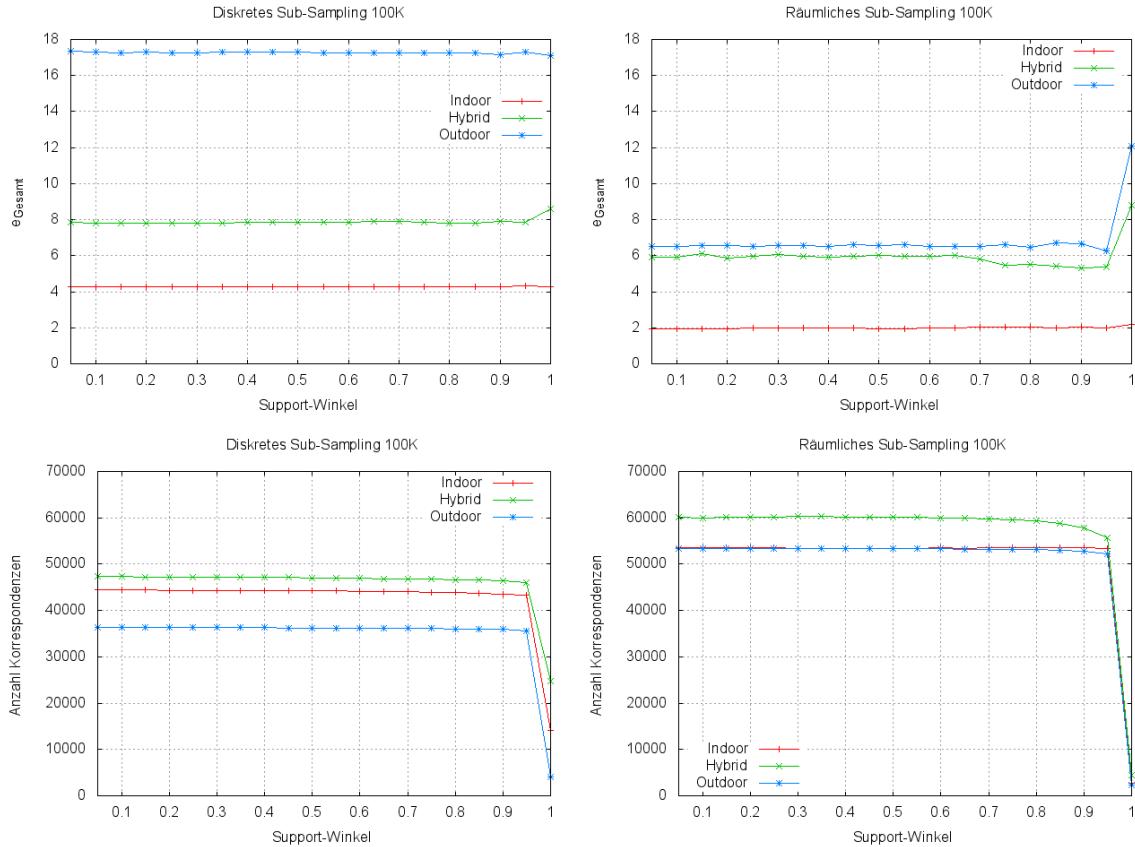
Vergleicht man die Anzahl der Korrespondenzen mit dem Registrierungsfehler kann für räumliches Sub-Sampling gesagt werden, dass der Registrierungsfehler mit einer steigenden Verschiedenheit der SI bis zu einem Minimum sinkt und sich anschließend einpendelt. Auf diskret reduzierten Punktwolken ist kaum ein Einfluss des Merkmalsradius zu erkennen. Eine zu große Wahl führt beim Outdoor-Szenario zum Divergieren, das Hybrid-Szenario alterniert um den initialen Fehler und das Indoor-Szenario zeigt überhaupt keine Reaktion. Vermutlich führt die hohe Punktdichte dazu, dass selbst bei kleinen Radien bereits SI entstehen, die unterschiedlich genug sind um eindeutige Korrespondenzen auszubilden, aber nicht in einem Maß, welches einer korrekte Ausbildung von Korrespondenzen erlaubt.

### 11.2.3. Support-Winkel

Analog zu den vorherigen zwei Abschnitten kann auch in diesem die Ähnlichkeit der SI anhand der Anzahl berechneter Korrespondenzen abgelesen werden. Die Ergebnisse der durchgeführten Untersuchung sind in Abbildung 11.8 präsentiert. Dabei wurden die verschiedenen Szenarien auf unterschiedlichen räumlichen und diskreten Auflösungen bei einem variierenden Support-Winkel registriert.

In den hier untersuchten Szenarien konnte kein wesentlicher Einfluss des Support-Winkels auf das Registrierungsergebnis oder die Ähnlichkeit der SI festgestellt werden. Lediglich für einen Support-Winkel von 1 (Kosinus des Winkels) ist für alle Szenarien zu beobachten, dass die Verschiedenheit der SI sinkt und der Registrierungsfehler steigt.

Ein Support-Winkel von 1 bedeutet, dass ein benachbarter Punkt nur dann in das SI projiziert wird, wenn seine Normale eine mit der Rotationsachse identische Richtung aufweist. Eine Betrachtung der resultierenden SI hat gezeigt, dass dadurch nur sehr wenige Punkte projiziert werden. Die Einträge nahezu aller SI sind daher 0 und diese somit identisch. Der nicht vorhandene Einfluss der Support-Winkel kleiner als 1 wird in den weitestgehend rauschfreien Punktdaten sowie der Glättung von SI durch die bilineare Interpolation vermutet. Indiz dafür ist der mittlere quadratische Abstand zwischen korrespondierenden Punkten, der ebenfalls unverändert bleibt und dadurch eine identische Korrespondenzbildung signalisiert. Einzige Ausnahme ist das Hybrid-Szenario auf räumliche reduzierten Punktdaten, welches sehr kleine und markante Strukturen enthält, die bei einer geringen Auflösung wahrscheinlich als Rauschen interpretiert werden.



**Abbildung 11.9.: SI Support-Winkel** - Bei einem Support-Winkel von 1 muss die Normale eines Punktes identisch zur Rotationsachse sein, um auf das SI projiziert zu werden. Dies führt dazu, dass für beinahe alle SI keine Abbildung der Punkte aus dem Suchraum erfolgt, wodurch diese identisch sind.

#### 11.2.4. Abweichung Durchschnittshistogramm

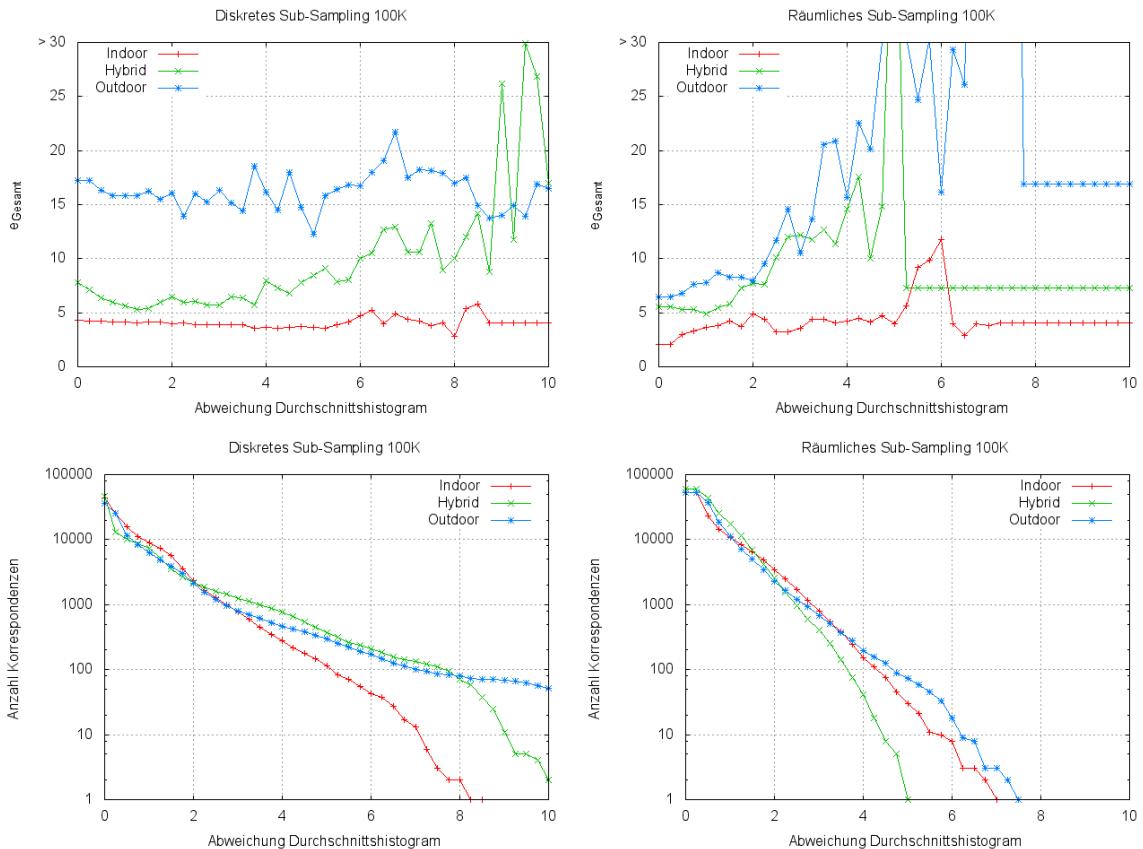
Analog zur Abweichung vom Durchschnittshistogramm bei den PFH (siehe Abschnitt 11.1.3), entscheidet auch dieser Parameter über die Anzahl und Ähnlichkeit der zur Korrespondenzsuche zugelassenen SI. Die geforderte Abweichung wird auch hier über das Vielfache ( $\alpha$ ) der Standardabweichung bestimmt, so fordern hohe Werte für  $\alpha$  eine große, niedrige eine entsprechend kleine Abweichung vom Durchschnittshistogramm. Zur Untersuchung wurde keine Verwerfung der Korrespondenzen durch einen spezifizierten Korrespondenzgrenzwert vorgenommen. Die Ergebnisse der auf unterschiedlichen, räumlich und diskret reduzierten Szenarien durchgeführten Untersuchungen sind Abbildung 11.10 visualisiert.

Wie zu erwarten sinkt bei beiden Sub-Sampling-Strategien die Anzahl der Korrespondenzen mit steigendem  $\alpha$ . Bei räumlich reduzierten Daten erfolgt dies jedoch wesentlich schneller. Vermutlich entsteht durch die bereits in den vorherigen Abschnitten festgestellte hohe Verschiedenheit der SI ein sehr allgemeines Durchschnittshistogramm mit hoher Standardabweichung. Dadurch werden schon bei kleinen Werten für  $\alpha$  sehr viele SI verworfen.

## 11. Analyse

Untermauern lässt sich dies durch den mit  $\alpha$  steigenden Registrierungsfehler auf nahezu allen Szenarien. Offensichtlich werden SI entfernt deren Existenz, eine Ausbildung korrekter Korrespondenzen fordert.

Bei diskret reduzierten Punktdaten ist ein allgemeines Alternieren um den Ausgangsfehler zu beobachten. Für einzelne  $\alpha$ -Werte jedoch ist eine deutliche Verbesserung zu beobachten. Auch hier bildet die Hybrid-Szene eine Ausnahme. Sie verhält sich ähnlich wie auf räumlich reduzierten Punktdaten.



**Abbildung 11.10.: SI Abweichung Durchschnittshistogramm** - Eine Verbesserung des Registrierungsergebnisses durch die Verwerfung von SI welche dem Durchschnittshistogramm zu ähnlich sind, erfolgt im Wesentlichen bei diskret reduzierten Punktwolken. Ein räumliches Sub-Sampling führt vermutlich zu bereits sehr unterschiedlichen SI.

### 11.2.5. Korrespondenzgrenzwert

Die Verwerfung erzeugter Korrespondenzen, deren korrespondierenden SI einen spezifizierten Grenzwert überschreiten, hat wie bei den PFH (siehe Abschnitt 11.1.4) eine Identifizierung falscher Korrespondenzen zur Aufgabe. Die Güte einer Korrespondenz ist dabei über die Merkmalsdistanz der korrespondierenden SI definiert. Für die hier durchgeführten Untersuchungen wurden die aus den vorherigen Abschnitten individuell für Szene

und Sampling-Strategie ermittelten Parameterwerte verwendet. Die Ergebnisse der Registrierungen auf den einzelnen Szenarien sind in Abbildung 11.11 festgehalten. Der initiale Fehler entspricht dabei einer Registrierung ohne Korrespondenzverwerfung durch einen Korrespondenzgrenzwert.

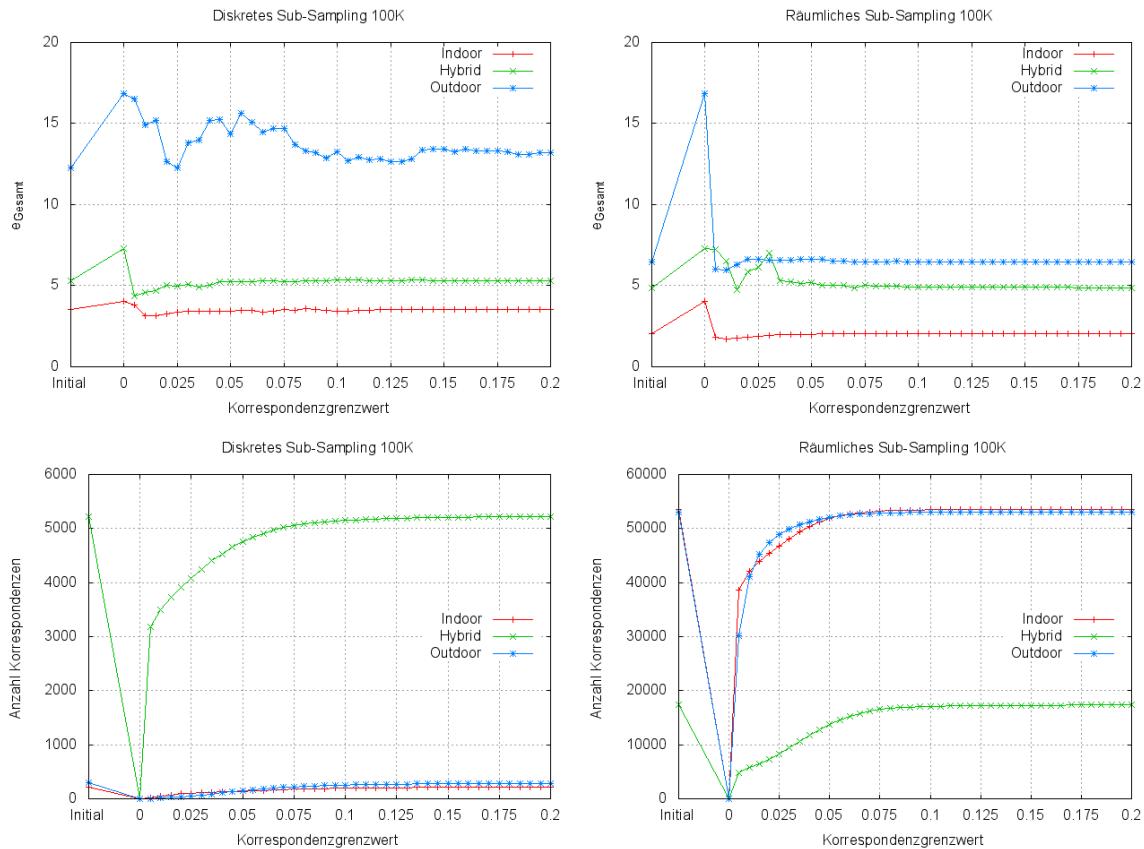


Abbildung 11.11.: SI Korrespondenzgrenzwert - „Initial“ entspricht einem unendlichen Korrespondenzgrenzwert (keine Korrespondenzverwerfung). Die Korrespondenzverwerfung bezüglich eines spezifizierten Grenzwertes verbessert das Registrierungsergebnis nur unwesentlich.

Ein Korrespondenzgrenzwert von 0 erfordert Korrespondenzen, deren korrespondierenden SI absolut identisch sind. Wie an der Anzahl der daraus resultierenden Korrespondenzen zu sehen ist, führt dies bei allen Szenarien, unabhängig von der Sub-Sampling-Strategie dazu, dass alle Korrespondenzen verworfen werden.

Indoor- und Hybrid-Szenario zeigen auf diskret reduzierten Punktdaten für sehr kleine Korrespondenzgrenzwerte eine geringe Verbesserung des Registrierungsergebnisses. Offensichtlich repräsentiert die Merkmalsdistanz der korrespondierenden SI bei diesen Szenarien die Güte der Korrespondenzen relativ gut. Weniger eindeutig verhält sich das diskret reduzierte Outdoor-Szenario, welches starke Schwankungen zwischen sehr schlechten und dem initialen Registrierungsergebnis aufweist. Für größere Werte als die in Abbildung 11.11 präsentierten verhält es sich jedoch stabil.

Auf räumlich reduzierten Punktdaten ist ein ähnliches Verhalten für die Hybrid-Szene zu beobachten. Offensichtlich enthalten beide Szenarien Korrespondenzen, deren korrespondierenden SI eine gewisse Verschiedenheit aufweisen, aber dennoch korrekte Korrespondenzen bilden. Die räumlich reduzierten Indoor- und Outdoor-Szenarien hingegen verhalten sich stabil. Eine Verbesserung des initialen Fehlers ist lediglich für sehr kleine Korrespondenzgrenzwerte zu beobachten.

Den Ergebnissen nach zu urteilen hat eine Verwerfung von Korrespondenzen durch einen spezifizierten Korrespondenzgrenzwert in dem hier betrachteten System nur einen marginalen Einfluss auf die Verbesserung des Registrierungsergebnisses. Vermutlich führt das räumliche Sub-Sampling zu bereits sehr markanten SI; die starke Reduktion der SI durch den Vergleich am Durchschnittshistogramm führt offensichtlich bei diskret reduzierten Daten hingegen zu einer sehr elitären Menge an Korrespondenzen.

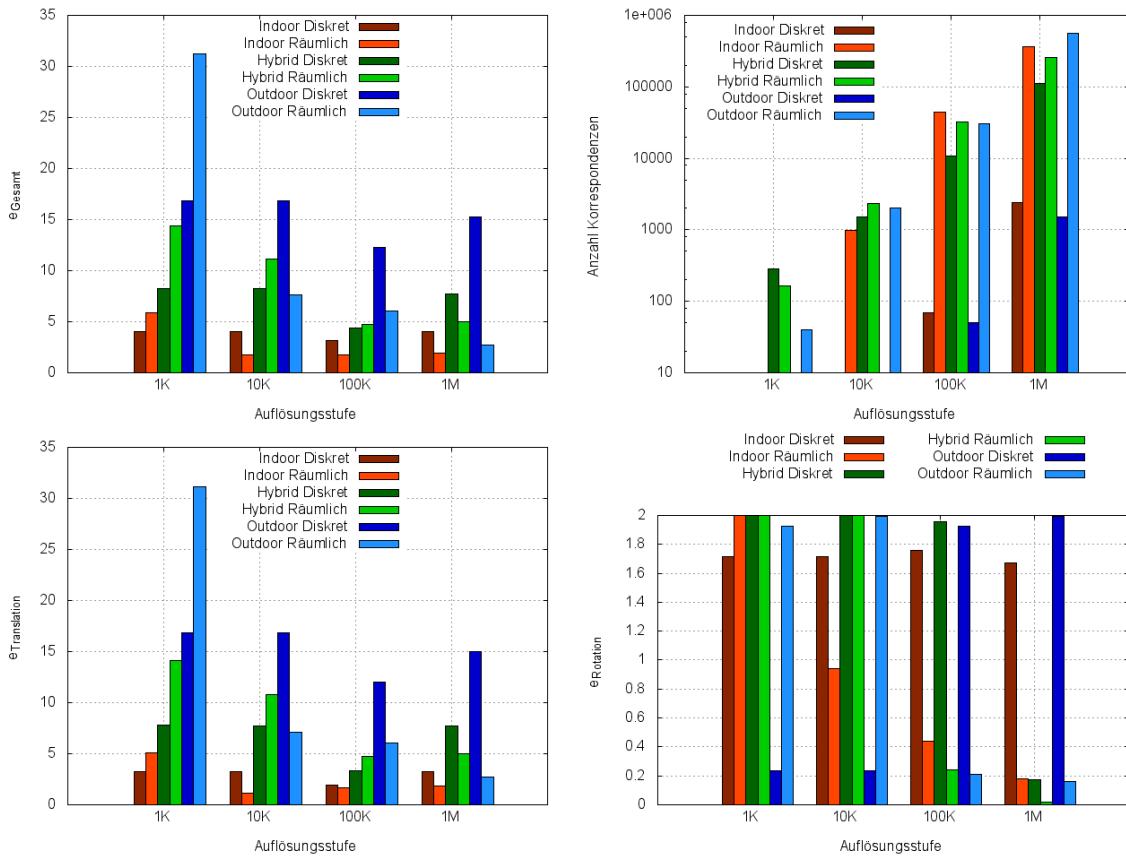
### 11.2.6. Konvergenz

Mit Hilfe der vorhergehenden Parameterstudie wurden für die unterschiedlichen Szenen und Sub-Sampling-Strategien Parametersets definiert, die für eine Auflösung von 100K scheinbar gute Registrierungsergebnisse erzielen. Für die folgenden Untersuchungen wurden sie zur Registrierung auf verschiedenen Auflösungen verwendet. Da der SI-Algorithmus nicht in der Lage ist eine initiale Ausrichtung zu verarbeiten, werden die globalen Fehler (siehe Anhang C) als Konvergenzkriterien im Sinne der definierten Anforderungen verwendet. Die Ergebnisse sind in Abbildung 11.5 präsentiert.

Bei einer diskreten und räumlichen Auflösung von 1K konnte keines der Testszenarien erfolgreich registriert werden. Das Hybrid-Szenario ist dabei die einzige Szenerie, für die sowohl auf räumlich wie auch diskret reduzierten Punktdaten Korrespondenzen gefunden werden konnten. Mittels räumlichem Sub-Sampling lies sich das Hybrid-Szenario bereits bei einer Auflösung von 10K registrieren. Bei einer diskreten Reduktion konnten ebenfalls keine Korrespondenzen gefunden werden. Dieselbe Problematik ist auch für das Outdoor-Szenario zu beobachten. Sehr geringe diskrete Auflösungen führen offensichtlich dazu, dass zu wenig markante Struktur zu detailliert vorhanden ist (große Bodenflächen), was zu sehr ähnlichen SI führt, die dann verworfen werden. Indiziert wird dies durch die Tatsache, dass durch die Abweichung vom Durchschnittshistogramm alle SI verworfen wurden.

Mit einer steigenden räumlichen Auflösung sinkt der Registrierungsfehler bei allen Szenarien, sodass diese konvergieren. Auch für höhere diskrete Auflösungen ist das zu beobachten. Der sehr hohe Rotationsfehler und eine Betrachtung der Korrespondenzpaare allerdings weisen auf das altbekannte Problem der großen Bodenflächen hin (siehe Abbildung 11.1.5). Diese führen dazu, dass Punkte korrespondierender Objekte falsche Korrespondenzen ausbilden. Korrekte Korrespondenzen anderer Objekte führen dann vermutlich zu einer symmetrischen Rotation. Der translatorische Anteil kann durch das Übereinanderschieben der störenden Objekte (hier im allgemeinen Flächen) dennoch teilweise gelöst werden.

Aufgrund unpraktikabler Laufzeiten (siehe nächster Abschnitt 11.2.7) wurde eine Auflösung von 10M nicht untersucht. Der Verlauf des Registrierungsfehler lässt vermuten, dass keine



**Abbildung 11.12.: SI Konvergenz** - Abgebildet ist der Gesamt-, Translations- und Rotationsfehler nach einer Registrierung der unterschiedlichen Szenarien auf verschiedenen, räumlichen und diskreten Auflösungen. Außerdem ist die Anzahl der Korrespondenzen abgebildet. Als initiale Ausrichtung ist die globale gegeben, der damit verbundene initiale Gesamtfehler liegt für die Indoor-Szenerie bei 4.05, für die hybride Szene bei 7.30 und für das Outdoor-Szenario bei 16.87 (siehe Anhang C).

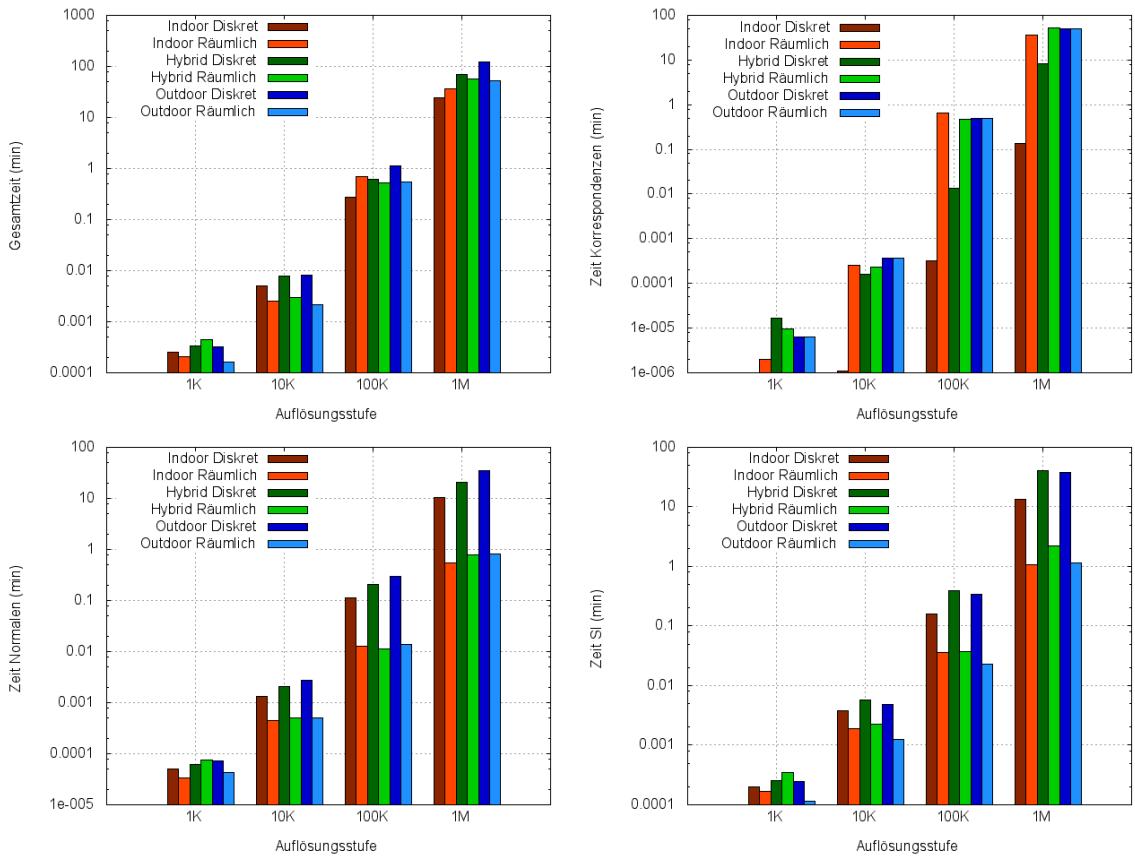
wesentliche Verbesserung des Registrierungsergebnisses zu erwarten ist. Basierend auf diesen Untersuchungen ist auszumachen, dass für ein erfolgreiche Registrierung der hier untersuchten Szenarien eine räumliche Reduktion der Punktwolken erforderlich ist.

Die genaueste Registrierung wurde beim räumlich reduzierten Indoor-Szenario mit einer Auflösung von 100K erzielt. Dabei betrug  $\bar{e}_{Translation} = (1.26, -1.06, -0.19) \text{ m}$  und  $\bar{e}_{Rotation} = (0.22, 0.05, 0.04) \text{ m}$ .

### 11.2.7. Effizienz

Die in Abbildung 11.6 festgehaltenen Laufzeiten beziehen sich auf die im vorherigen Abschnitt (Konvergenz 11.2.6) durchgeföhrten Untersuchungen. Eine Auflösung von 10M wurde aufgrund zu hoher und damit wenig praktikabler Laufzeiten nicht untersucht.

## 11. Analyse



**Abbildung 11.13.: SI Effizienz -** Abgebildet sind die Laufzeiten zur Berechnung der Normalen, der SI, der Korrespondenzen und die Gesamlaufzeit der verschiedenen Szenarien auf unterschiedlichen, diskreten und räumlichen Auflösungen.

Wie aufgrund der hohen Punktdichte bei diskret reduzierten Punktwolken zu erwarten, erfordert die Berechnung der Normalen und SI hier deutlich mehr Zeit als dies für räumlich reduzierte Punktdaten der Fall ist. Hingegen erfordert die deutliche höhere Anzahl von Korrespondenzen, welche durch räumliches Sub-Sampling zustande kommen (siehe vorheriger Abschnitt 11.2.6), mehr Zeit bei der Berechnung der Korrespondenzen (Korrespondenzfindung und Verwerfung).

Ergebnis ist, dass insgesamt betrachtet nur gering höhere Laufzeiten für diskret reduzierte Punktdaten entstehen. Eine Abhängigkeit von der Szenerie kann auf Grund der Untersuchungsergebnisse nur spekuliert werden. Vermutet wird jedoch, dass bei einer gleichen Parameterwahl eine nahezu konstante Gesamlaufzeit zu erwarten ist, da höhere Punktdichten zu mehr Laufzeit bei der Berechnung der Normalen und SI führen, aber im Allgemeinen auch zu eher ähnlichen SI und somit zu einer geringeren Anzahl der zur Korrespondenzsuche zugelassenen Punkte.

Für eine erfolgreich Registrierung muss mindestens eine Auflösung von 100K verwendet werden. Eine durchschnittliche Laufzeit wie für das diskret reduzierte Indoor-Szenario

bei einer Auflösung von 100K erfordert  $\approx 0.01$  Minuten für die Berechnung der Normalen,  $\approx 0.05$  Minuten für die SI und  $\approx 0.8$  Minuten zur Berechnung und Verwerfung der Korrespondenzen.

### 11.2.8. Stabilität

Wie die Parameterstudie (siehe Abschnitte 11.2.1, 11.2.2, 11.2.4, 11.2.5) bereits gezeigt hat, ist es prinzipiell möglich ein Parameterset zu definieren, mit dem sich die verschiedenen Szenarien registrieren lassen. Auch weisen die einzelnen Parameter ein allgemein robustes Verhalten gegenüber Änderungen des Wertebereiches auf. Über die verschiedenen Auflösungen hinweg war es zumindest für höhere Auflösungen möglich die einzelnen Szenarien mit konstanten Parameterwerten erfolgreich zueinander auszurichten.

Für sehr geringe Auflösungen war keine korrekte Registrierung möglich (siehe Abbildung 11.12). Auch für höhere diskrete Auflösungen konnte nahezu kein korrekte Registrierungsergebnis erzielt werden, auch wenn der Algorithmus im Sinne des Konvergenzkriterium erfolgreich war. Mit Hilfe des räumlichen Sub-Sampling war eine korrekte Ausrichtung der Punktwolken für alle Szenarien ab einer Auflösung von 100K möglich.

Betrachtet man die Registrierungsergebnisse über die einzelnen Szenarien hinweg, können alle Szenarien mit einem relativ ähnlichen Registrierungsfehler zueinander ausgerichtet werden; allerdings nur auf räumlich reduzierten Punktdaten. Unterschiede in den Laufzeiten zwischen den einzelnen Szenarien scheinen eher gering zu sein; ein eindeutiger Beleg dafür existiert jedoch nicht.

## 11.3. Iterative Closest Point

Der ICP kann Vorabinformation in Form einer initialen Ausrichtung berücksichtigen. Für die nachfolgenden Untersuchungen wurde die initiale Ausrichtung  $\Gamma_{(3,3,1)^T}^{-10}$  (siehe Abschnitt 5.3) verwendet. Sie zeichnet sich durch die Genauigkeiten der in Abschnitt 2.1.2 vorgestellten Sensoren aus. Die daraus resultierenden initialen Fehler sind in Anhang C präsentiert.

### 11.3.1. Fehlergrenzwert

Der Fehlergrenzwert hat im wesentlichen Einfluss auf die Genauigkeit des Registrierungsergebnisses. Dies scheint durch seine Abhängigkeit von den Punkt-zu-Punkt-Korrespondenzen, die zur Bestimmung der Transformation verwendet werden, einher zu gehen. Angenommen die Korrespondenzpunkte können gut zueinander ausgerichtet werden und der Fehlergrenzwert ist gering, dann würde ein große Wahl, das Konvergieren des Algorithmus trotz großer Entfernung der korrespondierenden Punkte erlauben.

Betrachtet man Abbildung 11.14 fällt auf, dass bei diskret reduzierten Punktdaten das Registrierungsergebnis schon bei deutlich kleineren Fehlergrenzwerten negativ beeinflusst

wird als dies bei räumlichem Sub-Sampling der Fall ist. Ursache davon scheint eine generell ungenauere Registrierung auf räumlich reduzierten Daten zu sein.

Betrachtet man die Punktdichte der unterschiedlichen Sub-Sampling-Strategien kann gesagt werden, dass diese beim räumlichen Sub-Sampling geringer ist als bei diskretem (gleich Punktzahl bei höherer Ausdehnung). Dies bedeutet auch, dass die Anzahl der im Überlappungsbereich enthaltenen Punkte (Korrespondenzpunkte) ebenfalls geringer ist. Zusätzlich entsprechen mittels räumlichem Sub-Sampling verarbeitete Punktwolken, einer Approximation der ursprünglichen Punktdaten, wodurch selbst bei einer hypothetisch perfekten Registrierung ohne Sensorfehler Spannungen entstehen. Diese werden unter dieser Betrachtung auf weniger Punkte verteilt, als dies bei diskret reduzierten Punktwolken der Fall ist, was zu einem höheren Fehler führt.

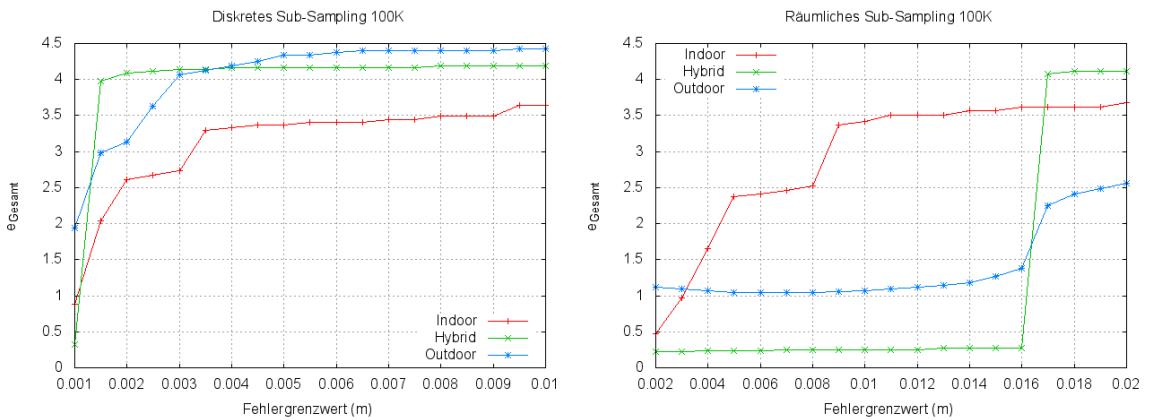


Abbildung 11.14.: ICP Fehlergrenzwert - Der Einfluss des Fehlergrenzwertes auf das Registrierungsergebnis ist bei räumlich reduzierten Punktdaten geringer als bei diskreten.

### 11.3.2. Transformationsgrenzwert

Der Transformationsgrenzwert kann als Forderung an die Stabilität des Registrierungsergebnisses betrachtet werden. Ändert sich die Transformation von einer Iteration zur anderen, haben sich auch die Korrespondenzen geändert. Ändern sich diese nicht, ist auch die resultierende Transformation dieselbe. Als Folge dessen kann die Höhe der Transformationsänderung in Zusammenhang mit der Menge und Unterschiedlichkeit der neu gebildeten Korrespondenzen gebracht werden.

Bei dieser Untersuchung wurden die verbleibenden Abbruchkriterien so gewählt, dass ein Konvergieren nur über den Transformationsgrenzwert möglich war. In den Ergebnissen (siehe Abbildung 11.15) ist ein szenenübergreifend einheitliches Verhalten zu sehen. Eine sehr geringe Erhöhung des Transformationsgrenzwertes führt dabei zu einer enormen Verschlechterung des Registrierungsergebnisses. Bei räumlich reduzierten Punktdaten ist der Schwellwert, bei welchem dieses Verhalten zu beobachten ist, höher.

Geht man davon aus, dass zum Erreichen des globalen Minimums der Fehlerfunktion viel kleine Schritte benötigt werden (breites globales Minimum (siehe Abbildung 11.16)), führt eine zu große Wahl des Transformationsgrenzwertes dazu, dass dieses nicht erreicht werden kann. Eine zu kleine Wahl könnte sogar das Übergehen dessen zur Folge haben. Wird wie in Abschnitt 11.3.1 davon ausgegangen, dass räumlich reduzierte Daten aufgrund der Punktapproximation generell höhere Fehler verursachen können, liegt der Schluss nahe, dass das globale Minimum der resultierenden Fehlerfunktion weniger tief aber sein Pik dafür steiler ist. Ein hoher Transformationsgrenzwert könnte dann trotzdem gute Ergebnisse liefern.

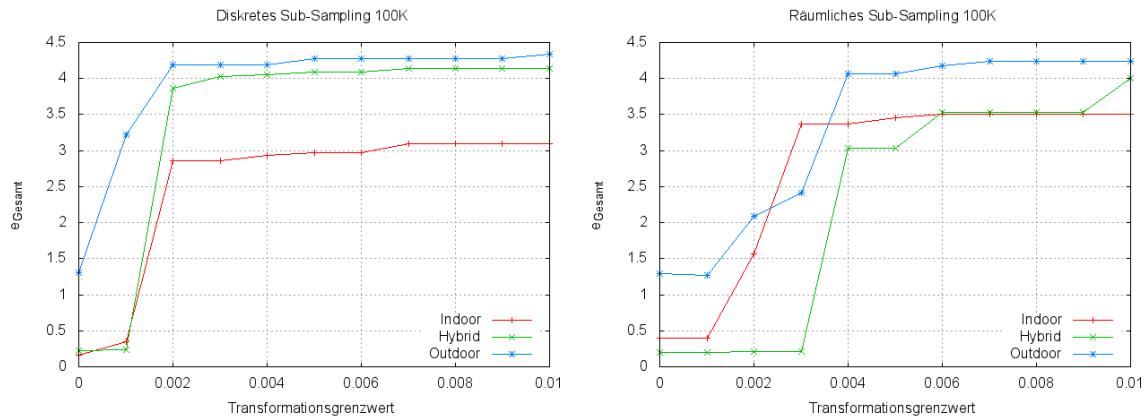


Abbildung 11.15.: ICP Transformationsgrenzwert - Das Unterbinden kleiner Transformationsänderungen verschlechtert das Registrierungsergebnis. Auf diskret reduzierten Punktdaten macht sich dies schneller bemerkbar als auf räumlich reduzierten.

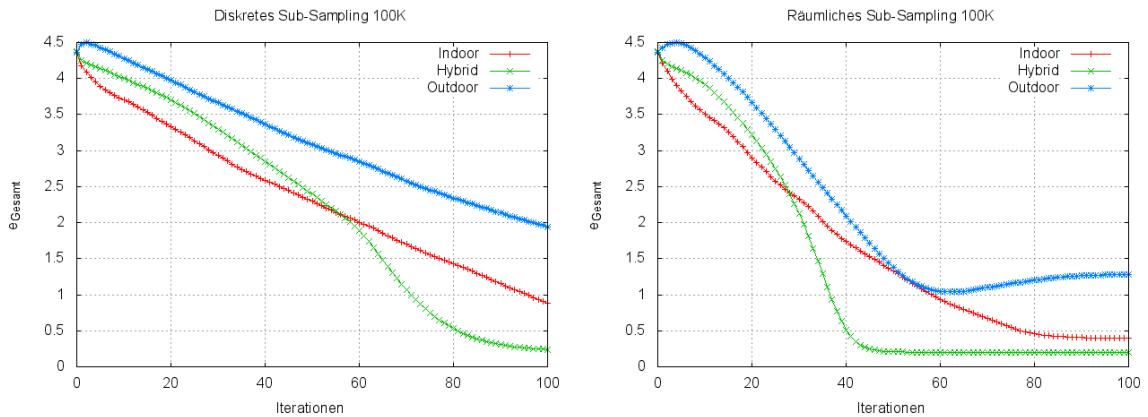
### 11.3.3. Iterationen

Die Anzahl maximaler Iterationen legt fest, wie viele Versuche zum Auffinden einer dem Transformations- und Fehlergrenzwert entsprechenden Lösung zur Verfügung stehen. Verhindert werden kann dadurch ein endloses Iterieren aufgrund alternierender Korrespondenzen bei schlechtem Fehlerwert.

Bei dieser Untersuchung wurden die verbleibenden Abbruchkriterien so gewählt, dass ein Konvergieren nur durch das Erreichen der maximalen Anzahl zulässiger Iterationen möglich war (siehe Abbildung 11.16). Auf diskret reduzierten Punktwolken ist dabei ein lineares Konvergieren in nahezu allen Szenarien zu beobachten; räumlich reduzierte Daten konvergieren im Vergleich dazu schneller.

Angenommen die geringere Dichte nach einem räumlichen Sub-Sampling sorgt dafür, dass weniger Korrespondenzen in jeder Iteration ausgebildet werden können, dann existieren auch weniger Kandidaten für falsche Punktkorrespondenzen und ein schnelleres Konvergieren ist die Folge.

Besonders auffällig ist das Divergieren der Outdoor-Szene auf räumliche reduzierten Punkt wolken nach dem Erreichen eines Minimums. Grund dafür ist, dass die verbleibenden Abbruchkriterien nicht ausreichend strikt gewählt sind und somit über das Minimum hinweg iteriert wird. Ist in einem solchen Falle kein Divergieren zu beobachten, kann von einem stabilen Minimum ausgegangen werden und der Algorithmus alterniert um dieses.



**Abbildung 11.16.: ICP Iterationen** - Bei gleichverteilten Punktdaten, wie durch räumliches Sub-Sampling erzeugt, kann eine zu hohe Anzahl von Iterationen eine Verschlechterung des Ergebnisses zur Folge haben. Vermieden werden kann dies durch eine striktere Wahl der verbleibenden Abbruchbedingungen.

### 11.3.4. Korrespondenzgrenzwert

Der Korrespondenzgrenzwert bestimmt im Allgemeinen die Schrittweite einer Iteration. Seine Wahl steht dadurch in Zusammenhang mit der initialen Ausrichtung. Ist er im Verhältnis zu dieser zu klein gewählt, können (speziell in den ersten Iterationen wichtig) keine korrekten Korrespondenzen ausgebildet werden. Dadurch besteht die Gefahr in eine beliebige Richtung zu iterieren. Eine zu große Wahl kann dazu führen, dass in jeder Iteration (speziell in der letzten Iteration problematisch) fehlerhafte Korrespondenzen ausgebildet werden, die im Allgemeinen einen negativen Einfluss auf das Registrierungsergebnis haben.

Belegt sei dies mit den in Abbildung 11.17 vorgestellten Ergebnissen. Zu sehen ist, dass der minimale Korrespondenzgrenzwert welcher zu einer erfolgreichen Registrierung führt bei allen Szenarien nahezu identisch ist. Auf räumlich reduzierten Daten ist bei einem zu großen Korrespondenzgrenzwert ein Divergieren der Indoor- und Outdoor-Szene zu sehen.

Abhängig von der Szene und Höhe des Korrespondenzgrenzwertes besteht die Möglichkeit, dass selbst beim Erreichen des globalen Minimums der Fehlerfunktion falsche Korrespondenzen gebildet werden. Bricht der Algorithmus nicht ab (zu strikte Abbruchkriterien) kann dadurch ein Divergieren erfolgen. Homogen verteilte Punkte neigen offensichtlich aufgrund des bereits thematisierten generellen Fehlers durch die Approximation von Daten (siehe Abschnitt 11.3.3) eher dazu, die Abbruchkriterien zu verfehlten.

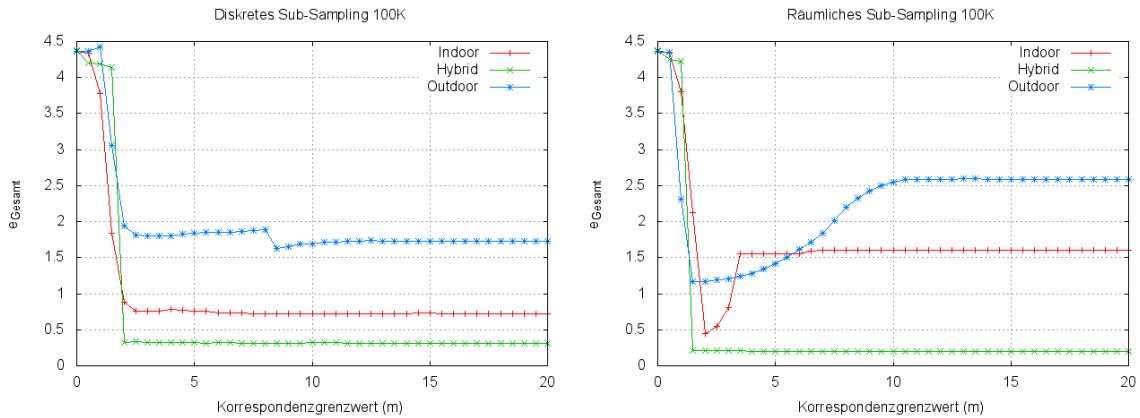


Abbildung 11.17.: ICP Korrespondenzgrenzwert - Zu hohe Korrespondenzgrenzwerte können abhängig von der Szene auf räumlich reduzierten Punktdaten zu einem Divergieren des Algorithmus führen.

### 11.3.5. Konvergenz

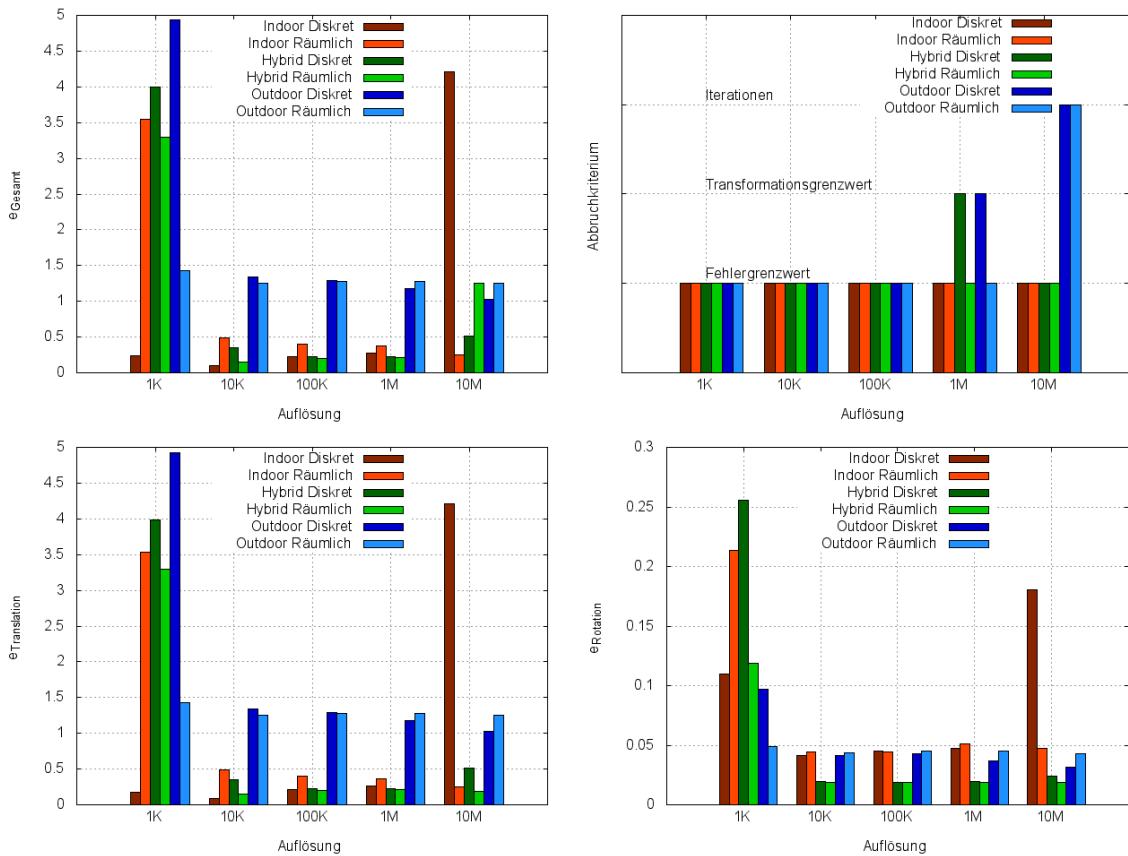
Wie in Anhang (siehe Anhang C) präsentiert kann der Algorithmus im Sinne der definierten Konvergenzanforderung, bei der initialen Ausrichtung  $\Gamma_{(3.0,3.0,1.0)^T}^{-10}$  als konvergiert betrachtet werden, wenn der Fehler  $e_{Gesamt}$  kleiner als 4,36 ist. Für die hier durchgeführte Analyse wurden Parametersets für die einzelnen Szenarien und Sub-Sampling-Strategien verwendet, die sich durch die vorhergehende Parameterstudie bei einer Auflösung von 100K als sinnvoll erwiesen haben.

Betrachtet man den Gesamtfehler der Registrierung (siehe Abbildung 11.18) ist festzustellen, dass der Algorithmus auf allen Auflösungen und über alle Szenarien hinweg konvergiert. Einige Ausnahme bildet die Outdoor-Szene bei einer diskreten Auflösung von 1K. Dem durchweg eher schlechten Registrierungsergebnis bei dieser Auflösung kann nachgesagt werden, dass generell zu wenig Struktur für eine gute Registrierung vorhanden ist. Widerlegt wird dies durch ein sehr gutes Registrierungsergebnis der Indoor-Szene unter Verwendung einer diskreten Reduktion der Punktwolken. Offensichtlich sind selbst bei dieser Auflösung die sehr nahe am Laserscanner liegenden Wände und Decken detailliert genug vorhanden um die richtige Richtung vorzugeben.

Bei einer Auflösung von 10M diskreter Reduktion sind vermutlich die eben genannten Strukturen ausschlaggebend für ein sehr schlechtes Registrierungsergebnis (siehe Abbildung 11.18). Der Abbruch durch den Fehlergrenzwert und der verhältnismäßig hohe Rotationsfehler lassen den Schluss zu, dass durch die hohe Punktanzahl ein lokales Minimum erschaffen wurde, welches tief genug für einen Abbruch war. Der vergleichsweise hohe Rotationsfehler lässt dabei eine symmetrische Lösung vermuten.

Für die anderen Szenarien kann durch räumliches Sub-Sampling das Registrierungsergebnis verbessert werden. Mit zunehmender Auflösung und der daraus resultierenden Punktdichte ändert sich dies zu Gunsten des diskreten Sub-Sampling. Vermutlich kommt dabei die

## 11. Analyse



**Abbildung 11.18.: ICP Konvergenz** - Abgebildet ist der Gesamt-, Translations- und Rotationsfehler nach einer Registrierung der unterschiedlichen Szenarien auf verschiedenen, räumlichen und diskreten Auflösungen. Außerdem ist der Grund des Abbruchs festgehalten. Als initiale Ausrichtung wurde für alle Szenarien  $\Gamma_{(3.0,3.0,1.0)^T}$  verwendet; der initiale Gesamtfehler liegt daher bei 4,36 (siehe Anhang C).

Ungenauigkeit des räumlichen Sub-Sampling durch die Approximation der Punktdaten zum Tragen.

Das beste Registrierungsergebnis wurde für das Indoor-Szenario bei einer Auflösung von 10K diskretem Sub-Sampling erreicht. Dabei betrug  $\bar{e}_{Translation} = (-0.07, -0.04, 0.00)$  m und  $\bar{e}_{Rotation} = (0.04, 0.04, 0.00)$  m.

### 11.3.6. Effizienz

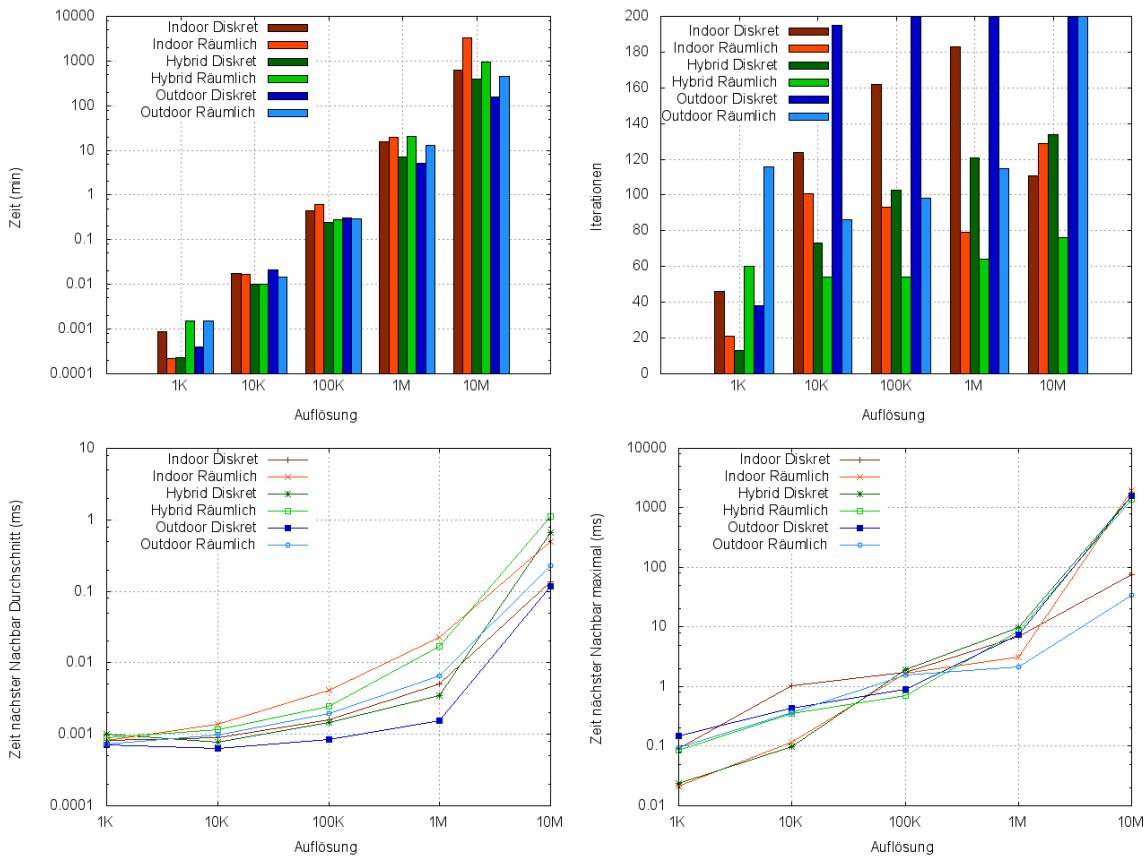
Die Laufzeit des ICP ist im Wesentlichen von der Größe der Punktwolken und der Anzahl benötigter Iterationen abhängig. Grund dafür ist, dass in jeder Iteration der nächste Nachbar eines Punktes gesucht werden muss. Besitzt die Zielpunktwolke  $n$  und die Datenpunktwolke  $k$  Punkte, kann mittels KD-Tree die Komplexität der Suche von  $\mathcal{O}(n \cdot k)$  auf  $\mathcal{O}(n \cdot \log(k))$  reduziert werden (siehe Abschnitt 2.3.3). Nach jeder Iteration müssen die Punkte der Datenpunktwolke erneut in diesen eingefügt und gelöscht werden, was einen zusätzlichen

Aufwand erfordert. Die resultierenden Laufzeiten sind in Abbildung 11.19 einzusehen. Sie basieren auf den im vorhergehenden Abschnitt durchgeföhrten Untersuchungen.

Festzustellen ist, dass bei räumlich reduzierten Punktwolken im Allgemeinen eine geringere Anzahl von Iterationen notwendig ist, um ein Abbruchkriterium zu erfüllen. Ursache könnte dabei sein, dass die gleichverteilte Repräsentation der Punktwolken größere Schrittweiten pro Iteration erlauben.

Weiterhin fällt auf, dass die Laufzeiten auf räumlich reduzierten Daten, trotz einer geringeren Anzahl von Iterationen, nahezu identisch zu den auf diskret reduzierten Punktwolken sind. Untersuchungen haben gezeigt, dass die Suche des nächsten Nachbarn, bei der hier verwendeten Variante des KD-Tree (siehe Abschnitt 2.3.3), für räumlich reduzierte Punktwolken mehr Zeit benötigt als dies für diskret reduzierte der Fall ist. Verursacht wird dies durch einen nicht balancierten KD-Tree nach einem räumlichen Sub-Sampling.

Abschließend kann außerdem gesagt werden, dass ein Registrieren ohne die Reduktion der Punktwolken nicht in angemessener Zeit möglich ist.



**Abbildung 11.19.: ICP Effizienz -** Oben Links: Laufzeit des ICP auf unterschiedlich reduzierten Punktwolken. Oben Rechts: Anzahl der zum Konvergieren benötigter Iterationen. Unten Links\Rechts: Durchschnittliche und maximale Laufzeit bei der Suche des nächsten Nachbarn mit den verwendeten KD-Trees auf diskret und räumlich reduzierten Punktwolken.

### 11.3.7. Stabilität

Wie die in Abschnitt 11.3.5 durchgeführten Untersuchungen gezeigt haben ist es möglich ein Parameterset zu definieren, mit welchem über verschiedene Auflösungen hinweg ein akzeptables Ergebnis erzielt werden kann. Außerdem konnte ein Parameterset definiert werden, mit welchem die drei Testszenarien, sowohl diskret als auch räumlich reduziert, registriert werden konnten.

Aus der bereits vorgenommenen Untersuchung der Parameter ist ersichtlich, dass eine Veränderung der Parameter auch das Registrierungsergebnis beeinflusst. Größere Toleranzbereiche können im Allgemeinen durch räumliches Sub-Sampling erzielt werden. Grund dafür ist offensichtlich die homogene Repräsentation einer Punktwolke, welche bei jeder Auflösung die Struktur der gesamten Szene erhält. Besonders auffällig bleibt dennoch, dass bereits sehr kleine Änderungen der Parameterwerte, sehr große Sprünge im Ergebnis verursachen. Nichts desto trotz ist es möglich ein Parameterset zu definieren, welches für alle Szenarien ein akzeptables Ergebnis liefert.

Wird als Stabilitätskriterium die Auflösung verwendet kann festgestellt werden, dass ihre Veränderung nur sehr geringen Einfluss auf das Registrierungsergebnis hat. Stabile Ergebnisse konnten für alle Auflösungen außer 1K erzielt werden. (siehe Abbildung 11.18).

Betrachtet man die Sub-Sampling-Strategien separat, zeigt sich ein stabiles Verhalten (siehe Abbildung 11.18). So wurde unabhängig von diesen stets ein sehr ähnliches Ergebnis erzielt. Während die Genauigkeit des Registrierungsergebnisses mit steigender diskreter Auflösung steigt, bleibt dieses bei einem räumlichen Sub-Sampling konstant. Ausnahme bildet dabei, wie bereits thematisiert, die Indoor-Szene bei einer diskreten Auflösung von 1K.

Die unterschiedlichen Testszenarien konnten alle erfolgreich registriert werden (siehe Abbildung 11.18). Das Indoor-Szenario erzielte dabei die besten Ergebnisse. Ursache sind die durch Wände und Decken vorhandenen starken Strukturen. Sie bieten dem Algorithmus gute Anhaltspunkte um in jeder Iteration die richtige Richtung zu erfassen. Das Hybrid-Szenario bietet ähnlich markante Bereiche. Der zudem recht große Überlappungsbereich erlaubt eine genaue Registrierung. Relativ ungenau wurde das Outdoor-Szenario registriert. Grund dafür sind offensichtlich die sehr ähnlichen und ineinander übergehenden Konturen.

Werden Translations- und Rotationsfehler genauer untersucht fällt auf, dass deren Anteile unabhängig von Auflösung, Sub-Sampling-Strategie und Szenario sind (siehe Abbildung 11.18). Rotation und Translation können nach diesen Fakten in selbem Maße stabil gelöst werden. Eine Betrachtung des Rotationsfehler  $\vec{e}_{Rotation}$  hat außerdem ergeben, dass die durch das Inklinometer festgelegten Rotationen um X- und Y-Achse stets beibehalten wurden.

## 11.4. Normal-Distribution Transformation

Auch der NDT ist in der Lage, Vorabinformation in Form einer initialen Ausrichtung zu berücksichtigen. Wie beim ICP wurde für die nachfolgenden Untersuchungen die, sich

durch die Genauigkeiten der in Abschnitt 2.1.2 vorgestellten Sensoren auszeichnende, initiale Ausrichtung  $\Gamma_{(3,3,1)^T}^{-10}$  (siehe Abschnitt 5.3) verwendet. Ihre initialen Fehler sind in Anhang C festgehalten.

#### 11.4.1. Iterationen

Die Anzahl maximaler Iterationen legt fest, wie viele Versuche zum Auffinden einer dem Transformationsgrenzwert entsprechenden Lösung zur Verfügung stehen. Verhindert wird dadurch ein endloses Suchen nach einer Lösung, wenn diese nicht gefunden werden kann. Bei den hierzu durchgeführten Untersuchungen wurde der Transformationsgrenzwert so gewählt, dass ein Abbruch des Algorithmus nur durch das Durchführen der maximalen Anzahl von Iterationen erfolgt (siehe Abbildung 11.20).

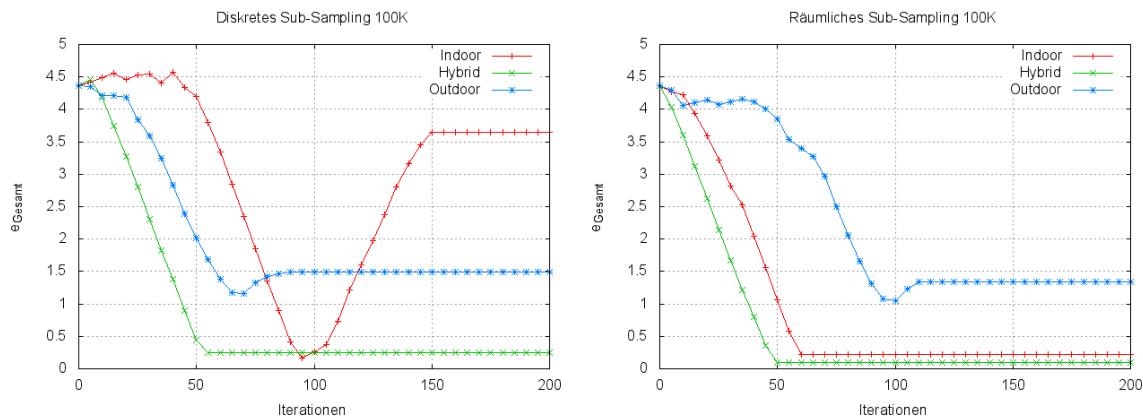


Abbildung 11.20.: NDT Iterationen - Registrierungsergebnis nach unterschiedlichen Iterationsschritten. Der Transformationsgrenzwert wurde dabei so gewählt, dass ein Abbruch durch das Erreichen der maximal zulässigen Iterationen erfolgt. Räumlich reduzierte Daten unterbinden das Divergieren der Indoor-Szene.

Auf diskret reduzierten Punktdaten ist dabei festzustellen, dass eine zu hohe Anzahl an Iterationen, wie im Falle des Indoor- und Outdoor-Szenario zu einem schlechteren Registrierungsergebnis führt. Anders ausgedrückt kann ein Übergehen der korrekten Lösung erfolgen, wenn diese nicht durch den Transformationsgrenzwert erkannt wird. Die Hybrid-Szene zeigt dieses Verhalten nicht. Vermutlich führen die darin enthaltenen, klar voneinander getrennten Strukturen, zu einem sehr tiefen und breitem globalen Minimum der Fehlerfunktion welches nicht verlassen werden kann. Große und flache Strukturen wie in Indoor- und Outdoor-Szene können symmetrische Minima bilden, die dann im Allgemeinen weniger breit und ausgeprägt sind und eventuell übersprungen werden. Indikator dafür ist der selbst bei divergenterem Verhalten stets niedrige Wert der NDT Fehlerfunktion.

Eine Registrierung auf räumlich reduzierten Daten behebt das Divergenz-Problem für die Indoor-Szenerie und führt sogar zu einem Konvergieren bei einer geringeren Anzahl von Iterationen. Das Outdoor-Szenario hingegen benötigt mehr Iteration als bei diskret reduzierten Punktwolken und konvergiert nicht mehr linear. Durch räumliches Sub-Sampling

wird eine Szene homogen und vollständig repräsentiert. Dies bedeutet, dass ein Objekt abhängig von seiner Größe durch entsprechend viele Punkte repräsentiert wird. Dadurch sind in einer Szene nahezu immer auch symmetriebrechende Objekte vorzufinden. Im Falle des Outdoor-Szenario verursachen diese offensichtlich, dass der hier korrekte Einfluss symmetrischer Objekte zu starken Einfluss auf das Registrierungsergebnis nimmt.

### 11.4.2. Transformationsgrenzwert

Der Transformationsgrenzwert kann wie beim ICP als Stabilitätskriterium interpretiert werden. Ändert sich die Transformation nur noch sehr geringfügig, wurde ein Minimum oder Plateau der Fehlerfunktion erreicht. Ein weiteres Optimieren führt dann zu keinem besseren Ergebnis oder sogar zu einem Überspringen des Minimums (schlechteres Ergebnis). Für die hier durchgeführten Untersuchungen wurde die Anzahl durchzuführender Iterationen so gewählt, dass ein Konvergieren nur durch das Erreichen des Transformationsgrenzwertes erfolgen kann (siehe Abbildung 11.21).

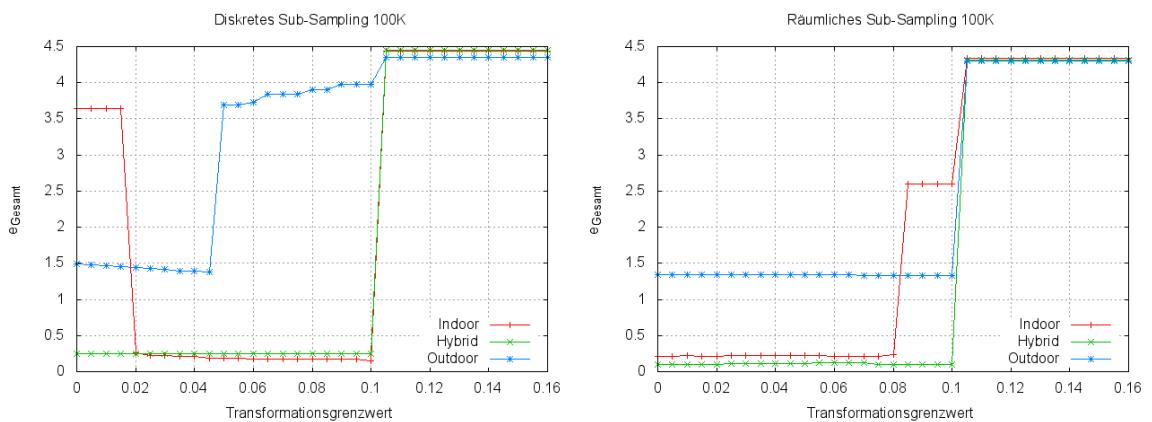


Abbildung 11.21.: NDT Transformationsgrenzwert - Ein zu kleiner Transformationsgrenzwert kann zu einem Überspringen von Minima führen. Die Anzahl durchzuführender Iterationen wurden bei dieser Untersuchung so gewählt, dass ein Abbruch durch den Transformationsgrenzwert erfolgt.

Auf räumlich reduzierten Punktdaten ist zu sehen, dass die Indoor-Szene bei kleinen Transformationsgrenzwerten ein sehr schlechtes Registrierungsergebnis aufweist. Dieses verbessert sich jedoch sprunghaft, wenn der Transformationsgrenzwert erhöht wird. Die deutlich höhere Anzahl durchgeföhrter Iterationen bei niedrigen Transformationsgrenzwerten weisen auf ein Überspringen des globalen Minimums hin. Die beiden anderen Szenen verhalten sich wenig auffällig.

Räumliches Sub-Sampling verhindert das Überspringen des globalen Minimums der Indoor-Szene. Vermutlich erzeugt dieses ein breiteres globales Minimum, welches selbst bei kleinen Transformationsgrenzwerten nicht übersprungen werden kann.

Ursache des starken Sprunges bei einem Transformationsgrenzwert von 0.1 ist die Schrittweite des Moré-Thuente Verfahrens. Sie definiert eine obere Schranke. Genauer wird diese jedoch

nicht diskutiert. Der interessierte Leser sei aber auf [MTMp92] und die Implementierung des NDT [Lib12b] verwiesen.

### 11.4.3. Zellgröße

Über die Zellgröße kann gesagt werden, dass sie bestimmt wie detailliert die Punktwolke in der PDF repräsentiert ist. Kleine Zellen führen zu einem hohen Einfluss der einzelnen Punkte auf die PDF, große Zellen hingegen glätten diese. Vergleichbar ist dies mit der Betrachtung feiner bzw. grober Merkmale. Welche Art der Merkmale dabei in der PDF repräsentiert werden sollen ist abhängig von der Beschaffenheit des zu registrierenden Szenario.

Die Ergebnisse der Untersuchung in Abbildung 11.22 zeigen, dass der Einfluss auf das Registrierungsergebnis sehr prägnant ist. Bei diskret reduzierten Daten profilieren sich, auf Grund der detaillierten Repräsentation einzelner Objekte der Szene, kleinere Zellgrößen. Bei räumlich reduzierten Punktwolken führt auch die Verwendung großer Zellgrößen zu guten Ergebnissen. Das Hybrid- und Outdoor-Szenario profitieren dabei am meisten von einer räumlichen Repräsentation bezüglich der Genauigkeit und Stabilität des Parameters. Ursache davon ist offensichtlich, dass die in ihnen vermehrt vorkommenden großen Strukturen, durch das räumliche Sub-Sampling mit einer gleichmäßigeren Punktdichte und somit einer aussagekräftigeren PDF repräsentiert werden. Bei allen Szenarien ist auf räumlich reduzierten Daten außerdem eine geringere Schwankung des Fehlers zu verzeichnen.

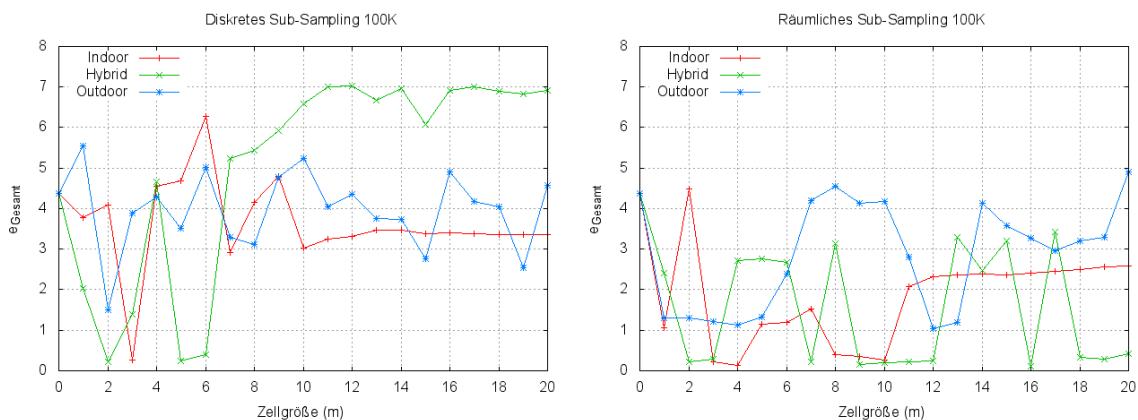


Abbildung 11.22.: NDT Zellgröße - Verhalten des Algorithmus bei unterschiedlichen Zellgrößen in den verschiedenen Szenarien. Bei räumlich reduzierten Punktwolken sind die Fehlerschwankungen geringer.

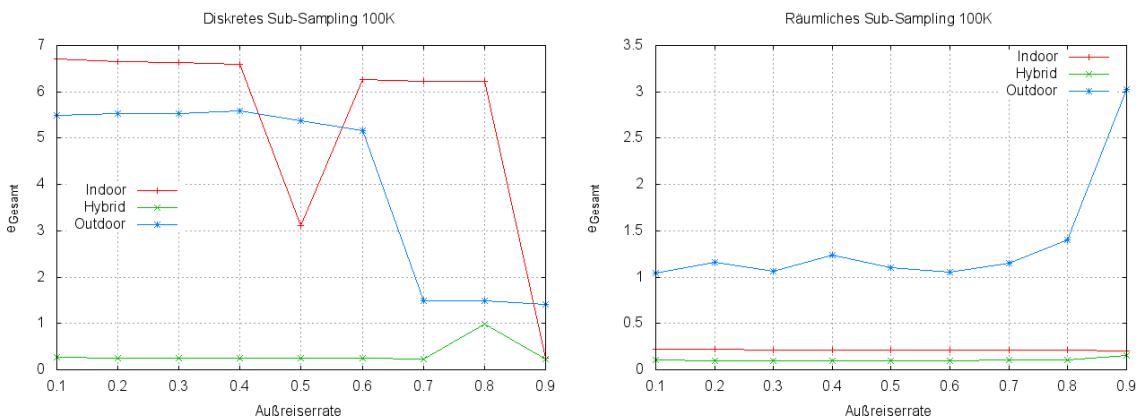
### 11.4.4. Ausreißerrate

Über die Ausreißerrate wird bestimmt, wie stark eine Struktur vertreten sein muss, um Einfluss auf die Fehlerfunktion zu erhalten. Anders ausgedrückt sinkt der Einfluss schwach repräsentierter Strukturen bei einer steigenden Ausreißerrate.

## 11. Analyse

Betrachtet man das Registrierungsergebnis bei den diskret reduzierten Szenarien fällt auf, dass besonders Indoor- und Outdoor-Szene von einer hohen Ausreißerrate profitieren. Sie enthalten demnach Bereiche die eine hohe Wahrscheinlichkeit zur falschen Korrespondenz aufweisen und aufgrund ihres relativ großen Anteils in der Szene eine hohe Ausreißerrate erfordern um „ausgeblendet“ werden zu können. In der Indoor-Szene könnten dies z.B. die an den Flur angrenzenden Zimmer sein. Durch die unterschiedlichen Blickwinkel sind in Ziel- und Datenpunktfolge unterschiedliche, aber nahezu identisch strukturierte Bereiche sichtbar. Für die Outdoor-Szene ist eine solches Objekt durch den Boden gegeben, der für Ziel- und Datenpunktfolge in unterschiedliche Richtungen ausgedehnt ist. Die Hybrid-Szene zeigt nahezu keinen Einfluss auf eine Änderung der Ausreißerrate. In ihr sind aber auch keine größeren Objekte enthalten, die durch Verdeckung in einer der beiden Punktfolgen nicht sichtbar wären.

Auf räumlich reduzierten Daten sind alle Objekte einer Szene abhängig von deren Größe ähnlich dicht repräsentiert. Dadurch müsste der Einfluss einzelner Strukturen auf die Fehlerfunktion nahezu identisch sein. Aufgrund dessen entsteht das beim diskreten Sub-Sampling auftretende Problem der Verdeckung in dieser Form nicht. Wie in der Outdoor-Szene zu sehen, kann eine hohe Ausreißerrate dennoch zu einem schlechteren Registrierungsergebnis führen. Ein plausible Erklärung dafür ist, dass kleine Strukturen, die für eine korrekte Ausrichtung notwendig sind, durch hohe Ausreißerraten zusätzlich an Einfluss verlieren. Diese wären in der Outdoor-Szene durch die zwei freistehenden Felsen gegeben.

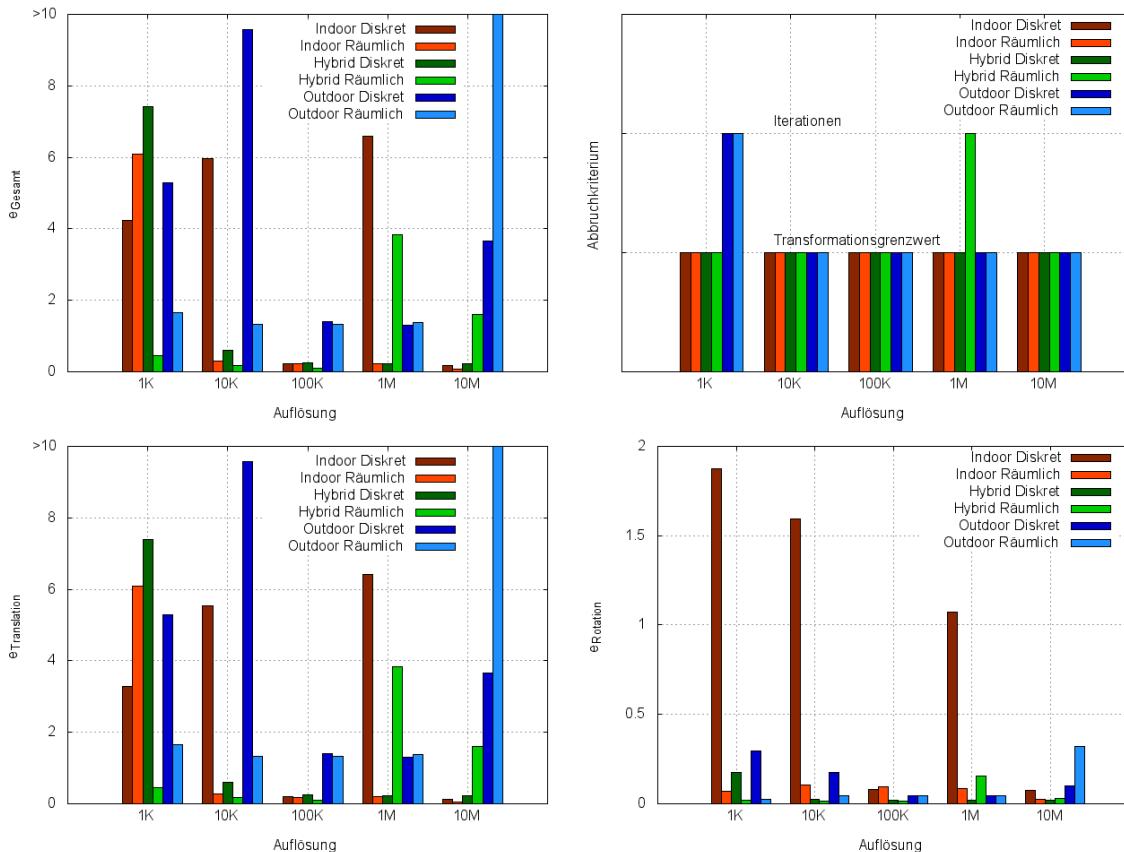


**Abbildung 11.23.: NDT Ausreißerrate** - Eine hohe Ausreißerrate kann den zur korrekten Ausrichtung störenden Objekten den Einfluss auf die Fehlerfunktion entziehen. Bei räumlichem Sub-Sampling kann dies aber auch für richtungsweisende Strukturen erfolgen was einen negativen Einfluss auf das Registrierungsergebnis zur Folge, hat.

### 11.4.5. Konvergenz

Für die hier durchgeführten Registrierungen wurden szenenabhängige Parameter verwendet, die sich durch die vorhergehende Parameterstudie als sinnvoll erwiesen haben. Mit

diesem Parametersatz wurde dann auf verschiedenen diskreten und räumlichen Auflösungen registriert. Die resultierenden Ergebnisse sind in Abbildung 11.24 zusammengetragen. Wie in Anhang (siehe Anhang C) präsentiert kann der Algorithmus im Sinne der Konvergenzanforderung und der hier verwendeten, initialen Ausrichtung  $\Gamma^{-10}_{(3.0,3.0,1.0)^T}$  als konvergiert betrachtet werden, wenn der Fehler  $e_{Gesamt}$  kleiner als 4,36 ist.



**Abbildung 11.24.: NDT Konvergenz** - Abgebildet ist der Gesamt-, Translations- und Rotationsfehler nach einer Registrierung der unterschiedlichen Szenarien auf verschiedenen, räumlichen und diskreten Auflösungen. Außerdem ist der Grund des Abbruchs festgehalten. Als initiale Ausrichtung wurde für alle Szenarien  $\Gamma^{-10}_{(3.0,3.0,1.0)^T}$  verwendet; der initiale Gesamtfehler liegt daher bei 4,36 (siehe Anhang C).

Auf diskret reduzierten Daten lässt sich die Indoor-Szene lediglich für die Auflösungen 100K und 10M registrieren. Eine Betrachtung des Rotationsfehlers der verbleibenden Auflösungen sowie der Abbruch durch das Erreichen des Transformationsgrenzwertes weist darauf hin, dass der Algorithmus offensichtlich in Richtung eines symmetrischen Minimum iteriert hat. Verursacht werden können symmetrische Minima durch Wände, Boden und Decke, welche in dieser Szene bei einem diskreten Sub-Sampling sehr markant repräsentiert werden. Räumliches Sub-Sampling erzeugt eine gleichverteilte Repräsentation der Punktwolken wodurch dieser Effekt offensichtlich verhindert wird und ein Konvergieren auf allen Auflösungen außer 1K möglich macht. Ein Blick auf den Translationsfehler lässt vermuten, dass

## 11. Analyse

nicht genügend Struktur vorhanden ist um eine eindeutige PDF zu erzeugen, da bei dieser geringen Auflösung nahezu ausschließlich Boden und Decke vorhanden ist.

Die Hybrid-Szene zeichnet sich durch viele unterschiedliche, kleine und große Strukturen aus. Dies scheint der Grund zu sein, dass dieses Szenario lediglich bei einer diskreten Auflösung von 1K nicht konvergiert. Ein verhältnismäßig schlechtes Ergebnis wird bei einer räumlichen Auflösung von 1M erzielt. Der Abbruch durch das Iterationskriterium und der hohe Translationsfehler könnte auf ein Alternieren in einem lokalen Minimum zurückgeführt werden.

Sehr durchwachsen sind die Ergebnisse des Outdoor-Szenario. Mit einer diskreten Datenreduktion ist einer Registrierung mit den Auflösungen 100K und 1M und 10M möglich. In den verbleibenden konnte die Translation nicht gelöst werden, was bei einem Abbruch durch den Transformationsgrenzwert wieder ein Indiz für ein lokales Minimum sein könnte. Die sehr ähnlichen Formen der Felsketten in dieser Szene bieten dafür genügend Anhaltspunkte. So scheint dies auch die Ursache einer nicht erfolgreichen Registrierung bei einer räumlichen Auflösung von 10M zu sein. Auf allen verbleibenden räumlichen Auflösung konnte das Outdoor-Szenario registriert werden.

Das beste Ergebnis wurde für die Indoor-Szene bei einer räumlichen Auflösung von 10M erzielt. Dabei wurde eine Genauigkeit von  $\vec{e}_{Translation} = (0.009, -0.058, 0.007)^T$  und  $\vec{e}_{Rotation} = (0.013, 0.002, 0.003)^T$  erreicht.

### 11.4.6. Effizienz

In Abbildung 11.25 ist der Zeitaufwand und die Anzahl der zum Konvergieren benötigter Iterationen des Algorithmus für verschiedene, diskrete und räumliche Auflösungen abgetragen. Diese basieren auf den im vorhergehenden Abschnitt durchgeführten Untersuchungen.

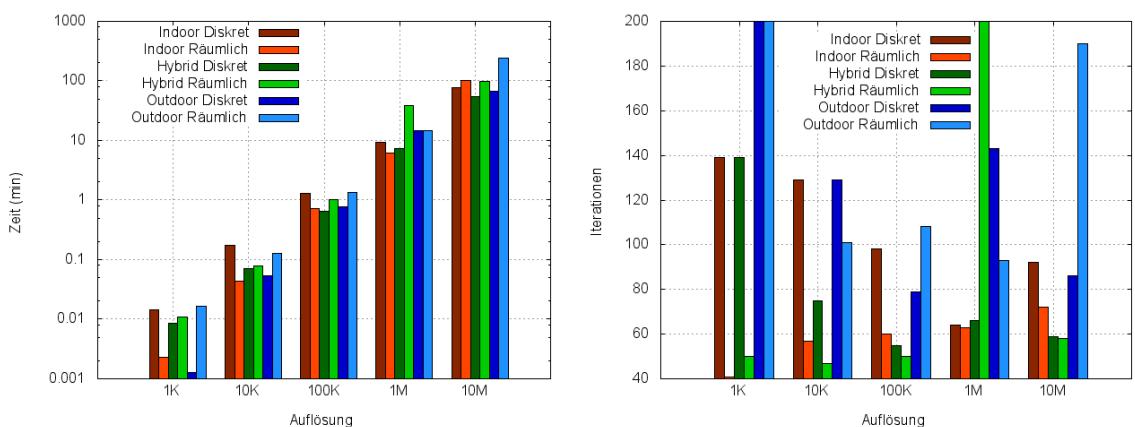


Abbildung 11.25.: NDT Effizienz - Links: Laufzeit des NDT auf unterschiedlich reduzierten Punktwolken. Rechts: Anzahl der zum Konvergieren benötigter Iterationen.

Zeitaufwändigste Operation des Algorithmus ist im Wesentlichen die Berechnung der Fehlerfunktion in jeder Iteration. Hierzu muss für jeden Punkt der Datenpunktswolke bestimmt werden, in welche Zelle der Zielpunktswolke er transformiert wurde. Anschließend muss die gesamte Zelle durchlaufen und die Fehlerfunktion berechnet werden. Der Berechnungsaufwand dieser Operation ist linear mit der Anzahl der Punkte in jeder Zelle und somit auch für die gesamte Punktswolke.

Vergleicht man die Anzahl durchgeführter Iterationen mit der Laufzeit fällt auf, dass räumlich reduzierte Daten bei einer meist geringeren Anzahl von Iterationen mehr Zeit benötigen. Ursache davon ist die räumliche Repräsentation der Zielpunktswolke als Voxel-Grid. Die hier verwendete Variante des Voxel-Grids [Lib12k] verarbeitet Suchanfragen über spezielle KD-Trees [Lib12i] die wie bereits in Kapitel (siehe Abschnitt 11.3.6) thematisiert auf räumlich reduzierten Punktdaten im Allgemeinen längere Laufzeiten aufweisen. Außerdem enthält der Voxel-Grid einer räumlich reduzierten Punktswolke meist mehr Zellen, als dies bei einer diskreten Reduktion der Fall ist. Dadurch entsteht ein nicht unbedingt zu vernachlässigender Mehraufwand bei der Verwaltung des Voxel-Grids, der zusätzlich Einfluss auf die Laufzeit nimmt.

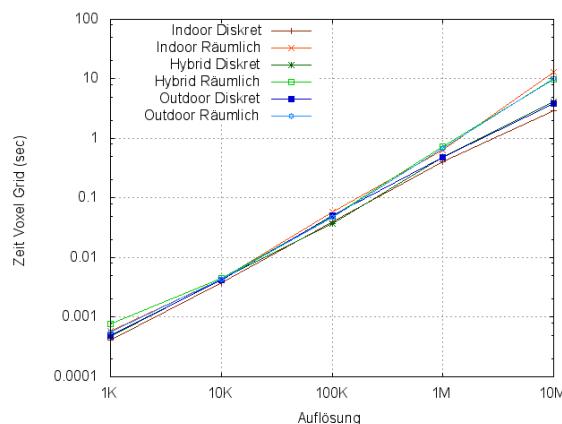


Abbildung 11.26.: NDT Effizienz - Abgebildet ist die zum Aufbau des Voxel-Grids benötigte Zeit für unterschiedliche Auflösungen. Für räumlich reduzierte Punktswolken wird mehr Zeit benötigt als für diskret reduzierte.

Nicht berücksichtigt wurde bisher der Zeitaufwand, welcher durch den Aufbau des Voxel-Grids selbst entsteht. Wie in Abbildung 11.26 zu sehen steigt dieser mit der Punktanzahl. Auch beim Aufbau des Voxel-Grids sind KD-Trees beteiligt, wodurch auch dabei eine zeitliche Abhängigkeit von der Sub-Sampling-Strategie zu erkennen ist. Der maximale Zeitaufwand bei 10M Punkten räumlich reduzierter Punktswolken beläuft sich dabei auf  $\approx 13$  Sekunden. Bei der damit verbundenen Laufzeit von 100.70 Minuten ist diese jedoch vernachlässigbar gering.

Generell kann gesagt werden, dass bei einer minimalen Laufzeit von 54.67 Minuten auf 10M Punkten und einem Registrierungsfehler  $e_{Gesamt}$  von 0.22, der NDT für ein punktbasiertes Verfahren äußerst performant ist.

#### 11.4.7. Stabilität

Die in Abschnitt 11.4.5 durchgeführte Untersuchung zeigt, dass es möglich ist ein Parameter-  
set für einzelne Szenen zu definieren, mit welchem eine Registrierung über verschiedene  
Auflösungen hinweg erfolgen kann. Unterschiedliche Parameter müssen dabei nicht nur  
für die einzelnen Szenen ermittelt werden, sondern auch für die unterschiedlichen Sub-  
Sampling-Methoden wie aus den Parameterstudien in den Abschnitten 11.4.1 - 11.4.4 bereits  
hervorgeht.

Betrachtet man die Stabilität des Registrierungsergebnisses über verschiedene Auflösungen  
hinweg (siehe Abbildung 11.24) wird deutlich, dass eine Repräsentation der Punkt wolken  
mit 1K für diesen Algorithmus im Allgemeinen zu gering ist. Für die restlichen Auflösungen  
konnten nahezu ausnahmslos akzeptable Ergebnisse im Sinne des Konvergenzkriteriums  
erreicht werden.

Die Ergebnisse in Abbildung 11.24 zeigen, dass auf räumlich reduzierten Punktdaten durch-  
weg ein besseres und über verschiedene Auflösungen hinweg stabileres Registrierungsergeb-  
nis erzielt werden konnte. Die homogene Repräsentation einer Szene führt offensichtlich zu  
einer aussagekräftigen und eindeutigen PDF. Diskretes Sub-Sampling hingegen verursacht  
häufig Probleme, da auf verschiedenen Auflösungen, unterschiedliche Merkmale stärker  
bzw. schwächer vorhanden sind.

In nahezu allen Untersuchungen konnte das Hybrid-Szenario, mit einer besseren Lösung als  
der initialen Ausrichtung registriert werden. Mittels räumlichem Sub-Sampling lässt sich  
diese Aussage auch für die anderen beiden Szenarien treffen. Die Symmetrie der Indoor-  
Szene sowie die sehr ähnlichen Objektformen der Outdoor-Szene können häufig nicht richtig  
verarbeitet werden (siehe Abbildung 11.24). Bemerkbar macht sich dies in sehr großen  
Translations- und Rotationsfehlern. Die durch das Inklinometer festgelegte X-Y-Ebene wurde  
jedoch stets beibehalten und nicht verändert.

## 12. Fazit und Ausblick

ZUM Vergleich der einzelnen Algorithmen untereinander wurden die jeweils besten Registrierungsergebnisse (siehe Abbildung 12.2), die kürzesten Laufzeiten (siehe Abbildung 12.1) sowie die dafür notwendigen Auflösungen betrachtet. Die kürzesten Laufzeiten beziehen sich dabei auf die kleinsten Auflösungen für die ein Konvergieren im Sinne des Konvergenzkriterium (siehe Abschnitt 1.2) erfolgte.

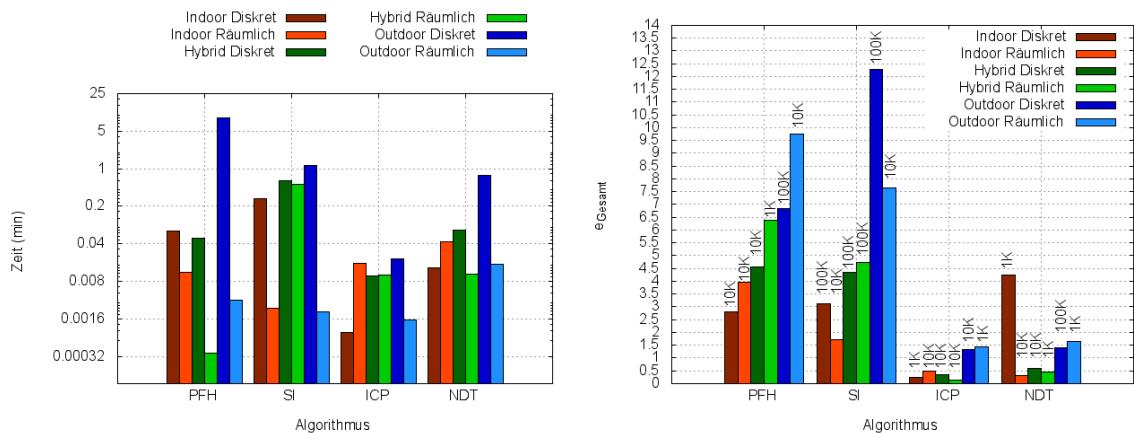


Abbildung 12.1.: Übersicht Laufzeit - Abgebildet sind die kürzesten Laufzeiten (links) der einzelnen Algorithmen, mit denen ein Registrierungsergebnis (rechts) auf den entsprechend reduzierten Szenarien erzielt werden konnte, welches der Konvergenzanforderung (siehe Abschnitt 1.2) entspricht. Die zugehörigen Auflösungen sind über den Registrierungsfehler abgetragen.

Betrachtet man die Laufzeiten der einzelnen Algorithmen kann gesagt werden, dass für kein Verfahren ein Registrieren ohne eine Reduktion der Punktwolken in praktikabler Zeit möglich ist. Über alle Szenarien hinweg zeigt sich der ICP als schnellster Algorithmus und erzielt dabei sogar die besten Registrierungsergebnisse. Mit einer nur unwesentlich längeren und ähnlich genauen Registrierung benötigt der NDT deutlich weniger Zeit als PFH und SI. Die SI weisen dabei die längsten Laufzeiten auf, was mit der höheren Auflösung zusammenhängt, die von ihnen benötigt wird, um das Konvergenzkriterium zu erfüllen. Dabei erreichen sie Genauigkeiten, welche auch von den PFH auf niedrigeren Auflösungen erreicht werden.

Starke Schwankungen bei der Genauigkeit sowie der Laufzeit sind bei beiden Verfahren für die einzelnen Szenarien zu beobachten. Ein räumliches Sub-Sampling führt in allen Szenarien zu einer Reduktion der benötigten Zeit. Besonders stabil in der Laufzeit, dem Registrierungsergebnis und der dafür notwendigen Auflösung zeigt sich der ICP und nur unwesentlich schlechter der NDT. Beide Verfahren profitieren besonders bei geringen

## 12. Fazit und Ausblick

Auflösungen davon, dass keine zeitaufwendige Extraktion von Merkmalen notwendig ist. Räumliches Sub-Sampling führt bei ihnen nicht immer zu einer kürzeren Laufzeit.

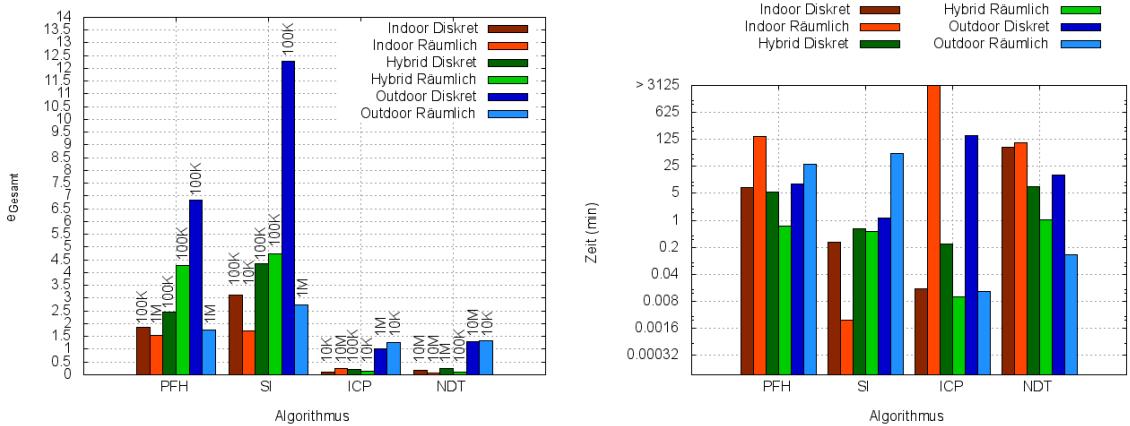


Abbildung 12.2.: Übersicht Genauigkeit - Abgebildet sind die besten Registrierungsergebnisse (links) der einzelnen Algorithmen auf den entsprechend reduzierten Szenarien. Die zugehörigen Auflösungen sind über den Registrierungsfehler abgetragen. Außerdem sind die dafür notwendigen Laufzeiten (rechts) einzusehen.

Die genauesten Registrierungsergebnisse wurden vom NDT erreicht. Für räumlich reduzierte Punktwolken waren bei genaueren Ergebnissen geringere Auflösungen notwendig. Nahezu identische Genauigkeiten erzielt der ICP auf meist geringeren Auflösungen, wodurch auch die Laufzeiten seiner besten Registrierungsergebnisse im Allgemeinen geringer ausfallen. Über die verschiedenen Szenarien hinweg erweisen sich die Registrierungsergebnisse sehr stabil; lediglich die Outdoor-Szenerie reißt aus.

PFH und SI registrieren mit einem nahezu identischen Fehlern auf den einzelnen Szenarien. Etwas genauere Ergebnisse erzielt dabei der PFH-Algorithmus bei ebenfalls nahezu identischen Auflösungen. Einen deutlichen Zeitvorteil können dabei jedoch die Spin-Images für sich verbuchen. Räumliches Sub-Sampling führt bei beiden Algorithmen zu einem genaueren Ergebnis.

Im Vergleich zu ICP und NDT erzielen PFH und SI sehr deutlich ungenauere Registrierungsergebnisse. Erwähnt sei dabei, dass eine besonders akkurate Registrierung nicht ihrer Konzeption entspricht.

Werden die Ergebnisse aller Algorithmen aus Kapitel 11 resümiert kann gesagt werden, dass für ein räumliches Sub-Sampling bei allen Verfahren und über sämtliche Auflösungen hinweg ein stabileres Registrierungsergebnis, als es für eine diskrete Reduktion der Fall ist, zu beobachten war. Auch scheint dies für die Stabilität der Parameter zu gelten. Für SI und PFH ging außerdem meist eine Reduktion der Laufzeit einher, für ICP und NDT ergab sich meist das Gegenteil. Eine alternative Datenstruktur zum hier verwendeten KD-Tree könnte dies jedoch ändern.

Basierend auf den in dieser Diplomarbeit durchgeführten Untersuchungen kann der Schluss gezogen werden, dass bei einer ausreichend genauen Sensorinformation der ICP, als das Beste der hier untersuchten Verfahren ist, um unterschiedliche Szenarien in praktikabler

---

Zeit sehr genau zu registrieren. Er erzielte die stabilsten Registrierungsergebnisse über die verschiedenen diskreten und räumlichen Auflösungen hinweg und erlaubt eine Registrierung der verschiedenen Szenarien mit einem gemeinsamen Parameterset.

Sind Sensorinformationen nur unzuverlässig oder sehr ungenau vorhanden sollte auf den SI-Algorithmus zurückgegriffen werden. Auch er registriert verschiedene Szenarien relativ zuverlässig und benötigt dafür deutlich weniger Zeit als der PFH, auch wenn er nicht dessen Genauigkeiten erreicht. Wird besonderen Wert auf die Genauigkeit und die Laufzeit gelegt, unabhängig von einer Stabilität über verschiedene Szenarien hinweg, empfiehlt sich eine Verwendung des NDT- bzw. bei nicht vorhandener initialer Ausrichtung der FPFH-Algorithmus.

In dieser Arbeit nicht behandelt aber sporadisch untersucht wurde das Verhalten des ICP und NDT bei verschiedenen initialen Ausrichtungen. Ergeben hat sich dabei, dass beide Algorithmen selbst bei einer initialen Ausrichtung von  $\Gamma_{\pm 5, \pm 5, \pm 3}^{\pm 20}$  noch immer Ergebnisse liefern, die als konvergiert betrachtet werden können. Eine Obergrenze soll hierdurch jedoch nicht definiert sein. Wird eine Untersuchung der initialen Ausrichtungen durchgeführt erscheint auch sinnvoll, eine iterative Anpassung der Parameter zu betrachten.

Für deskriptorbasierte Verfahren existieren bereits Ansätze, die eine Detektion sogenannter Keypoints (siehe Abschnitt 7.2.1), auf den zu registrierenden Punktwolken durchführen. Merkmale (z.b. PFH oder SI) werden anschließend nur für die Keypoints berechnet. Dies kann den Rechenaufwand reduzieren, wenn die Berechnung der Keypoints schneller erfolgt, als das Berechnen und Verwerfen wenig markanter Merkmale. Außerdem bietet es die Möglichkeit, aussagekräftige Bereiche einer Szene über eine Repräsentation zu identifizieren, die zur Korrespondenzfindung ungeeignet ist.

Weiterhin wäre aufschlussreich das Verhalten feinregistrierender Algorithmen (ICP und NDT) prozessiert auf den Ergebnissen grobregistrierender Algorithmen (PFH und SI) zu untersuchen. Besonders interessant könnte dabei die Betrachtung der Stabilität auf unterschiedlich diskret und räumlich reduzierten Szenarien ausfallen.



## 13. Zusammenfassung

INLEITEND wurde die Problematik der paarweisen Registrierung von hochaufgelösten Punktwolken mit Verfahren erläutert, die auf geometrischen Merkmalen basieren. Diese motivierte eine Analyse alternativer, punkt- und deskriptorbasierter Verfahren; Anforderungen an diese wurden in Abschnitt 1.2 definiert.

Kapitel 2 befasste sich mit den zum Verständnis notwendigen Grundlagen. Vorgestellt wurde dabei der zur Erzeugung der Punktwolken verwendete Laserscanner FARO® Focus<sup>3D</sup> sowie mögliche Sensoren zur Erhebung von Lage und Positionsinformationen. Anschließend wurden die Eigenschaften der untersuchten Punktwolken und ihre mögliche Repräsentationen im Arbeitsspeicher des Computers beschrieben.

Nachdem das in dieser Diplomarbeit betrachtete Registrierungsproblem in Abschnitt 3.1 definiert war, wurde eine Klassifikation verschiedener Algorithmen vorgestellt, die dieses Problem zu lösen versuchen (siehe Abschnitt 3.2).

Für die Evaluierung der Registrierungsergebnisse wurden im darauf folgenden Kapitel (4) Fehlermaße definiert, die eine gesonderte Betrachtung der Rotation und Translation erlauben.

Kapitel 5 beschrieb die implementierte Testumgebung in welcher die Analysen durchgeführt wurden. Außerdem wurden die einzelnen Testszenarien, eine Indoor- Hybrid- und Outdoor-Szene, vorgestellt.

Zur Reduktion der Datenmengen wurden in Kapitel 6 die Methoden des diskreten und räumlichen Sub-Sampling vorgestellt sowie deren Unterschiedlichkeit der Betrachtung von Entfernungsinformation herausgearbeitet.

Die anschließenden vier Kapitel präsentierten die zur Untersuchung herangezogenen Algorithmen. Dazu gehören die PFH (Kapitel 7), die SI (Kapitel 8), der ICP (Kapitel 9) und der NDT (Kapitel 10).

In Kapitel 11 wurde dann gezeigt, dass punktbasierte Verfahren in der Lage sind hochaufgelöste 3D-Punktwolken zu registrieren. Festgestellt wurde dabei, dass der Berechnungsaufwand der vorgestellten Verfahren jedoch eine Reduktion der Datenmenge erfordert. Außerdem führen nicht reduzierte Daten häufig zu schlechteren Ergebnissen, da offensichtlich zu viel Information zu Mehrdeutigkeit führt. Verfahren die keine Repräsentation der Punktwolken in Form von Merkmalen verwenden (ICP und NDT) sind bei geringen Auflösungen deutlich schneller und auch präziser. Speziell auf reduzierten Punktmengen ist das Berechnen von Merkmalen weitaus aufwendiger als eine direkter Punkt-zu-Punkt Vergleich über mehrere Iterationen hinweg.



## A. Transformationsbestimmung mittels Singulärwertzerlegung

$\mathbf{E}^s$  sei ein Registrierungsproblem wie in Abschnitt 3.1 gegeben. Dabei sind  $P$  und  $Q$  die zu registrierenden Punktwolken und  $C$  eine dafür bestimmte Korrespondenzmenge.  $C_P = \{c_{pi} | c_{pi} \in P \wedge c_{pi} \in C, 1 \leq i \leq |P|\}$  bezeichnet die Menge der Punkte aus  $P$  die zu Punkten aus  $Q$  korrespondieren. Analog ist  $C_Q = \{c_{qj} | c_{qj} \in Q \wedge c_{qj} \in C, 1 \leq j \leq |Q|\}$  definiert.

Die Translation  $T$  lässt sich mit Hilfe der Schwerpunkte  $S_{C_P}$  und  $S_{C_Q}$  der Korrespondenzmengen wie folgt berechnen:

$$(A.1) \quad T = S_{C_P} - S_{C_Q} \quad \text{mit} \quad S_{C_P} = \frac{1}{C_P} \sum_{i=1}^{C_P} c_{pi}, \quad S_{C_Q} = \frac{1}{C_Q} \sum_{j=1}^{C_Q} c_{qj}$$

Mittels Korrelationsmatrix und Singulärwertzerlegung [AHB87] lässt sich anschließend die Rotation bestimmen. Um eine korrekte Korrelationsmatrix  $H \in \mathbb{R}^{3 \times 3}$  zu erhalten, müssen die Korrespondenzmengen zentriert werden. Bewerkstelligt wird dies über die zuvor berechneten Schwerpunkte, sodass die Bildungsvorschrift von  $K$  gegeben ist durch:

$$(A.2) \quad H = \sum_{k=1}^{|C|} (c_{k_p} - S_{C_P})(c_{k_q} - S_{C_Q})^T$$

Diese lässt sich über die Singulärwertzerlegung als Produkt von Matrizen darstellen:

$$H = UDV^T, \quad U, D, V \in \mathbb{R}^{\times}$$

$U$  und  $V^T$  beinhalten dabei die durch die Abbildung  $H$  beschriebene Rotation,  $D$  die Verzerrung. Die gesuchte Rotation  $R$  ist dann gegeben durch [ELF97]:

$$(A.3) \quad R = VU^T$$

Aus einer Menge von Korrespondenzen wurde somit eine Transformation  $(R, T)$  berechnet, die alle korrespondierenden Punkte möglichst aufeinander abbildet. Dies bedeutet, dass die Summe der quadratischen Abstände zwischen korrespondierenden Punkten nach Anwendung der Transformation minimal ist.



## B. Normalen-Berechnung mittels Hauptkomponenten Analyse (PCA)

Die Normale eines sich auf einer Oberfläche befindenden Punktes zu bestimmen entspricht der Bestimmung der Tangentialebene an diesem Punkt [Ruso9]. Das hierzu verwendete Verfahren nennt sich Hauptkomponenten Analyse bzw. Principal Component Analysis (PCA).

Es sei  $P$  eine Punktwolke wie in Abschnitt 3.1 definiert und außerdem weiter ein Punkt  $p \in P$  und seine Kugelumgebung mit Radius  $r$ . Die Menge seiner Nachbarn ist dann gegeben durch  $K = \{k_i \in P \mid \|p - k_i\|_2 \leq r, 1 \leq i < |P|\}$ . Der zugehörige Schwerpunkt  $s$  der Menge  $K$  definiert als:

$$(B.1) \quad s = \frac{1}{|K|} \sum_{i=1}^{|K|} k_i$$

Mit Hilfe dessen lässt sich die folgende Kovarianzmatrix berechnen:

$$(B.2) \quad C = \frac{1}{|K|} \sum_{i=1}^{|K|} (k_i - s) \cdot (k_i - s)^T$$

Diese ist symmetrisch und positiv semidefinit. Dadurch besitzt sie reelle Eigenwerte  $\lambda_j \in \mathbb{R}$ . Die zugehörigen Eigenvektoren  $\vec{v}_j$  bilden ein Orthogonalsystem entsprechend den Hauptkomponenten von  $K$ . Der zum kleinsten Eigenwert  $0 \leq \lambda_0 \leq \lambda_1 \leq \lambda_2$  gehörende Eigenvektor  $\vec{v}_0$  entspricht dann einer Approximation der gesuchten Normale  $\pm \vec{n}_p = (n_x, n_y, n_z)$ .

Oft ist es notwendig die Normalen aller Punkte einer Punktwolke  $P$  einheitlich auszurichten (Bestimmen des Vorzeichens). Als dafür notwendigen Bezugspunkt wird der Standpunkt des Laserscanners  $v_P$ , oft auch als Viewpoint bezeichnet, herangezogen. Für einen Punkt  $p \in P$  muss das Vorzeichen so gewählt werden, dass gilt:

$$(B.3) \quad \vec{n}_p \cdot (v_P - p) > 0$$



## C. Initiale Ausrichtungen

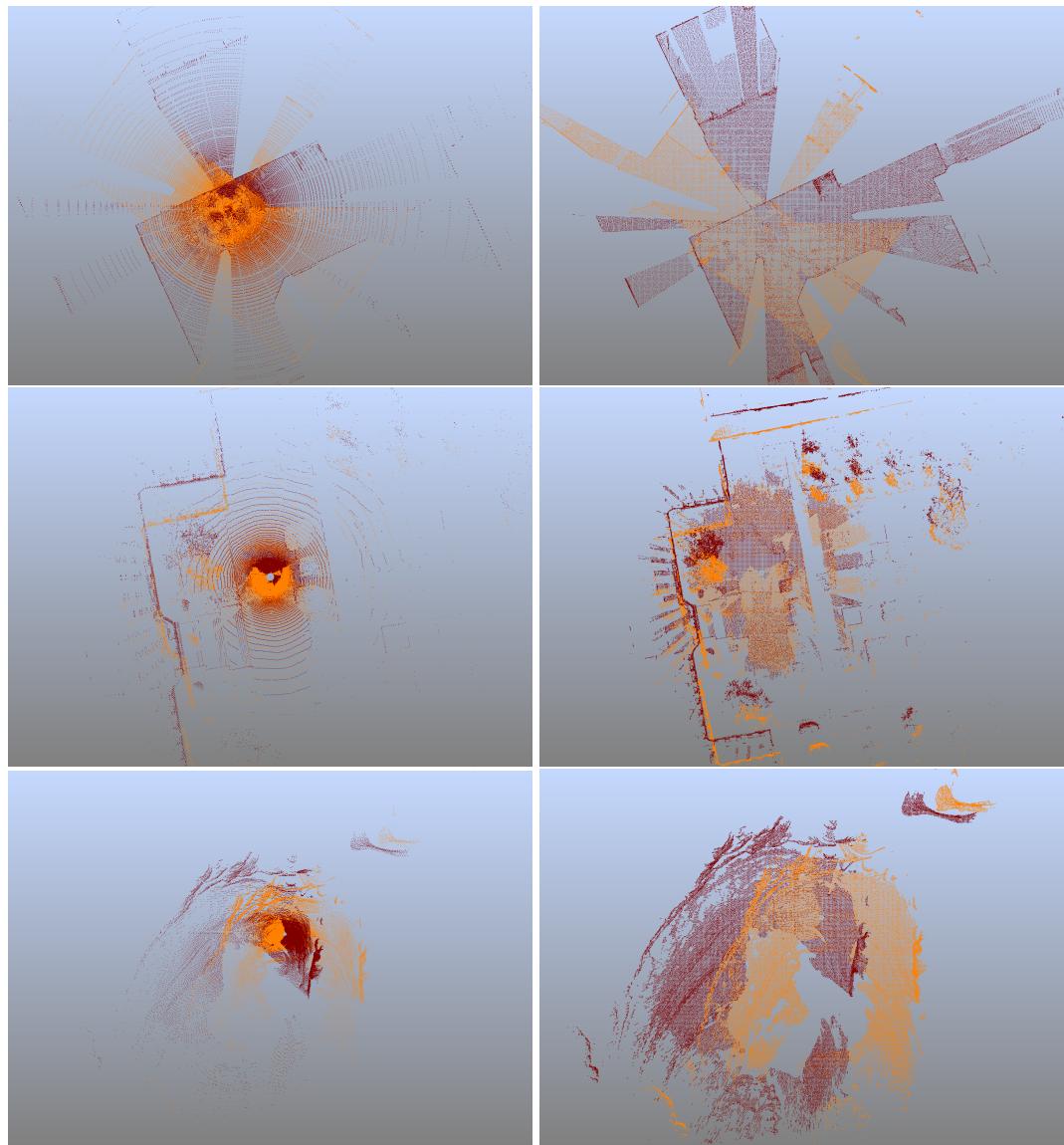
Wird die initiale Transformation der Datenpunktewolke als Registrierungsergebnis betrachtet, lassen sich für diese die in Kapitel 4 eingeführten Fehlermaße berechnen. Sind die Fehler nach einer Registrierung geringer als die initial berechneten, war die Registrierung erfolgreich.

Wird keine initiale Ausrichtung vorgenommen liegen die Punktwolken in einem gemeinsamen (globalen) Koordinatensystem, mit der Platzierung des Laserscanners im Ursprung vor. Die Rotation entspricht der Ausrichtung des Laserscanners bei der Aufnahme. Die Fehler der „globalen“ Ausrichtung sind in Tabelle C.1 einzusehen, die visuelle Darstellung davon in Abbildung C.2.

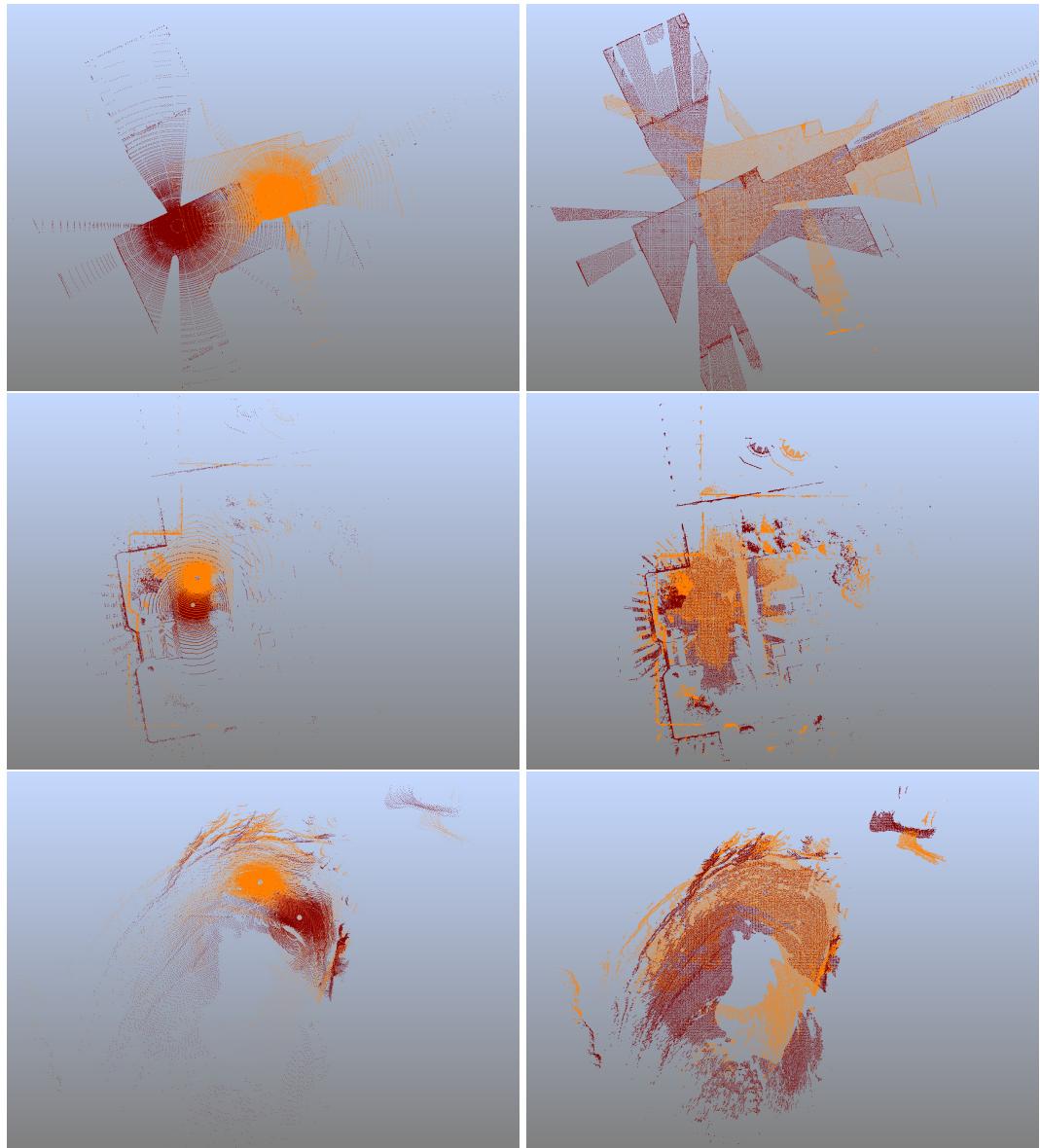
Ist die initiale Ausrichtung gemäß Gleichung 5.1 für alle Szenarien dieselbe, sind auch die resultierenden Fehler identisch. Diese sind in Tabelle C.1 für  $\Gamma_{(3.0,3.0,1.0)^T}^{-10}$  festgehalten und in Abbildung C.1 zu betrachten.

### C. Initiale Ausrichtungen

---



**Abbildung C.1.: Globale Ausrichtung** Zu sehen sind die verschiedenen Testszenarien in der „globalen“ Ausrichtung. Von oben nach unten: Indoor-Szene, Hybrid-Szene, Outdoor-Szene. Von links nach rechts: Diskretes Sub-Sampling, räumliches Sub-Sampling.



**Abbildung C.2.: Initiale Ausrichtung** - Zu sehen sind die verschiedenen Testszenarien nach der initialen Ausrichtung  $\Gamma_{(3,0,3,0,1,0)^T}^{-10}$ . Von oben nach unten: Indoor-Szene, Hybrid-Szene, Outdoor-Szene. Von links nach rechts: Diskretes Sub-Sampling, räumliches Sub-Sampling.

### C. Initiale Ausrichtungen

---

		Indoor-Szenario	Hybrid-Szenario	Outdoor-Szenario
<b>Globale Fehler</b>				
$e_{Gesamt}$		4.05	7.30	16.87
$e_{Translation}$	(m)	3.24	7.28	16.86
$e_{Rotation}$		1.70	0.04	0.23
$\vec{e}_{Translation}$	<b>X (m)</b>	2.96	-1.29	-14.50
	<b>Y (m)</b>	-1.33	7.18	7.33
	<b>Z (m)</b>	0.00	-0.14	4.51
$\vec{e}_{Rotation}$	<b>X (<math>\frac{1}{m}</math>)</b>	0.94	0.02	0.14
	<b>Y (<math>\frac{1}{m}</math>)</b>	0.76	0.01	0.04
	<b>Z (<math>\frac{1}{m}</math>)</b>	0.00	0.00	0.00
<b>Initiale Fehler <math>\Gamma_{(3.0,3.0,1.0)^T}^{-10}</math></b>				
$e_{Gesamt}$			4.37	
$e_{Translation}$	(m)		4.37	
$e_{Rotation}$			0.18	
$\vec{e}_{Translation}$	<b>X (m)</b>		-3.00	
	<b>Y (m)</b>		-3.00	
	<b>Z (m)</b>		-1.00	
$\vec{e}_{Rotation}$	<b>X (<math>\frac{1}{m}</math>)</b>		0.00	
	<b>Y (<math>\frac{1}{m}</math>)</b>		0.10	
	<b>Z (<math>\frac{1}{m}</math>)</b>		0.00	

**Tabelle C.1.: Initiale Fehler** - Aufgeführt sind die resultierenden Fehler der globalen Ausrichtungen sowie der durch  $\Gamma_{(3.0,3.0,1.0)^T}^{-10}$  gegebene initiale Ausrichtung der einzelnen Testszenarien. Die resultierenden Fehler für  $\Gamma_{(3.0,3.0,1.0)^T}^{-10}$  sind für alle Szenarien identisch.

# Literaturverzeichnis

- [AHB87] K. S. Arun, T. S. Huang, S. D. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5):698–700, 1987. (Zitiert auf Seite 119)
- [Ameo8] D. of Defense United States of America. Global Positioning System Standard Positioning Service Performance Standard, 2008. (Zitiert auf Seite 18)
- [Bero9] S. Bertele. Extraktion charakteristischer Positionen und Objekte in hochaufgelösten 3D-Scandaten, 2009. (Zitiert auf Seite 31)
- [BL97] J. S. Beis, D. G. Lowe. Shape Indexing Using Approximate Nearest-Neighbour Search in High-Dimensional Spaces. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, S. 1000–. IEEE Computer Society, Washington, DC, USA, 1997. (Zitiert auf den Seiten 25 und 26)
- [BM92] P. J. Besl, N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992. (Zitiert auf den Seiten 34 und 65)
- [BOo08] C. Basdogan, A. C. Oztireli. A new feature-based method for robust and efficient rigid-body registration of overlapping point clouds. *The Visual Computer*, 24(7–9):679–688, 2008. (Zitiert auf Seite 32)
- [Bogo08] I. Bogicevic. Korrespondenzfindung zwischen Targetmengen im 3D-Raum, 2008. (Zitiert auf Seite 31)
- [BS96] K. Brunnstrom, A. J. Stoddart. Genetic algorithms for free-form surface matching. In *Proc. 13th Int Pattern Recognition Conf*, Band 4, S. 689–693. 1996. (Zitiert auf Seite 34)
- [BS03] P. Biber, W. Strasser. The normal distributions transform: a new approach to laser scan matching. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2003)*, Band 3, S. 2743–2748. 2003. (Zitiert auf den Seiten 35, 73, 74 und 75)
- [CF01] R. J. Campbell, P. J. Flynn. A survey of free-form object representation and recognition techniques. *Comput. Vis. Image Underst.*, 81(2):166–210, 2001. (Zitiert auf Seite 30)
- [CHC] C. song Chen, Y.-P. Hung, J. bo Cheng. RANSAC-based DARCES: A New Approach for Fast Automatic Registration of Partially Overlapping Range Images. (Zitiert auf Seite 34)

- [CHH99] O. Carmichael, D. Huber, M. Hebert. Large data sets and confusing scenes in 3-D surface matching and recognition. In *Proceedings of the 2nd international conference on 3-D digital imaging and modeling, 3DIM'99*, S. 358–367. IEEE Computer Society, Washington, DC, USA, 1999. (Zitiert auf Seite 33)
- [CJ97] C. S. Chua, R. Jarvis. Point Signatures: A New Representation for 3D Object Recognition. *Int. J. Comput. Vision*, 25(1):63–85, 1997. (Zitiert auf Seite 32)
- [CSK05] D. Chetverikov, D. Stepanov, P. Krsek. Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*, 23(3):299 – 309, 2005. (Zitiert auf Seite 34)
- [CTL04] C. K. Chow, H. tat Tsui, T. Lee. Surface registration using a dynamic genetic algorithm. *Pattern Recognition*, 37:105–117, 2004. (Zitiert auf Seite 35)
- [CYL98] D. H. Chung, I. D. Yun, S. U. Lee. Registration of multiple-range views using the reverse-calibration technique. *Pattern Recognition*, 31(4):457 – 464, 1998. (Zitiert auf Seite 33)
- [DB06] C. Dold, C. Brenner. Registration of terrestrial laser scanning data using planar patches and image data. Institute for Cartography and Geoinformatics Leibniz University of Hannover, 2006. (Zitiert auf Seite 31)
- [DH01] M. H. Daniel Huber. Fully Automatic Registration Of Multiple 3D Data Sets, 2001. (Zitiert auf Seite 28)
- [Dolo05] C. Dold. Extended Gaussian Images for the registration of terrestrial scan data. ISPRS WG III/3, III/4, V/3 Workshop Laser scanning 2005, Enschede, the Netherlands, 2005. (Zitiert auf Seite 33)
- [ELF97] D. W. Eggert, A. Lorusso, R. B. Fisher. Estimating 3D rigid body transformations: a comparison of four major algorithms. *MAchine Vision and Applications*, 9:272–290, 1997. (Zitiert auf den Seiten 69 und 119)
- [EMSS96] M. Erdmann, E. G. Mit, D. A. Simon, D. A. Simon. Fast and Accurate Shape-Based Registration. Technischer Bericht, Carnegie Mellon University Pittsburgh, Pennsylvania 15213, 1996. (Zitiert auf Seite 67)
- [FAR12] FARO. FARO Europe GmbH & Co. KG, 2012. URL <http://www.faroeurope.com>. (Zitiert auf den Seiten 16, 17, 18, 19, 20, 21, 31 und 48)
- [FBF77] J. H. Friedman, J. L. Bentley, R. A. Finkel. An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Trans. Math. Softw.*, 3(3):209–226, 1977. (Zitiert auf Seite 67)
- [FSMo08] M. S. Foster, J. R. Schott, D. W. Messinger. Spin-image target detection algorithm applied to low density 3D point clouds, 2008. (Zitiert auf den Seiten 33 und 63)
- [HH99] D. F. Huber, M. Hebert. A new approach to 3-D terrain mapping. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS '99*, Band 2, S. 1121–1127. 1999. (Zitiert auf Seite 33)

- [Hoc05] N. Hochgeschwender. Untersuchung von statistischen Selbstlokalisierungsmethoden in der mobilen Robotik. Projektarbeit am Institut für Informatik Fachbereich Elektrotechnik und Informatik Hochschule Ravensburg-Weingarten University of Applied Sciences, 2005. (Zitiert auf den Seiten 19 und 27)
- [JA09] B. JONES, M. AOUN. Learning 3D Point Cloud Histograms. CS229 Machine Learning Project, 2009. (Zitiert auf Seite 32)
- [JH99] A. E. Johnson, M. Hebert. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):433–449, 1999. (Zitiert auf den Seiten 33 und 62)
- [JK97] A. Johnson, S. B. Kang. Registration and Integration of Textured 3-D Data. In *International Conference on Recent Advances in 3-D Digital Imaging and Modeling (3DIM '97)*, S. 234 – 241. 1997. (Zitiert auf Seite 67)
- [Joh97] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. Dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1997. (Zitiert auf den Seiten 33, 61, 62 und 63)
- [JRE<sup>+</sup>04] J. Talaya, R. Alamus, E. Bosch, A. Serra, W. Kornu, A. Baron. Integration of a terrestrial laser scanner with GPS/IMU orientation sensors. In XX Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS), 2004. (Zitiert auf den Seiten 18 und 27)
- [KTMAo8] T. Kaminade, T. Takubo, Y. Mae, T. Arai. The generation of environmental map based on a NDT grid mapping -Proposal of convergence calculation corresponding to high resolution grid-. In *Proc. IEEE Int. Conf. Robotics and Automation ICRA 2008*, S. 1874–1879. 2008. (Zitiert auf Seite 75)
- [LCE06] E. Lomonosov, D. Chetverikov, A. Ekárt. Pre-registration of arbitrarily oriented 3D surfaces using a genetic algorithm. *Pattern Recogn. Lett.*, 27(11):1201–1208, 2006. (Zitiert auf Seite 34)
- [Lib12a] P. C. Library. Point Cloud Library, 2012. URL <http://docs.pointclouds.org>. (Zitiert auf den Seiten 43 und 64)
- [Lib12b] P. C. Library. Point Cloud Library Class - pcl :: ndt :: NormalDistributionsTransform. Point Cloud Library API Trunk, 2012. URL [http://docs.pointclouds.org/trunk/classpcl\\_1\\_1\\_normal\\_distributions\\_transform.html](http://docs.pointclouds.org/trunk/classpcl_1_1_normal_distributions_transform.html). (Zitiert auf den Seiten 75 und 107)
- [Lib12c] P. C. Library. Point Cloud Library Class - pcl :: registration :: CorrespondenceEstimation. Point Cloud Library API v1.6.0 (released on 2012.07.17), 2012. URL [http://docs.pointclouds.org/1.6.0/classpcl\\_1\\_1registration\\_1\\_1\\_correspondence\\_estimation.html](http://docs.pointclouds.org/1.6.0/classpcl_1_1registration_1_1_correspondence_estimation.html). (Zitiert auf Seite 59)

- [Lib12d] P. C. Library. Point Cloud Library Class - pcl::registration::TransformationEstimationSVD. Point Cloud Library API v1.6.0 (released on 2012.07.17), 2012. URL [http://docs.pointclouds.org/1.6.0/classpcl\\_1\\_1registration\\_1\\_1\\_transformation\\_estimation\\_s\\_v\\_d.html](http://docs.pointclouds.org/1.6.0/classpcl_1_1registration_1_1_transformation_estimation_s_v_d.html). (Zitiert auf den Seiten 59 und 64)
- [Lib12e] P. C. Library. Point Cloud Library Class - pcl::FPFHEstimation. Point Cloud Library API v1.6.0 (released on 2012.07.17), 2012. URL [http://docs.pointclouds.org/trunk/classpcl\\_1\\_1\\_f\\_p\\_f\\_h\\_estimation.html](http://docs.pointclouds.org/trunk/classpcl_1_1_f_p_f_h_estimation.html). (Zitiert auf den Seiten 58 und 59)
- [Lib12f] P. C. Library. Point Cloud Library Class - pcl::IterativeClosestPoint. Point Cloud Library API v1.6.0 (released on 2012.07.17), 2012. URL [http://docs.pointclouds.org/1.6.0/classpcl\\_1\\_1\\_iterative\\_closest\\_point.html](http://docs.pointclouds.org/1.6.0/classpcl_1_1_iterative_closest_point.html). (Zitiert auf Seite 70)
- [Lib12g] P. C. Library. Point Cloud Library Class - pcl::MultiscaleFeaturePersistence. Point Cloud Library API v1.6.0 (released on 2012.07.17), 2012. URL [http://docs.pointclouds.org/1.6.0/classpcl\\_1\\_1\\_multiscale\\_feature\\_persistence.html](http://docs.pointclouds.org/1.6.0/classpcl_1_1_multiscale_feature_persistence.html). (Zitiert auf Seite 59)
- [Lib12h] P. C. Library. Point Cloud Library Class - pcl::NormalEstimation. Point Cloud Library API v1.6.0 (released on 2012.07.17), 2012. URL [http://docs.pointclouds.org/1.6.0/classpcl\\_1\\_1\\_normal\\_estimation.html](http://docs.pointclouds.org/1.6.0/classpcl_1_1_normal_estimation.html). (Zitiert auf den Seiten 58 und 64)
- [Lib12i] P. C. Library. Point Cloud Library Class - pcl::search::KdTree. Point Cloud Library API v1.6.0 (released on 2012.07.17), 2012. URL [http://docs.pointclouds.org/1.6.0/classpcl\\_1\\_1search\\_1\\_1\\_kd\\_tree.html](http://docs.pointclouds.org/1.6.0/classpcl_1_1search_1_1_kd_tree.html). (Zitiert auf den Seiten 26, 64 und 111)
- [Lib12j] P. C. Library. Point Cloud Library Class - pcl::UniformSampling. Point Cloud Library API v1.6.0 (released on 2012.07.17), 2012. URL [http://docs.pointclouds.org/1.6.0/classpcl\\_1\\_1\\_uniform\\_sampling.html](http://docs.pointclouds.org/1.6.0/classpcl_1_1_uniform_sampling.html). (Zitiert auf Seite 51)
- [Lib12k] P. C. Library. Point Cloud Library Class - pcl::VoxelGrid. Point Cloud Library API v1.6.0 (released on 2012.07.17), 2012. URL [http://docs.pointclouds.org/1.6.0/classpcl\\_1\\_1 voxel\\_grid.html](http://docs.pointclouds.org/1.6.0/classpcl_1_1 voxel_grid.html). (Zitiert auf Seite 111)
- [Mag09] M. Magnusson. *The Three-Dimensional Normal-Distributions Transform, an Efficient Representation for Registration, Surface Analysis, and Loop Detection*. Dissertation, Oerebro University, 2009. Oerebro Studies in Technology 36. (Zitiert auf den Seiten 35, 73, 75 und 76)
- [MLD07] M. Magnusson, A. Lilienthal, T. Duckett. Scan registration for autonomous mining vehicles using 3D-NDT: Research Articles. *J. Field Robot.*, 24(10):803–827, 2007. (Zitiert auf den Seiten 35, 73 und 74)

- [MMRM10] A. Mandow, J. L. Martenez, A. J. Reina, J. Morales. Fast range-independent spherical subsampling of 3D laser scanner points and data reduction performance evaluation for scene registration. *Pattern Recognition Letters*, 31(11):1239 – 1250, 2010. (Zitiert auf Seite 47)
- [MPIDo6] A. Makadia, A. Patterson, IV, K. Daniilidis. Fully automatic registration of 3d point clouds. In *IN CVPR '06: PROCEEDINGS OF THE 2006 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION*, S. 1297–1304. 2006. (Zitiert auf Seite 33)
- [MTMp92] J. J. More, D. J. Thuente, P. Mcs-p. Line Search Algorithms With Guaranteed Sufficient Decrease. *ACM Trans. Math. Software*, 20:286–307, 1992. (Zitiert auf den Seiten 76 und 107)
- [NÜo2] A. Nüchter. Autonome Exploration und Modellierung von 3D-Umgebungen. RHEINISCHE FRIEDRICH-WILHELMUS-UNIVERSITÄT BONN, INSTITUT FÜR INFORMATIK III, Institut für Autonome Intelligente Systeme, 2002. (Zitiert auf den Seiten 19 und 27)
- [NLHo7] A. Nuechter, K. Lingemann, J. Hertzberg. Cached k-d tree search for ICP algorithms. In *International Conference on 3-D Imaging and Modeling*, S. 419–426. 2007. (Zitiert auf den Seiten 26 und 67)
- [PS03] S.-Y. Park, M. Subbarao. An accurate and fast point-to-plane registration technique. *Pattern Recogn. Lett.*, 24(16):2967–2976, 2003. (Zitiert auf Seite 68)
- [RBB09] R. B. Rusu, N. Blodow, M. Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. Kobe, Japan, 2009. (Zitiert auf den Seiten 32, 53 und 56)
- [RBMB08] R. B. Rusu, N. Blodow, Z. C. Marton, M. Beetz. Aligning Point Cloud Views using Persistent Feature Histograms. In *Proceedings of the 21st IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nice, France, 2008. (Zitiert auf den Seiten 32, 53 und 56)
- [REF<sup>+</sup>94] P. Robotique, P. Epidaure, J. Feldmar, J. Feldmar, J. Feldmar, N. Ayache, N. Ayache, N. Ayache. Rigid, Affine and Locally Affine Registration of Free-Form Surfaces. *IJCV*, 18:99–119, 1994. (Zitiert auf Seite 33)
- [Ripo5] B. Ripperda, N. Marker-Free Registration of Terrestrial Laser Scans Using the Normal Distribution Transform. Proceedings of the ISPRS Working Group V/4 Workshop 3D-ARCH 2005, 2005. (Zitiert auf den Seiten 73 und 74)
- [RL01] S. Rusinkiewicz, M. Levoy. Efficient Variants of the ICP Algorithm. In *INTERNATIONAL CONFERENCE ON 3-D DIGITAL IMAGING AND MODELING*. 2001. (Zitiert auf den Seiten 65, 66 und 70)

- [RMBBo8] R. B. Rusu, Z. C. Marton, N. Blodow, M. Beetz. Learning Informative Point Classes for the Acquisition of Object Model Maps. In *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. Hanoi, Vietnam, 2008. (Zitiert auf den Seiten 32, 53 und 54)
- [Rus09] R. B. Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. Dissertation, Computer Science department, Technische Universitaet Muenchen, Germany, 2009. Advisor: Univ.-Prof. Michael Beetz (TUM) Ph.D.; Committee: Univ.-Prof. Dr. Nassir Navab (TUM), Univ.-Prof. Michael Beetz (TUM) Ph.D., Prof. Kurt Konolige (Stanford) Ph.D., Prof. Gary Bradski (Stanford) Ph.D.; summa cum laude. (Zitiert auf den Seiten 32, 53, 54, 55, 56, 57, 59 und 121)
- [SB05] S. Schuhmacher, J. Boehm. Georeferencing of terrestrial laserscanner data for applications in architectural modeling, 2005. (Zitiert auf Seite 27)
- [SCM10] A. Sehgal, D. Cernea, M. Makaveeva. Real-Time scale invariant 3d range point cloud registration. In *Proceedings of the 7th international conference on Image Analysis and Recognition - Volume Part I, ICIAR'10*, S. 220–229. Springer-Verlag, Berlin, Heidelberg, 2010. (Zitiert auf Seite 55)
- [SMFFo05] J. Salvi, C. Matabosch, D. Fofi, J. Forest. A review of recent range image registration methods with accuracy evaluation. Institute of Informatics and Applications, University of Girona, Av.Lluis Santalo, Girona, Spain Laboratoire Electronique, Informatique et Image, University of Burgundy, rue de la fonderie, Le Creusot, France, 2005. (Zitiert auf den Seiten 30, 32, 33 und 34)
- [TCC98] J.-P. Tarel, H. Civi, D. B. Cooper. Pose Estimation of Free-Form 3D Objects without Point Matching using algebraic surface Models. In *Proceedings of IEEE Worshop Model Based 3D Image Analysis*, S. 13–21. Mumbai, India, 1998. <Http://perso.lcpc.fr/tarel.jean-philippe/publis/mb3ia.html>. (Zitiert auf Seite 33)
- [Tre12] Treemetrics. TreeMetrics Ltd. National Software Centre, 2012. URL <Http://www.treemetrics.com>. (Zitiert auf den Seiten 19 und 20)
- [Tro08] M. Troester. Kantenorientierte Suche nach bildbestimmenden Geraden in 3D-Laser-Scan Daten, 2008. (Zitiert auf Seite 31)
- [TTo06] E. Takeuchi, T. Tsubouchi. A 3-D Scan Matching using Improved 3-D Normal Distributions Transform for Mobile Robotic Mapping. In *IROS*, S. 3068–3073. IEEE, 2006. (Zitiert auf Seite 75)
- [WHHo3] E. Wahl, U. Hillenbrand, G. Hirzinger. Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification. In *Proc. Fourth Int. Conf. 3-D Digital Imaging and Modeling 3DIM 2003*, S. 474–481. 2003. (Zitiert auf den Seiten 32, 53, 54 und 57)

Alle URLs wurden zuletzt am 31.12.2012 geprüft.

# Danksagung

**A**n dieser Stelle möchte ich mich bei allen Personen bedanken, die mich bei der Erstellung dieser Arbeit unterstützt und mich auf meinem Weg durchs Studium begleitet haben.

Ein besonderer Dank gilt Reinhard Becker, Martin Ossig und Michael Schanz. Sie haben mir diese Diplomarbeit in Kooperation mit der Firma FARO ermöglicht.

Weiteren Dank möchte ich meinem Betreuer Stefan Bertele aussprechen, der stets für mich ansprechbar war und mir viel Freiheit in der Gestaltung dieser Arbeit gelassen hat. In diesem Zuge bedanke ich mich auch bei Christian Häckh, Sebastian Bartmann und Timo Dritschler, die mich in Diskussionen gefordert und mit guten Ideen inspiriert haben.

Ebenfalls bedanken möchte ich mich bei Helmut Kramer und allen Mitarbeitern der Abteilung für Software der Firma FARO, die mir bei Problemen mit Rat und Tat zur Seite standen.

Zu guter Letzt möchte ich diese Arbeit meiner Familie sowie meiner Freundin Lina Abou-Warda widmen, die mich zu jedem Zeitpunkt unterstützt haben und auf sehr viel gemeinsame Zeit verzichten mussten.



## **Erklärung**

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

---

Ort, Datum , Simon Kontschak