



UNIVERSITÄT ZU LÜBECK  
INSTITUT FÜR ROBOTIK  
UND KOGNITIVE SYSTEME

# Physikalisch motivierte Punktwolkenregistrierung

Entwurf und Evaluierung eines neuen Konzepts zur effizienten  
Registrierung von Punktwolken

Philipp Jauer

Dissertation

Universität zu Lübeck  
Institut für Robotik und Kognitive Systeme





UNIVERSITÄT ZU LÜBECK

**Aus dem Institut für Robotik und Kognitive Systeme  
der Universität zu Lübeck  
Direktor: Prof. Dr.-Ing. Achim Schweikard**

**Physikalisch motivierte Punktwolkenregistrierung  
Entwurf und Evaluierung eines neuen Konzepts zur effizienten  
Registrierung von Punktwolken**

Inauguraldissertation  
zur  
Erlangung der Doktorwürde  
der Universität zu Lübeck

Aus der Sektion Informatik/Technik

vorgelegt von  
Philipp Jauer  
aus Wilhelm-Pieck-Stadt Guben

Lübeck, 2018



1. Berichterstatter: Prof. Dr.-Ing. Achim Schweikard

2. Berichterstatter: Prof. Dr. rer. nat. habil. Heinz Handels

Tag der mündlichen Prüfung: 26. Oktober 2018

Zum Druck genehmigt. Lübeck, den 01. November 2018



The fact that we live at the bottom of a deep gravity well, on the surface of a gas covered planet going around a nuclear fireball 90 million miles away and think this to be normal is obviously some indication of how skewed our perspective tends to be.

---

Douglas Adams



## Kurzfassung

Diese Arbeit stellt ein neues Konzept zur rigiden Punktwolkenregistrierung vor. Im Gegensatz zu bereits existierenden Ansätzen basiert dieses Verfahren auf den Gesetzen der klassischen Mechanik, Elektrostatik und Thermodynamik. Das Registrierungsproblem wird durch die Modellierung der Punktwolken als Mehrteilchensysteme gelöst. Kräfte, die auf sämtliche Einzelteilchen einwirken, konzentrieren sich auf den jeweiligen Massenschwerpunkt der Wolke. Durch die Kraftereinwirkung entsteht eine sukzessive Bewegung der zu registrierenden Punktwolke in Richtung einer statischen Referenzwolke, bis sich beide möglichst optimal überlagern. Die Punktwolkenbewegung leitet sich aus dem physikalischen Modell der Starrkörpertransformation ab. Kraftfelder bilden die Metrik der Registrierung und sind frei gestaltbar. Die in dieser Arbeit vorgestellten Beispielmetriken sind sowohl vom Newtonschen Gravitationsgesetz als auch vom Coulombschen Gesetz der Elektrostatik inspiriert. Dementsprechend lässt sich die Lage der Punktwolken zueinander nicht nur durch ihre räumliche Punktverteilung beschreiben, sondern auch durch zusätzliche Merkmale wie Farb- oder Helligkeitswerte.

Darüber hinaus wird ein effizienter Algorithmus vorgestellt, der das neue Registrierungsverfahren implementiert. Die Regularisierung von Punktwolkenbewegungen basiert auf der Methode des Simulated Annealing. Um die Laufzeitkomplexität bei sehr hochauflösenden Punktwolken zu verringern, wird das Monte-Carlo-Verfahren eingesetzt. Zudem ist der Algorithmus parallelisierbar und kann entsprechend performant auf Mehrkernrechnerarchitekturen wie zum Beispiel Grafikkartenprozessoren ausgeführt werden. Schließlich werden die Punktwolkenbewegungen durch Duale Quaternionen repräsentiert, die eine zusätzliche Glättung der Bewegung ermöglichen.

Eine umfangreiche Evaluierung des Algorithmus zeigt, dass das neue Konzept eine konkurrenzfähige Alternative zum bisher aktuellen Stand der Technik darstellt. Im Vergleich zu diversen Varianten und Implementierungen des Iterative-Closest-Point (ICP)-Algorithmus können Punktwolken mit einer um mehr als 50 % höheren Genauigkeit registriert werden. Ferner erweist sich der Algorithmus als ebenso robust wie der ICP, wenn Punktwolkendaten fehlerhaft oder verrauscht sind. Die Laufzeitkomplexität ist dabei stets konstant niedrig und unabhängig von der Punktwolkenauflösung.



## Abstract

This thesis presents a new concept for rigid point cloud registration. In contrast to existing approaches, it is based on the physics laws of classical mechanics, electrostatics and thermodynamics. The registration problem is solved by modelling point clouds as many-particle systems. Forces acting on the individual particles concentrate on the respective centre of mass of the cloud. Due to the application of force, a successive movement of the point cloud to be registered occurs in the direction of the static reference point cloud until both are aligned. The point cloud motion is derived from the physical model of rigid-body transformation. Force fields represent the metric of the registration and can be freely modeled. The sample metrics presented in this thesis are inspired by both, Newton's law of gravitation and Coulomb's law of electrostatics. Accordingly, the position of the point clouds can be described not only by their spatial point distribution, but also by additional characteristics such as color or intensity values.

Furthermore, an efficient algorithm is presented which implements the new registration approach. The regularization of point cloud movements is based on the method of simulated annealing. In order to keep the runtime complexity as low as possible even with very high-resolution point clouds, the Monte-Carlo-method is used. Furthermore, the algorithm can be parallelized and can be executed on multi-core processing architectures such as graphics processor units. Finally, the point cloud movements are represented by dual quaternions, which allow an additional smoothing of the movement.

An extensive evaluation of the algorithm shows that the new concept represents a competitive alternative to today's state of the art. Compared to various variants and implementations of the Iterative Closest Point (ICP) algorithm, point clouds can be registered with more than 50 % higher accuracy. Furthermore, the algorithm proves to be just as robust as the gold standard if point cloud data is afflicted by noise. The runtime complexity is always constantly low and independent of the point cloud resolution.



# Inhaltsverzeichnis

<b>Kurzfassung</b>	<b>ix</b>
<b>Abstract</b>	<b>xi</b>
<b>Mathematische Notationen</b>	<b>xv</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Grundlagen</b>	<b>7</b>
2.1 Punktwolken . . . . .	10
2.2 Punktwolkenregistrierung . . . . .	11
2.2.1 Stand der Technik . . . . .	12
2.2.2 Iterative-Closest-Point-Algorithmus . . . . .	15
<b>3 Algorithmenentwurf</b>	<b>27</b>
3.1 Punktwolken analog zur Physik . . . . .	29
3.1.1 Mehrteilchensysteme . . . . .	30
3.1.2 Starrkörpertransformation . . . . .	32
3.2 Metriken analog zu Kraftfeldern . . . . .	37
3.2.1 Newtons Gravitationsgesetz . . . . .	38
3.2.2 Coulombs Gesetz . . . . .	39
3.3 Implementierung . . . . .	44
3.3.1 Regularisierung . . . . .	46
3.3.2 Reduktion der Komplexität . . . . .	52
3.3.3 Transformationsbeschreibung . . . . .	61
<b>4 Evaluierung</b>	<b>71</b>
4.1 Machbarkeitsstudie . . . . .	74
4.1.1 A: Genauigkeit und Reproduzierbarkeit . . . . .	75
4.1.2 B: Effizienz . . . . .	79
4.1.3 C: Robustheit . . . . .	83

4.1.4	D: Merkmalerweiterung . . . . .	87
4.2	Parameteranalyse . . . . .	91
4.2.1	Transformationsdarstellungen . . . . .	92
4.2.2	Komplexität . . . . .	96
4.2.3	Regularisierung . . . . .	100
<b>5</b>	<b>Fazit</b>	<b>107</b>
	<b>Literaturverzeichnis</b>	<b>113</b>
	<b>Abbildungsverzeichnis</b>	<b>119</b>
	<b>Tabellenverzeichnis</b>	<b>121</b>
	<b>Abkürzungsverzeichnis</b>	<b>123</b>
	<b>Danksagung</b>	<b>125</b>
	<b>Lebenslauf</b>	<b>127</b>

## Mathematische Notationen

Symbol	Beschreibung
$\mathcal{X}$	Modell: statische Referenzpunktwolke in der Registrierung
$\mathbf{x}_j \in \mathbb{R}^3$	$j$ -te Punkt des Modelles
$\mathcal{F}_{\mathcal{X}}$	$D$ -Dimensionaler Merkmalsraum des Modelles
$\mathbf{f}_j^{\mathcal{X}} \in \mathbb{R}^D$	Merkmal des $j$ -te Punktes im Modell
$\mathcal{Y}$	Template: zu registrierende Punktwolke
$\mathbf{y}_i \in \mathbb{R}^3$	$i$ -te Punkt des Templates
$\mathcal{F}_{\mathcal{Y}}$	$D$ -Dimensionaler Merkmalsraum des Templates
$\mathbf{f}_i^{\mathcal{Y}} \in \mathbb{R}^D$	Merkmal des $i$ -te Punktes im Template
${}^{k-1}\mathcal{T}_k$	Rigide Transformation zwischen Iteration $k - 1$ und $k$
$\mathbf{x}_k$	Linearbewegung der $k$ -ten Iteration
$\theta_k$	Rotationswinkel der $k$ -ten Iteration
$\hat{\mathbf{L}}$	Rotationsachse der $k$ -ten Iteration
$E_k$	Kinetische Energie der $k$ -ten Iteration
$T_k$	Temperatur der $k$ -ten Iteration
$c \in ]0, 1[$	Abkühlungskonstante
$\varepsilon > 0$	Temperaturschwellwert (Abbruchkriterium)
$\mathcal{M}$	Stichprobengröße aus dem Modell
$\mathcal{N}$	Stichprobengröße aus dem Template



# 1 Einleitung

Das visuelle System ermöglicht den meisten Menschen die Bewältigung zahlreicher Probleme des täglichen Lebens. Die Verarbeitung visueller Reize erfolgt in Bruchteilen von Sekunden und das meist unbewusst. So ist der Mensch zum Beispiel in der Lage Gegenstände von verschiedenen Seiten zu betrachten und sich einen dreidimensionalen (3D) Eindruck vom Gesamtobjekt zu verschaffen. Das Gesehene wird im Gedächtnis konserviert, kann stets abgerufen und so auch anderen Personen beschrieben werden. Später kann eine Person dieses Objekt unmittelbar wiedererkennen, ganz gleich von welcher Seite er es betrachtet oder welchen Ausschnitt er davon sieht. Ähnliches gilt für die Orientierung sowohl in bekannten als auch unbekanntem Umgebungen. Werden neue Orte besucht, wird aus dem Gesehenen im Geiste eine Karte konzipiert. Zudem können bereits vorhandene "Karten" um neue Eindrücke erweitert werden. Verändert sich die Umgebung, werden diese Lagepläne instinktiv aktualisiert. Mit Hilfe dieser gedanklichen Karten ist der Mensch in der Lage, sich räumlich zu orientieren.

Bereits seit mehreren Jahrzehnten ist es das Ziel der Computer Vision, auch Maschinen das Sehen und Verstehen beizubringen. Mit Hilfe von Kameras, Sensoren und komplexen Algorithmen sollen Computer ähnliche visuelle Fähigkeiten erlangen wie der Mensch. Der anfänglich rein akademische Nutzen ist in der heutigen zunehmend digitalisierten Welt unentbehrlich. Heimcomputer, Smartphones, Spielekonsolen, Entertainmentssysteme oder auch Homeautomationssysteme bauen auf diesen visuellen Fähigkeiten auf. Das vereinfacht die Bedienung und bietet neue Konzepte der Unterhaltung.

Grundvoraussetzung vieler solcher Anwendungen ist die Teildisziplin der Computer Vision, die sich mit dem 3D-Sehen beschäftigt. Beispielsweise setzt der wachsende Trend in der autonomen oder teil-autonomen Mobilität viele dieser Techniken voraus [37, 87]. Hier sind es Fahrzeuge und Drohnen, die mit Sensoren ausgestattet sind, welche Entfernungen messen und somit ein 3D-Bild von der Umgebung erzeugen. Ähnlich dem visuellen System des Menschen erstellen Algorithmen Karten, mit deren Hilfe es dem Fahrzeug oder Fluggerät gelingt, sich sicher in der Umgebung zu orientieren und zu manövrieren [97]. Aber auch die Assistenzsysteme heutiger Automobile verwenden zum Teil die Fähigkeit des 3D-Sehens. Sei es bei der Einparkhilfe oder bei Sicherheitssystemen,

die dafür sorgen, dass das Fahrzeug bei Gefahren bremst bzw. im Stau eigenständig dem voraus fahrenden Verkehr folgt. Hinzu kommt die berührungslosen Bedienung von Steuerelementen im Fahrzeuginnenraum, die ebenfalls das 3D-Sehen voraussetzt.

Darüber hinaus kommen in der Medizin die Techniken der 3D Computer Vision vermehrt zum Einsatz. Das betrifft u.a. Assistenzsysteme, die es Ärzten<sup>1</sup> ermöglichen lediglich mit Hilfe von Gesten Operationslampen oder Monitore zu bewegen [33]. Außerdem werden die Technologien dazu genutzt, um Bewegungen von Patienten zu messen und diese zum Beispiel in der robotergestützten Strahlentherapie auszugleichen [102]. Aber auch roboterisierte Operationswerkzeuge können mittels 3D-Sehens die Bewegungen des Patienten erkennen und entsprechend reagieren [93]. Das unterstützt die Ärzte bei der Operation und hält medizinische Eingriffe minimalinvasiv.

Neben den Kameras und Sensoren sind es die Algorithmen der Computer Vision, die alle diese Aufgaben ermöglichen. Gerade bei den zuvor genannten Anwendungsbeispielen ist eine Klasse von Algorithmen besonders notwendig: die Registrierung. Hierbei handelt es sich um ein Verfahren der Computer Vision bzw. Bildverarbeitung, mit dessen Hilfe eine möglichst genaue und sinngemäße Überlagerung zwischen mehreren Bildern errechnet werden kann. Unabhängig davon, ob sich die Kamera oder die Szene bewegt, stimmen die Inhalte zweier zeitlich aufeinanderfolgender Bilder oftmals nicht überein und sind zueinander räumlich versetzt. Die Aufgabe der Bildregistrierung ist es, anhand der Bilddaten zu ermitteln, in welchem Maß sich diese Bildinhalte zueinander verschoben haben.

Mit dem Ergebnis der Registrierung können die Bilder wieder passgenau übereinander gelegt werden. In diesem Fall lassen sich sowohl Karten aus mehreren Bildern als auch Gegenstände, Gebäude und Landschaften über mehrere Aufnahmen hinweg und aus verschiedenen Blickwinkeln zusammensetzen. Das heißt, Objekte werden dreidimensional rekonstruiert und als digitale 3D-Computermodelle konserviert, visualisiert oder in Medien wie Spielen, Filmen und virtuellen Realitäten eingebettet. Überdies können mittels der Registrierung Bewegungen in den Bildern gemessen werden. Anstatt die Bilder zu verschieben, wird hier der Abstand und Versatz der Bildinhalte zueinander berechnet, der nötig wäre, um die Bilder zu überlagern. Dieser Abstand entspricht dann der Bewegung von Objekten im Bild.

Oftmals verwenden die Anwendungen, die 3D-Bildinformationen voraussetzen, im Speziellen die Punktwolkenregistrierungen. Eine 3D-Punktwolke beschreibt die diskret abge-

---

<sup>1</sup>Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung von männlicher und weiblicher Sprachformen verzichtet. Sämtliche Personenbezeichnungen gelten gleichwohl jeglichen Geschlechts.

tasteten Oberflächen- und Raumstrukturen eines Objekts. Das heißt, es handelt sich hierbei um eine punktweise Repräsentation eines Objektes, einer Landschaft oder auch der Körperoberfläche eines Menschen. Die Punktwolke ist daher eine logische Konsequenz aus der Aufnahmetechnik vieler Sensoren. Schließlich können die diskreten Bildpunkte beim Abtasten der Umgebung durch Techniken wie Laser-Entfernungsscanner und Time-Of-Flight-Kameras direkt als Punktwolke dargestellt werden. Des Weiteren ist die Oberflächenstruktur segmentierter medizinischer Bilder wie die der Computertomographie, Magnetomographie oder Ultraschall ebenfalls diskret und lässt sich folglich direkt in Punktwolken konvertieren.

Die Entwicklung eines optimalen Algorithmus zur Punktwolkenregistrierung ist eine permanente Herausforderung, die es zu lösen gilt. Problematisch ist die Tatsache, dass die meisten zu registrierenden Punktwolken nur wenige eindeutige Korrespondenzen aufweisen. Somit ist es für die Algorithmen schwer bis unmöglich eine exakte Überlagerung der Bilder zu finden. Die Verfahren können demnach lediglich versuchen, die möglichst optimale Lösung zu finden. Hinzu kommt, dass die Entwicklung von Kamera- und Sensorsystemen extrem schnelllebig ist. So steigt mit neuen Technologien oder der kontinuierlichen Verbesserung bereits vorhandener Systeme die Auflösung der Punktwolken permanent an. Das Gleiche gilt auch für die Bildraten, mit denen diese Systeme neue Aufnahmen generieren. In diesem Fall müssen fortwährend Strategien entwickelt werden, welche die zunehmende Komplexität der Punktwolkenregistrierung bewältigen können. Letztlich sind es aber auch die Anwendungen selbst, die zunehmend höhere Anforderungen an die Registrierungsalgorithmen stellen. So muss die Genauigkeit der Registrierung stets verbessert werden und gleichzeitig die benötigte Rechenzeit sinken.

In den letzten drei Jahrzehnten wurden daher zahlreiche und verschiedene Verfahren entwickelt, um Punktwolken zu registrieren. Einige haben sich als so mächtig herausgestellt, dass diese kontinuierlich modifiziert und erweitert werden. Trotz der hohen Anzahl an möglichen Lösungsansätzen gibt es kein vollkommenes Verfahren. Es gibt immer wieder Szenarien, bei denen die Algorithmen an ihre Grenzen stoßen und Anwendungsbeispiele, bei denen keine zufriedenstellende Lösung für die Registrierung gefunden werden kann. Daher ist es zwingend notwendig, die potenten bestehenden Verfahren weiter zu entwickeln und gegebenenfalls an bestimmte Anwendungen anzupassen. Aber mindestens genauso notwendig ist es, neue Konzepte zu entwerfen, die das Potential haben, die aktuellen Standards abzulösen.

Aus diesem Grund soll im Rahmen dieser Arbeit ein neues Konzept zur Punktwolkenregistrierung geschaffen werden, wobei sich das neue Paradigma an Realweltphänomenen und physikalischen Gesetzmäßigkeiten orientieren soll. Schließlich wurde in

der Vergangenheit des Öfteren gezeigt, dass die Anwendung von physikalischen Analogien in algorithmische Problemstellungen einen Mehrwert einbringen kann. Ein sehr prominentes Beispiel dafür sind die Thin-Plate-Splines [23]. Ein anderes Beispiel ist die Lösung von Optimierungsproblemen mit Hilfe thermodynamischer Prinzipien: dem Simulated Annealing [12].

**Als Ziel** dieser Dissertation gilt es daher, herauszufinden,

*ob es möglich ist, das Problem der Punktwolkenregistrierung in einer Art und Weise zu formulieren, dass es mit Hilfe von physikalischen Gesetzen gelöst werden kann. Wenn dies möglich ist, soll geprüft werden, ob dieser Ansatz zudem konkurrenzfähig zu bestehenden Standards ist.*

Hierfür werden in Kapitel 2 die nötigen Grundlagen zusammengefasst. Das sind zum einen die formalen Definitionen der beiden Begriffe Punktwolke sowie Registrierung und zum anderen der Überblick über den Stand der Technik. Auf dem Gebiet der Punktwolkenregistrierung zeigt ein Verfahren, der Iterative-Closest-Point-Algorithmus, überdurchschnittlich gute Ergebnisse. Dieser kann als Goldstandard angesehen werden und dient daher als Indikator für die Machbarkeit des neuen Konzepts. Damit die Funktionsweisen beider Algorithmen besser voneinander abgrenzt werden können, wird der Iterative-Closest-Point-Algorithmus am Ende des Grundlagenkapitels 2 genauer erläutert.

Das darauf folgende Kapitel 3 befasst sich mit der Herleitung des neuen Registrierungskonzeptes sowie dessen algorithmische Umsetzung. Dazu wird als erstes eine Analogie zwischen der Punktwolkenregistrierung und physikalischen Gesetzen hergeleitet. Mit dessen Hilfe sollen Punktwolken bewegt werden. Diese Bewegung dient im späteren Verlauf dazu, die Wolken so zu verschieben, dass diese sich optimal überlagern. Damit sich die Punktwolken zudem in die korrekte Richtung bewegen, müssen Abstandsmaße eingeführt werden. Diese Metriken bestimmen während der Registrierung zu jedem Zeitpunkt den Weg zur optimalen Überlagerung. Auch hier werden wieder Analogien zur Physik verwendet.

Die Beschreibung der Punktwolkenbewegung und die Metriken ergeben zusammen das neue Registrierungskonzept. Allerdings muss noch die algorithmische Umsetzung geklärt werden, womit sich der zweite Abschnitt des Kapitels 3 befasst. Wichtig ist dabei, zum einen die Regularisierung der Registrierung. Um die Punktwolkenbewegung auf dem Weg zur optimalen Lösung zu kontrollieren und wenn nötig zu drosseln, wird auch dafür ein Verfahren verwendet, dass sich an physikalischen Gesetzmäßigkeiten orientiert. Zum anderen wird ein Vorschlag unterbreitet, der die Komplexität des Algorithmus

erheblich reduziert. Denn gerade bei hoch aufgelösten Punktwolken sollte der Algorithmus in der Lage sein, möglichst optimale Ergebnisse in angemessener Zeit zu liefern. Schließlich wird noch eine Transformationsdarstellung eingeführt, die zwar in der Punktwolkenregistrierung unüblich ist, aber viele Vorteile bietet. Mit dieser Transformationsdarstellung sollen schlussendlich die Genauigkeit und Reproduzierbarkeit der Registrierung gesteigert werden.

An den Entwurf des Algorithmus zur Punktwolkenregistrierung schließt sich das Evaluierungskapitel 4 an. Hier wird in einer Machbarkeitsstudie untersucht, inwiefern der neue Algorithmus mit dem aktuellen Standard vergleichbar ist. Dazu werden verschiedene Varianten und Implementierungen des Iterative Closest Point Algorithmus dem neuen Verfahren gegenübergestellt. Gemessen werden diese mit Kriterien, die für Registrierungsalgorithmen notwendig sind: Genauigkeit, Reproduzierbarkeit, Effizienz und Robustheit. Außerdem wird eine weitere Eigenschaft überprüft: die Genauigkeitssteigerung durch Merkmalerweiterungen. Hierbei werden die einzelnen Punkte einer Punktwolke um zusätzliche Merkmale erweitert. Das können zum Beispiel die Farben und Helligkeitswerte eines Bildpunktes sein. Solche Merkmale werden nicht von allen Registrierungsverfahren unterstützt, können aber die Performanz erheblich steigern.

Da bei dem hier entwickelten Registrierungsalgorithmus eine Transformationsdarstellung zum Einsatz kommt, die bei den anderen Registrierungsverfahren unüblich ist, muss auch diese näher untersucht werden. Dies erfolgt im zweiten Abschnitt des Evaluierungskapitels 4. Das Vorgehen wird auf seine Effizienz hin analysiert. Außerdem werden weitere Parameter des Algorithmus untersucht, die zu einem gewissen Teil veränderbar sind und somit die Performanz der Registrierung beeinflussen können.

Abschließend wird im letzten Kapitel 5 ein Fazit verfasst, welches die Leitfrage der Arbeit kritisch beantwortet.



## 2 Grundlagen

Um Texturen und Formen von Objekten bzw. Umgebungen zu digitalisieren, werden anders als bei der zweidimensionalen Fotografie keine realen Farbinformationen auf eine Ebene projiziert. Stattdessen werden die Entfernungen bzw. Tiefeninformationen gemessen und abgebildet. Die Momentaufnahme eines Ausschnittes der realen Welt kann auf mannigfaltige Art und Weise generiert werden. In der Regel wird hier zwischen aktiven und passiven Verfahren unterschieden [7, 55, 103, 103].

Ein Großteil der Techniken messen diese Entfernungen aktiv. Das sind Sensor- und Kamerasysteme, die in der Regel ein energetisches Signal aussenden. Dieses wird an den Oberflächen der Szene reflektiert, sodass ein Teil des Signals vom Sensor wieder empfangen werden kann. Die Laufzeit zwischen dem Senden und dem Empfang des Signals wird gemessen und gibt Aufschluss darüber, wie weit die jeweiligen Strukturen vom Sender entfernt sind. Hierzu gehören u. a. Sensortechnologien wie Time-of-Flight, Ultraschall, Radar<sup>1</sup> und Lidar<sup>2</sup> sowie Ladar<sup>3</sup>.

Die Entfernungen können zudem auch passiv ermittelt werden. Dazu gehören Verfahren, bei denen die Entfernungsinformationen aus zweidimensionalen Bildern berechnet werden. Das kann zum Beispiel über die Triangulation zwei definiert versetzter Kameras geschehen (Stereovision). Möglich sind aber auch Rekonstruktionen aus definiert bewegten Bildsensoren, wie dies beispielsweise in der Computer- oder Magnetresonanztomographie geschieht. Aber auch charakteristische Lichtmuster (Structured-Light-Scanner) geben Aufschluss über Entfernungsinformationen. Hierfür werden Lichtmuster in die Szene projiziert. Abhängig von der Beschaffenheit und Entfernung der Objekte, wird das Muster unterschiedlich reflektiert. Über die Abweichung vom Originalmuster und mit Hilfe von Triangulation wird die Entfernung berechnet.

Schließlich werden die so generierten Tiefenbilder sowohl be- als auch verarbeitet und gegebenenfalls für den Anwender visualisiert. Die hierfür verwendeten Verfahren hängen nicht nur von der Anwendung ab sondern auch von der jeweiligen Beschreibungsart der Tiefenbilder. Das heißt, es gibt verschiedene Repräsentationsmöglichkeiten

---

<sup>1</sup>Radio detection and ranging

<sup>2</sup>Light detection and ranging

<sup>3</sup>Laser detection and ranging

[55]. Die sogenannten Punktwolken sind nur eine mögliche Form. So gibt es zum Beispiel zweidimensionale Tiefenbilder bzw. Tiefenkarten. Diese sind vergleichbar zur zweidimensionalen Fotografie. Nur dass die einzelnen Bildpunkte bzw. Pixel nicht die Farben der aufgenommenen Szene speichern, sondern deren Tiefe als Grauwert repräsentieren. Der Vorteil einer solcher Repräsentationsart ist die einfache Verwendung gängiger Bilderverarbeitungs- und -bearbeitungstechniken. Der eindeutige Nachteil ist die wenig intuitive Lesbarkeit der Bilder, da es sich wiederum um eine Projektion der eigentlichen 3D-Szene in eine Ebene handelt.

Intuitiver und besser darstellbar sind sogenannte Volumenbilder [55]. Diese sind eine direkte Repräsentation der 3D-Szene. Ähnlich den zweidimensionalen Bildern werden die Farbinformationen der Szene als einzelne Bildpunkte gespeichert. Die sogenannten Voxel sind das 3D-Äquivalent der Pixel, also Quader mit einheitlicher Kantenlänge in einem räumlich äquidistanten und quadratischen Verbund. Der entscheidende Vorteil dieser Repräsentationsart ist die Verwendung von sogenannten Volumenrendering-Algorithmen. Mit diesen lassen sich die 3D-Szenen realitätsgetreu darstellen und das in Echtzeit inklusive Oberflächenstrukturen, Farben oder auch Schatten. Ohne spezielle Anpassungen ist zum einen der hohe Speicherbedarf nachteilig. Es müssen stets leere Voxel vorgehalten bzw. gespeichert werden, die keine Informationen bezüglich der Szene enthalten, aber räumlich in der Szene liegen. Zum anderen ist die räumliche Vergrößerung der Volumen aufwendiger und besetzt ebenfalls zusätzlichen Speicherplatz ohne Informationsgehalt. Denn immer wenn Bildinformationen dazukommen, die das Volumen vergrößern, reicht es in der Regel nicht aus einen einzelnen Voxel hinzuzufügen, da das Volumen stets quadratisch bleiben muss.

Beliebig erweiterbar und weniger speicherintensiv sind Punktwolken. Hier werden nur die tatsächlich vorhandenen Bildpunkte einer 3D-Szene gespeichert bzw. dargestellt. Jeder Punkt der Wolke repräsentiert somit einen Oberflächenpunkt der Szene und wird räumlich an die entsprechende Stelle im gleichen Koordinatensystem platziert. Die Punkte selbst können dann auch weitere Informationen speichern, wie zum Beispiel die Farbe des dargestellten Objektes. Die Abbildung 2.1 verdeutlicht das Prinzip dieser Repräsentationsart. Zu sehen ist links oben ein Objekt, das mit einer zweidimensionalen Farbkamera aufgenommen wurde. Rechts daneben verdeutlichen die roten Punkte im Farbbild, an welchen Stellen die Tiefenbildkamera das Objekt und somit die entsprechenden Entfernungen gemessen hat. Die Punktwolke ist schließlich im unteren Bild dargestellt und repräsentiert die räumliche Szene in den Koordinaten des Kamerasystems. Die Aufnahme der Punktwolke wurde aus der gleichen Richtung gemacht, wie des Farbbildes. Lediglich die Punktwolke als solche wurde für die Visualisierung gedreht und stellt demnach

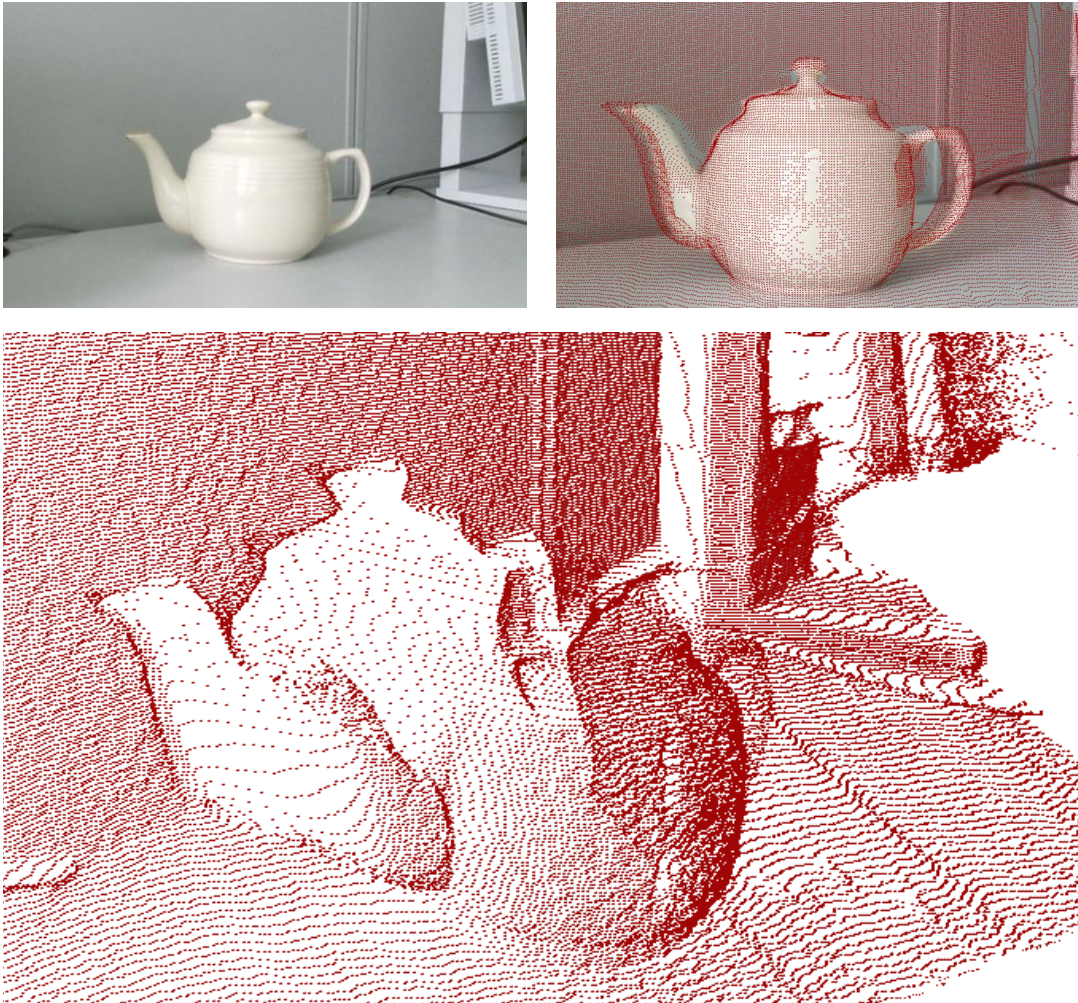


Abbildung 2.1: Exemplarische Darstellung einer Punktwolke (unten) verglichen mit einer Farbaufnahme (oben links) des gleichen Objektes. Die Punkte an denen die Szene mit der Tiefenbildkamera abgetastet wurde ist oben rechts mit der Farbaufnahme überlagert dargestellt.

die Szene aus einem anderen Blickwinkel dar.

Die Abbildung 2.1 verdeutlicht zudem, dass die Darstellung der Punktwolken zwar dreidimensional und für Anwender verständlich ist, für einen besseren Gesamteindruck jedoch die Oberflächenstrukturen fehlen. Deren Berechnung ist aber aufwendiger als beispielsweise bei Volumenbildern. Der Vorteil der Punktwolkenrepräsentationen ist, dass diese beliebig räumlich erweitert werden können. Lediglich bei der Darstellung sind hier

Grenzen gesetzt. Darüber hinaus wird kein unnötiger Speicher belegt. Neuen Punkte müssen einzig an die richtige Koordinate der Wolke platziert werden. Um diese passenden Koordinaten zu finden, an denen die neuen Punkte positioniert werden sollen, müssen beide Punktwolken, die bisher vorhandene und die Menge der neuen Punkte, optimal überlagert werden. Die Grundvoraussetzung dafür ist in der Regel Punktwolkenregistrierung.

Da in den folgenden Kapiteln dieser Arbeit ein neues Verfahren für die Registrierung von Punktwolken vorgestellt und evaluiert wird, sollen in diesem Kapitel die notwendigen Grundlagen geklärt werden. Dazu werden sowohl die Begriffe Punktwolke als auch Punktwolkenregistrierung formalisiert. Danach folgt ein Überblick über den Stand der Technik auf dem Gebiet der Punktwolkenregistrierung. Hierbei sticht ein Verfahren deutlich hervor, der Iterative-Closest-Point-Algorithmus. Aufgrund seiner hohen Akzeptanz kann dieser als Goldstandard angesehen werden. Daher wird der Iterative-Closest-Point-Algorithmus auch als Vergleichswert für die spätere Evaluierung in Kapitel 4 dienen. Die grundlegenden Funktionsweisen und Varianten des Iterative-Closest-Point-Algorithmus werden am Ende dieses Kapitel in Abschnitt 2.2.2 beschrieben.

## 2.1 Punktwolken

Im Folgenden wird eine Punktwolke als Menge<sup>4</sup> von Punkten in der Form

$$\mathcal{X} = (\mathbf{x}_j \in \mathbb{R}^3, 1 \leq j \leq M) \quad (2.1)$$

definiert. Diese Wolke  $\mathcal{X}$  wird im folgenden auch als Modell bezeichnet. Eine zweite Punktwolke

$$\mathcal{Y} = (\mathbf{y}_i \in \mathbb{R}^3, 1 \leq i \leq N) \quad (2.2)$$

heißt Template. Die Vektoren  $\mathbf{y}_i$  und  $\mathbf{x}_j$  beschreiben jeweils die Positionen des  $j$ -ten und des  $i$ -ten Punktes im selben Bezugssystem.

Die Punkte einer Wolke können auch durch weitere Merkmale erweitert werden. Ein solches Merkmal könnte zum Beispiel die Farbe des jeweiligen Punktes sein, repräsentiert durch einen 3D-Farbvektor. Denkbar wäre aber auch ein eindimensionaler (1D) Intensitätswert, der beispielsweise einen Helligkeitswert beschreibt oder auch einen Temperaturwert. Aber auch eine Kombination aus mehreren Merkmalen wie Farben und weiteren lokalen Eigenschaften eines Punktes sind möglich.

---

<sup>4</sup>In der Literatur wird ebenfalls der Begriff Punktmenge (Point Set) äquivalent verwendet.

Solch ein Merkmalsraum wird folgend ebenfalls durch eine Menge beschrieben. Für das Modell gilt dann der Merkmalsraum

$$\mathcal{F}_X = (\mathbf{f}_j^x \in \mathbb{R}^D, 1 \leq j \leq M) \quad (2.3)$$

und für das Template

$$\mathcal{F}_Y = (\mathbf{f}_i^y \in \mathbb{R}^D, 1 \leq i \leq N). \quad (2.4)$$

Somit wird jedem Punkt  $x_j$  das Merkmal  $\mathbf{f}_j^x$  zugeordnet und jedem Punkt  $y_i$  das Merkmal  $\mathbf{f}_i^y$ . Je nach Anzahl der verschiedenen Merkmale eines Punktes kann die entsprechende Dimension  $D$  des Raumes beliebig groß sein. Hierbei wird nur gefordert, dass  $D$  für sämtliche Punkte der zu registrierenden Wolken stets gleich groß ist.

## 2.2 Punktwolkenregistrierung

Die Registrierung zweier Punktwolken ist die Beschreibung ihrer Lagebeziehung zueinander. Zwei Wolken sind genau dann optimal registriert, wenn sich korrespondierende Strukturen in beiden Wolken sinngemäß und optimal überlagern.

Für die Registrierung von Punktwolken gibt es diverse Methoden und Ansätze. Da es sich bei der Registrierung von Punktwolken um ein Optimierungsproblem handelt, ist das Grundprinzip der meisten Verfahren ähnlich: Es muss eine Transformation  $\mathcal{T}$  gefunden werden, die das Template  $\mathcal{Y}$  derart bewegt, dass sich Template und Modell optimal überlagern. Allgemein kann das als Fehlerfunktion

$$\Delta(\mathcal{X}, \mathcal{T}(\mathcal{Y})) \rightarrow \min \quad (2.5)$$

formuliert werden.  $\Delta$  ist ein Abstands- bzw. Fehlermaß und wird Metrik benannt. Diese ist neben der Transformation  $\mathcal{T}$  eines der beiden zentralen Elemente in der Registrierung.

Die Transformation  $\mathcal{T}$  kann beliebig beschrieben werden. Möglich ist zum Beispiel die separate Bewegung eines jeden einzelnen Punktes in der Wolke, sodass das gesamte Template nicht nur verschoben, sondern auch deformiert wird. In dieser Arbeit werden aber ausschließlich rigide Transformationen betrachtet. Das heißt, eine Transformation setzt sich aus einer Translation und einer Rotation zusammen. Skalierende und scherende Transformationen sowie affine werden nicht berücksichtigt. Somit wird jeder Punkt im Template gleichermaßen bewegt. Hierfür wird oftmals eine Kombination aus Rotations- und Translationsdarstellung verwendet, wie zum Beispiel Rotationsmatrizen gepaart mit Translationsvektoren. Aber auch die kompakte Darstellung der Transformation durch homogene Transformationsmatrizen ist in der Punktwolkenregistrierung üblich [18, 24, 92, 96].

### 2.2.1 Stand der Technik

Für die Registrierung von Punktwolken wurden in den letzten Jahrzehnten zahlreiche Verfahren entwickelt und vorgestellt. Für viele dieser Techniken wurden zusätzlich diverse Varianten und Erweiterungen vorgeschlagen, die deren Effizienz verbessern. Darüber hinaus wurden mehrere Publikationen verfasst, die einen Großteil dieser Registrierungsmethoden miteinander vergleichen. Eine detaillierte Auflistung und Analyse sämtlicher Verfahren ist im Rahmen dieser Arbeit nicht möglich. Trotzdem soll der folgende Überblick über die gängigsten Verfahren dabei helfen, das neue Konzept der vorliegenden Arbeit abzugrenzen.

Prinzipiell lassen sich Algorithmen zur Punktwolkenregistrierung in zwei Kategorien unterteilen: grobe und feine Registrierungen. In vielen Anwendungen kommt zudem eine Kombination von grober und feiner Registrierung zum Einsatz. In diesen Fällen dienen die groben Registrierungsverfahren als Initialisierung für die feinen Registrierungen. Das liegt darin begründet, dass grobe Algorithmen mit wenig Rechenzeit hinreichend genaue Ergebnisse produzieren. Feine Registrierungsverfahren hingegen können diese approximierten Vorregistrierungen verwenden, um Punktwolken abschließend präziser zu überlagern.

Üblicherweise berechnen die groben Registrierungsverfahren die Punktwolkenüberlagerung mit Hilfe von sogenannten Landmarken. Das sind bekannte Korrespondenzen zwischen beiden Punktwolken, also ausgewählte Punkte, die in jeweils beiden Wolken vorhanden sind und eindeutig zugeordnet werden können. Die Auswahl der Landmarken kann manuell, semiautomatisch oder komplett automatisch geschehen. Somit setzt sich die Lösung grober Registrierungsprobleme größtenteils aus drei Forschungsbereichen zusammen: Die Erkennung dieser Landmarken, deren Beschreibung und die entsprechenden Überlagerungsstrategien [19]. Da es zwischen zwei Punktwolken häufig nur sehr wenige oder evtl. gar keine eindeutig korrespondierenden Punkte gibt, ist auch die Qualität der Registrierung oftmals ungenauer als bei den feinen Verfahren. Andererseits benötigt die grobe Registrierung aufgrund der bekannten Zuordnung nur einen Iterationsschritt und somit wenig Rechenzeit.

Feine Registrierungsverfahren berechnen hingegen die Überlagerung zweier Punktwolken ohne vorher Landmarken oder Strukturen aus den Bilddaten zu extrahieren. In der Regel sind das iterative Optimierungsverfahren. Sie sind aber auch anfälliger für Fehlregistrierungen durch lokale Optima. Das ist vor allem dann der Fall, wenn die Transformation zwischen den zu registrierenden Punktwolken sehr groß ist. Um solche Fehlerquellen zu minimieren oder gar zu vermeiden, setzen viele der feinen Verfahren die grobe Initialregistrierung voraus. Außerdem kann eine approximative Vorregistrierung die Anzahl an

benötigten Iterationen senken und folglich auch die Laufzeit des Algorithmus. Je kleiner der Abstand beider Punktwolken vor Beginn der Registrierung ist, um so geringer die Rechenzeit.

Der Algorithmus, den diese Arbeit beschreibt, ist der Gruppe der feinen Registrierungen zuzuordnen. Somit muss sich dieser auch mit diesen Vertretern messen. Vergleichende und analysierende Arbeiten zu Verfahren der feinen Punktwolkenregistrierung sind in zahlreichen Publikationen bereitgestellt [2, 3, 19, 61, 82, 90, 91, 98].

Feine Registrierungsalgorithmen arbeiten meist iterativ und bestehen pro Iteration aus zwei Berechnungsschritten. Im Ersten werden Korrespondenzen zwischen den zwei zu registrierenden Punktwolken ermittelt. Im Zweiten wird anhand dieser Korrespondenzen eine der beiden Wolken derart bewegt, dass sich beide ein Stück annähern. Im nächsten Iterationsschritt werden entweder neue Korrespondenzen gesucht oder die bereits bestehenden verwendet, um die Punktwolken erneut zu bewegen. Das geschieht so lange, bis sich beide Wolken möglichst optimal überlagern.

Das bekannteste und am häufigsten verwendete Verfahren zur Punktwolkenregistrierung ist der Iterative-Closest-Point (ICP)-Algorithmus. Wie der Name bereits andeutet, verwendet dieser Algorithmus in jeder Iteration die am nächsten benachbarten Punkte beider Wolken als Korrespondenzen. Die ständige Weiterentwicklung des ICP, dessen zahlreiche Varianten und die hohe Registrierungsgenauigkeit in fast allen Anwendungsszenarien sind der Grund dafür, dass der ICP für die rigide Punktwolkenregistrierung als Goldstandard gewertet werden kann.

Aber die Paarung von Punkten über ihre direkte räumliche Nachbarschaftsbeziehung hat einen entscheidenden Nachteil. So ist der am nächsten benachbarte Punkt nicht zwangsläufig auch der strukturell am Besten korrespondierende Punkt. Das gilt vor allem dann, wenn die beiden zu registrierenden Punktwolken zu weit voneinander entfernt sind. Gerade bei großen rotatorischen Abweichungen ist es sehr unwahrscheinlich, dass die nächsten Nachbarn auch reale strukturelle Punktpaare darstellen. Daher ist der ICP in der Regel auf eine gute bis sehr gute initiale Vorregistrierung angewiesen. Gleiches gilt aber auch für verrauschte oder fehlerbehaftete Punktwolken. Hier kann der am nächsten benachbarte Punkt ebenso ein Ausreißer sein, der entweder keine oder falsche reale Punktkorrespondenz darstellt. Zudem sind Punktwolken problematisch, die sich nur an wenigen räumlichen Stellen überlappen.

Der ICP liefert im Mittel gute Ergebnisse, aber die oben genannten Szenarien erschweren die Registrierung und erhöhen die Wahrscheinlichkeit von Fehlregistrierung. Andere Algorithmen bieten hingegen Alternativen zur Wahl der Punktkorrespondenzen. Einige verwenden zum Beispiel weiche Korrespondenzen zwischen den Punktwolken

[29, 85] oder modellieren die Punktwolken als Kerndichtefunktionen [14, 100]. Dann können die Korrespondenzen über Korrelationen ermittelt werden. Solche Ansätze erhöhen oftmals die Robustheit der Registrierung bei stark verrauschten Punktwolken oder nur wenig überlappenden Punktwolken. Dafür sind diese in der Regel rechenaufwendiger und folglich langsamer als der ICP.

Der Coherent-Point-Drift (CPD)-Algorithmus besticht zum Beispiel durch seine hohe Genauigkeit in der Registrierung und gilt als besonders robust bei Rauschen [66, 67]. Dieser bündelt die Punkte der einen Wolke mit Hilfe von Gaussian Mixture Models. Die Schwerpunkte dieser Cluster werden dann mit den Punkten aus der anderen Punktwolke durch EM<sup>5</sup>-Algorithmen zugeordnet. Der entscheidende Nachteil des CPDs ist seine sehr hohe Laufzeit.

Eine weitere Alternative zum ICP ist der Normal-Distribution-Transform (NDT)-Algorithmus [10, 60]. Hier werden die Punktwolken in kleinere räumliche Zellen unterteilt. Die jeweiligen Zellen wiederum bilden die Normalverteilung aller Punkte in der jeweiligen Zelle ab. Diese Normalverteilung der Punktgruppen liefert somit die Aufenthaltswahrscheinlichkeit der lokalen Oberflächenstruktur in jedem Bereich. Bei großen Zellen zeichnet sich dieser Algorithmus durch eine sehr geringe Laufzeit aus. Einhergehend damit ist aber der Verlust von Registrierengenauigkeiten zu erwarten.

Parallel zu dieser Arbeit haben Golyanik et al. einen Ansatz veröffentlicht, der dem dieser Arbeit in einem Punkt ähnelt [30]. Die Punkte einer Wolke werden als Partikel beschrieben, welche durch das Gravitationsfeld der anderen Punktwolke so beeinflusst werden, dass sich beide Punktwolken anziehen. Diese Modellierung der Metrik ist soweit die einzige Gemeinsamkeit. Die Punktwolkentransformation, die dazu führt, dass sich eine Wolke auf die andere zubewegt, wird ähnlich wie bei anderen gängigen Verfahren über eine Singulärwertzerlegung gelöst. Somit ermöglicht dieses Verfahren eine rigide Punktwolkentransformation inklusive Skalierungen. Die Evaluierung des Algorithmus, im Vergleich zu ICP und CPD, zeigt vielversprechende Ergebnisse. Vor allem bei sehr stark verrauschten Punktwolken liefert das Verfahren konsistent gute Ergebnisse. Die Autoren diskutieren diverse Ansätze, um die Rechenzeit ihres Algorithmus zu senken, da deren derzeitiger Ansatz eine sehr hohe Laufzeit aufweist. So benötigt der Algorithmus ungefähr 1.5 bis 10 Minuten, um Punktwolken mit jeweils 2,000 Punkten zu registrieren. Hinzu kommt, dass die Laufzeit zusätzlich vom Rauschen der Punktwolken negativ beeinflusst wird.

Neben diesen Beispielen gibt es noch diverse Verfahren sowie Erweiterungen und Verbesserungen bereits bestehender Algorithmen. U. a. löst eine Gruppe von Methoden das

---

<sup>5</sup>Expectation-Maximization

Registrierungsproblem mit Hilfe von evolutionären Algorithmen [91]. Zudem gibt es Verbesserungsvorschläge für bestehende und etablierte Verfahren, die diese parallelisieren und somit die Berechnung auf Mehrkernrechenarchitekturen wie Grafikkarten auslagern [4, 25, 34, 53, 57, 84, 99]. Zwar existieren derzeit keine Überblickarbeiten, die diese Parallelvarianten miteinander vergleichen, trotzdem kann der Schluss gezogen werden, dass die Genauigkeit und Robustheit nur wenig bis kaum beeinflusst werden, dafür aber die Laufzeit der Algorithmen erheblich gesenkt wird.

Es gibt demnach keinen Algorithmus, der perfekt auf jedes individuelle Anwendungsszenario angepasst ist. Der rigide Punktwolkenregistrierungsalgorithmus, der in den meisten Vergleichen konstant gut abschneidet und oftmals den anderen Verfahren überlegen ist, ist der **ICP** [2, 5, 6, 86]. Abhängig von der jeweiligen Variante des **ICP** und deren Implementierung liefert dieser gute Registrierungsergebnisse mit sehr geringer Rechenzeit. Das liegt aber vor allem daran, dass der **ICP** eines der ältesten Verfahren ist und bereits von diversen Forschungsgruppen kontinuierlich verfeinert und verbessert wurde. Diese Tatsache und wahrscheinlich auch seine einfache Umsetzbarkeit machen den **ICP** zum Goldstandard, an dem neue Verfahren stets gemessen werden.

Die beiden Varianten des **ICP**, die sich sowohl auf Punkt-zu-Punkt- als auch auf Punkt-zu-Ebenen-Korrespondenzen beziehen, sind im Schnitt am effizientesten. In der Regel ist der Punkt-zu-Ebenen-**ICP** am genauesten. Wohingegen die Punkt-zu-Punkt-Variante weniger anfällig ist, wenn Punktwolken verrauscht oder fehlerhaft sind. Andere Varianten des **ICP** können sich zwar in speziellen Szenarien behaupten, sind aber im Allgemeinen den Punkt-zu-Punkt und Punkt-zu-Ebenen-**ICPs** unterlegen. Dazu gehört beispielsweise der sogenannte Generalized-**ICP**, der Ebenen-zu-Ebenen-Korrespondenzen verwendet [94].

### 2.2.2 Iterative-Closest-Point-Algorithmus

Seit der Veröffentlichung durch die beiden unabhängigen Arbeitsgruppen um Besl und McKay [9] sowie Chen und Medioni [13] sind verschiedenste Varianten des **ICP** entwickelt worden, die u. a. dessen Genauigkeit und Effizienz verbessern [5, 6, 79, 80, 88]. Darüber hinaus wurden die gängigsten Varianten des **ICP** in den verschiedensten Programmiersprachen implementiert und in vielfältige Softwarebibliotheken integriert [15, 28, 50, 51, 80, 81, 89].

Die prinzipielle Funktionsweise des **ICP** lässt sich wie im Flussdiagramm 2.2 zusammenfassen. Nach der Initialisierung, versucht der **ICP** das Registrierungsproblem iterativ mit jeweils zwei Phasen zu lösen:

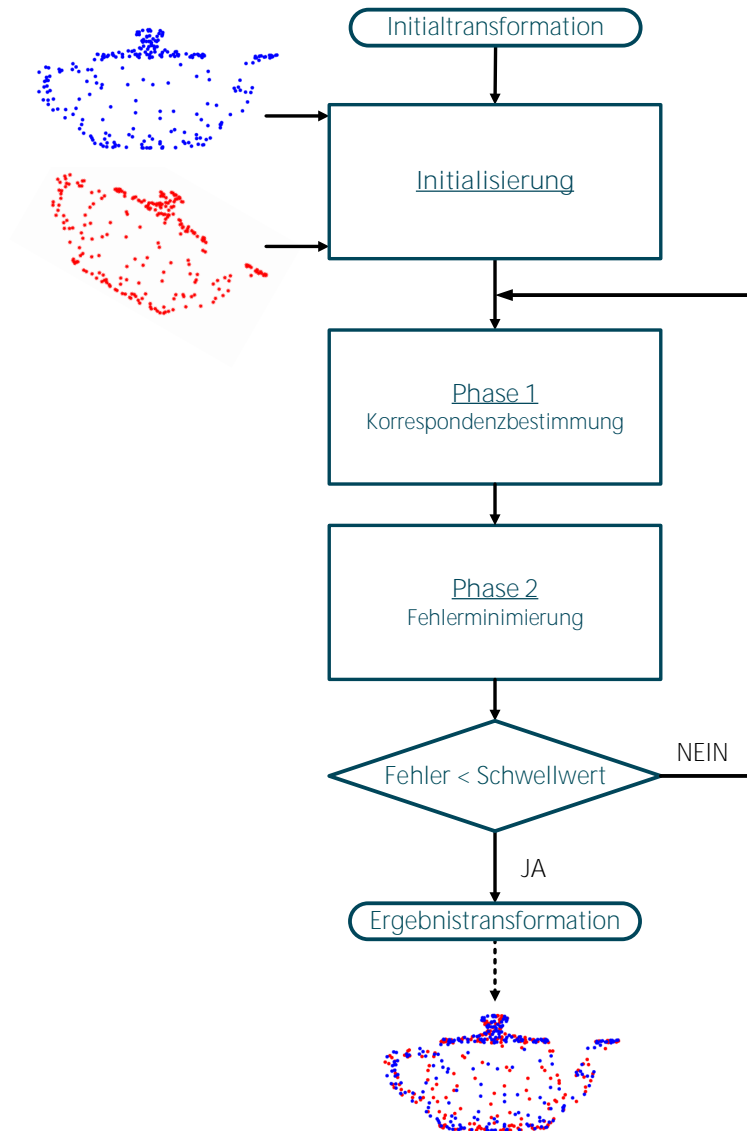


Abbildung 2.2: Flussdiagramm des Iterative-Closest-Point-Algorithmus.

**Phase 1: Korrespondenzbestimmung** In der ersten Phase werden korrespondierende Punktpaare der beiden Wolken gesucht. Wie der Name des Algorithmus suggeriert, wird in jeder Iteration aufs Neue jedem Punkt des Templates der Punkt aus dem Modell zugeordnet, der diesem am nächsten gelegen ist. Der Abstand zwischen den Punkten wird über die Euklidischen Distanz ermittelt. Genauer bedeutet das, dass

$$\forall \mathbf{y}_i \in \mathcal{Y} : \exists \mathbf{x}_j \in \mathcal{X} \mid \min_j \|\mathcal{R} \cdot \mathbf{y}_i + \mathbf{t} - \mathbf{x}_j\|. \quad (2.6)$$

$\mathcal{R}$  ist eine orthogonale  $3 \times 3$ -Matrix und  $t$  ein 1D Translationsvektor. Zusammen beschreiben beide die aktuelle rigide Transformation der Template-Punktwolke. Die hierfür optimale Transformation zu finden, ist Aufgabe der zweiten Phase.

Aus algorithmischer Sicht ist die Suche nach dem nächstgelegenen Nachbarn sehr rechenintensiv. Mit der Brute-Force-Methode, also sprichwörtlich „auf Biegen und Brechen“, müssten alle  $M$  Punkte des Modells mit den  $N$  Punkten des Templates verglichen werden. Damit wäre die Zeitkomplexität dieser Phase in  $\mathcal{O}(M \cdot N)$ , pro Iteration. Es ist offensichtlich, dass diese Lösung ineffizient ist, vor allem wenn die Auflösung beider Punktwolken steigt. Daher werden die Punkte des Modells in der Regel als sogenannter *kD-Baum* repräsentiert.

*kD-Bäume* sind Datenstrukturen, die eine erhebliche Reduzierung der Zeitkomplexität auf  $\mathcal{O}(N \log(M))$  für die Nachbarschaftssuche ermöglichen [62]. Sie zerlegen den Raum um die Datenpunkte in immer kleinere Regionen, bis am Ende jedem Punkt eine Region eindeutig zugeordnet werden kann. Diese Aufteilung ergibt einen Binärsuchbaum, also einen Suchbaum bei dem von jedem Knoten zwei Entscheidungen abgehen. Die Zerlegung der Datenpunkte in einen *kD-Baum* kann auf mannigfaltige Art und Weise geschehen. Am gängigsten ist es, zyklisch durch die  $k$  Dimensionen der Datenpunkte zu permutieren. Dabei entspricht jede Dimension einer Tiefe des Baums. Beginnend mit der Wurzel und der ersten Dimension wird zunächst die komplette Punktwolke in zwei Regionen unterteilt. Die Trennung der Punkte erfolgt in der Regel über den Mittelwert oder den Median. Das ist im zweidimensionalen (2D)-Fall eine Gerade und im 3D-Fall eine Ebene entlang der  $k$ -ten Dimension. Im nächsten Schritt werden jeweils die Punkte der getrennten Regionen anhand der folgenden Dimension unterteilt. Das erfolgt solange, bis nur noch ein Punkt einer Region eindeutig zugeordnet werden kann. Die Blätter des Baums entsprechen dann den jeweiligen Punkten der Wolke bzw. deren nähere Umgebung. In der Abbildung 2.3 ist an einem einfachen 2D Beispiel veranschaulicht wie eine Punktwolke nach diesem Verfahren aufgeteilt wird und als *kD-Baum* gespeichert ist.

Ein Punkt der anderen Wolken kann entsprechend der Raumaufteilung über die Baum-suche seinem nächsten Nachbarn zugeordnet werden. Da der binäre Suchbaum in seiner Höhe balanciert ist, hat die Suche eines einzelnen Nachbarn eine Zeitkomplexität von  $\mathcal{O}(\log(M))$ . Die Konstruktion des *kD-Baums* hat eine Zeitkomplexität von  $\mathcal{O}(M \log(M))$ , muss aber nur ein einziges Mal bei der Initialisierung des ICP-Algorithmus ausgeführt werden. Daher beansprucht die Initialisierung des ICP in der Regel die meiste Rechenzeit.

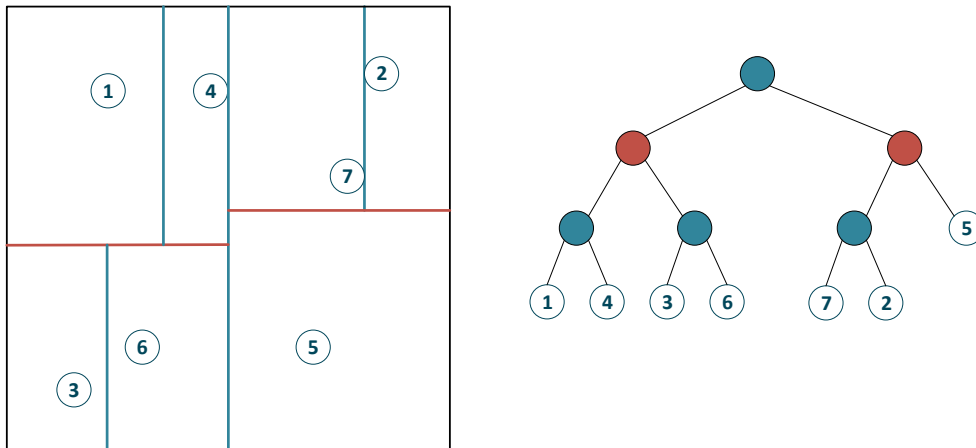


Abbildung 2.3: Exemplarische Konstruktion eines  $k$ D-Baums. Links ist die Unterteilung der 2D-Punktwolke veranschaulicht, bestehend aus sieben Punkten, und rechts der korrespondierende Suchbaum.

Um die Suchzeiten in der ersten Phase des ICP weiter zu senken, werden oft auch mehrere Punkte in einer Region gespeichert, also nicht nur ein Punkt pro Blatt. Das reduziert die Tiefe des Baums und somit auch die Komplexität der Tiefensuche. Dabei hängt die optimale Anzahl an Punkten pro Region stark von der Punktverteilung in der Punktwolke ab [32]. Außerdem ist es möglich, die Suchzeit durch eine approximierte Suche zu beschleunigen. Hierbei wird nicht nach den Punktpaaren mit dem kleinsten Abstand gesucht. Es reicht oftmals aus ein irgendein Punktpaar zu finden, das in einem definierten Toleranzbereich voneinander entfernt ist [32].

Neben diesen gibt es noch weitere Maßnahmen, um die Suchzeit in den  $k$ D-Bäumen zu reduzieren [31, 71, 72, 105]. Trotz der Optimierung durch  $k$ D-Bäume und deren verbesserte Konstruktions- und Suchstrategien bleibt das Verfahren zeitaufwändig. Dies gilt vor allem dann, wenn Punktwolken sehr hoch aufgelöst sind und diese aus mitunter mehreren hunderttausend Punkten pro Wolke bestehen. Daher ist es gängige Praxis, die Auflösung von Punktwolken vor der Registrierung zu reduzieren, zum Beispiel durch eine zufällig gewählte Stichprobe [88]. Der Nachteil ist hierbei: Wenn diese Stichprobe zufällig schlecht gewählt ist und eindeutige Strukturen, die eine Registrierung verbessern würden, herausgefiltert wurden, dann zieht sich dieser Fehler durch die gesamte Registrierung. Vorteilhafter wäre es, in jeder Iteration eine neue Stichprobe zu wählen. Dann müsste aber stets ein neuer  $k$ D-Baum konstruiert werden, was wiederum sehr hohe Laufzeiten nach sich ziehen würde.

**Phase 2: Fehlerminimierung** Die zuvor ermittelten Punktkorrespondenzen bilden nun die Grundlage für die zweite Phase: die Fehlerminimierung. Mit "Fehler" ist der Abstand zwischen den Punktkorrespondenzen gemeint. Das heißt, es muss eine Transformation gefunden werden, die das Template auf das Modell zubewegt, sodass dieser Fehler minimal wird. Hierfür gibt es zwei populäre Ansätze: Einer misst den Fehler direkt über den Punkt-zu-Punkt-Abstand [1, 9], wohingegen der zweite Ansatz den Fehler indirekt über einen Punkt-zu-Ebenen-Abstand berechnet [8, 13, 75].

**Punkt-zu-Punkt-Verfahren** Das Punkt-zu-Punkt-Verfahren ist in Abbildung 2.4 veranschaulicht. Es vergleicht direkt alle Punkte  $y_i$  des Templates mit den jeweiligen korrespondierenden Modell-Punkten  $x_i$ . Damit sich diese beiden Punkte überlagern können, muss der Template-Punkt in Richtung des entsprechenden Modell-Punktes transformiert werden. Solch eine Transformation kann durch

$$\mathbf{x}_i = \mathcal{R} \cdot \mathbf{y}_i + \mathbf{t} \quad (2.7)$$

beschrieben werden. Hierbei ist  $\mathcal{R}$  die Rotation des  $i$ -ten Template-Punktes und  $\mathbf{t}$  die Translation.

Mit Hilfe dieser Lagebeziehung kann im Umkehrschluss auch das Fehlerfunktional

$$\mathbb{F}(\mathcal{R}, \mathbf{t}) = \sum_{i=1}^N \|\mathcal{R} \cdot \mathbf{y}_i + \mathbf{t} - \mathbf{x}_i\| \quad (2.8)$$

formuliert werden. Somit kann der Überlagerungsfehler der beiden Punktwolken über die Summe aller Euklidischen Distanzen zwischen den korrespondierenden Punktpaaren ermittelt werden. Sobald  $\mathbb{F}(\mathcal{R}, \mathbf{t})$  minimal ist, überlagern sich beide Wolken optimal.

Ferner sind

$$\mathbf{x}'_i = \mathbf{x}_i - \mathbf{c}_x, \text{ mit } \mathbf{c}_x = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i, \quad (2.9)$$

die um den Massenschwerpunkt verschobenen Modell-Punkte, die zu den jeweiligen Template-Punkten  $\mathbf{y}_i$  korrespondieren, und

$$\mathbf{y}'_i = \mathbf{y}_i - \mathbf{c}_y, \text{ mit } \mathbf{c}_y = \frac{1}{N} \sum_{i=1}^N \mathbf{y}_i, \quad (2.10)$$

die um den Massenschwerpunkt verschobenen Template-Punkte. Umgeformt und in das

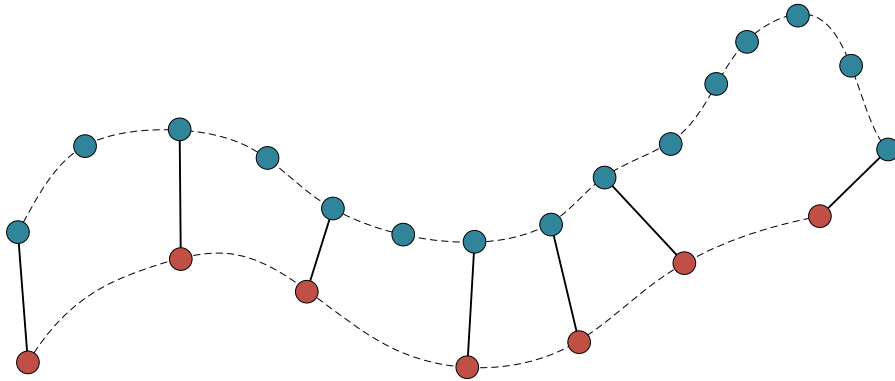


Abbildung 2.4: 2D-Beispiel der Punkt-zu-Punkt-Variante des ICP. Jedem Punkt des Templates (rot) wird genau der Punkt des Modells (blau) mit dem kleinsten euklidischen Abstand zugeordnet. Die Fehlerminimierung erfolgt anhand aller dieser Abstände.

Fehlerfunktional aus Gleichung (2.8) eingesetzt, ergibt sich daraus

$$\begin{aligned} \mathbb{F}(\mathcal{R}, \mathbf{t}) &= \sum_{i=1}^N \|\mathcal{R} \cdot (\mathbf{y}'_i + \mathbf{c}_y) + \mathbf{t} - (\mathbf{x}'_i + \mathbf{c}_x)\| \\ &= \sum_{i=1}^N \|\mathcal{R} \cdot \mathbf{y}'_i - \mathbf{x}'_i + (\mathcal{R} \cdot \mathbf{c}_y + \mathbf{t} - \mathbf{c}_x)\|. \end{aligned} \quad (2.11)$$

Hieraus lassen sich der translatorische und der rotatorische Anteil gesondert extrahieren und berechnen. Die Translation kann mit

$$\mathbf{t} = \mathbf{c}_x - \mathcal{R} \cdot \mathbf{c}_y \quad (2.12)$$

beschrieben werden. Die dafür benötigte Rotation  $\mathcal{R}$  ergibt sich aus der Minimierung von

$$\sum_{i=1}^N \|\mathcal{R} \cdot \mathbf{y}'_i - \mathbf{x}'_i\| \rightarrow \min \quad (2.13)$$

und kann über die folgenden vier Schritte berechnet werden (vgl. [1]):

1. Berechne die  $3 \times 3$ -Matrix

$$H = \sum_{i=1}^N \mathbf{y}'_i \mathbf{x}'_i{}^\top \quad (2.14)$$

2. Ermittle die Singulärwertzerlegung (SVD<sup>6</sup>) von  $H$  mit

$$H = U\Lambda V^T \quad (2.15)$$

3. Berechne

$$\mathcal{R} = UV^T \quad (2.16)$$

4. Wenn für die Determinante von  $\mathcal{R}$  gilt, dass  $\det(\mathcal{R}) = +1$ , dann ist  $\mathcal{R}$  die gesuchte Rotation. Für den Fall, dass  $\det(\mathcal{R}) = -1$ , ist  $\mathcal{R}$  nicht valide und der Algorithmus kann das Ergebnis nicht berechnen.

Allgemein ist dieser Ansatz dahingehend limitiert, dass es in realen Punktwolken sehr unwahrscheinlich ist, dass echte Punkt-zu-Punkt-Korrespondenzen zwischen beiden auftreten. Das liegt vor allem daran, dass sich die Punktwolken aufgrund von Sensorbewegung oder Veränderungen in der Szene nur teilweise überlagern. Aber selbst wenn zwischen zwei Aufnahmen die Sensorposition statisch bleibt und die Szene unbewegt ist, verursacht das natürliche Sensorrauschen Abweichungen in den Punktzuordnungen. In diesen Fällen würden falsche Punktkorrespondenzen zu Fehlregistrierungen führen oder aber zumindest zu einer verlangsamten Konvergenz des Algorithmus. Dann ist es besser, den Abstand über den Punkt-zu-Ebene-Abstand zu minimieren und nicht über den Punktpaaren.

**Punkt-zu-Ebene-Abstand** Auch diese Abstandsminimierung basiert auf den Punktkorrespondenzen aus der ersten Phase. Anstatt jedoch den Abstand zwischen den Punktpaaren zu minimieren, wird hier die Entfernung von einem Template-Punkt zur korrespondierenden Tangentialebene des Modell-Punkts minimiert (siehe Abbildung 2.5).

Wenn  $\mathbf{n}_i = (n_{1i}, n_{2i}, n_{3i})^T$  der Normalenvektor der Tangentialebene auf  $x_i$  ist, dann ergibt sich das Fehlerfunktional zu

$$\mathbb{F}(\mathcal{R}, \mathbf{t}) = \sum_{i=1}^N ([\mathcal{R} \cdot y_i + \mathbf{t} - x_i] \circ \mathbf{n}_i)^2 \quad (2.17)$$

Der Operator  $\circ$  symbolisiert das Skalarprodukt. Demnach leitet sich das Fehlerfunktional aus der Hessischen Normalform von Ebenen ab. Die Normalenvektoren  $\mathbf{n}_i$  auf  $x_i$  können beispielsweise über die lineare Interpolation benachbarter Punkte errechnet werden. Für den 3D-Fall ergeben sich daraus zwei zueinander orthogonale Tangentialvektoren im Punkt  $x_i$ . Über das Kreuzprodukt dieser beiden Vektoren berechnet sich die Normale auf  $x_i$ .

<sup>6</sup>Abkürzung aus dem Englischen: Singular Value Decomposition

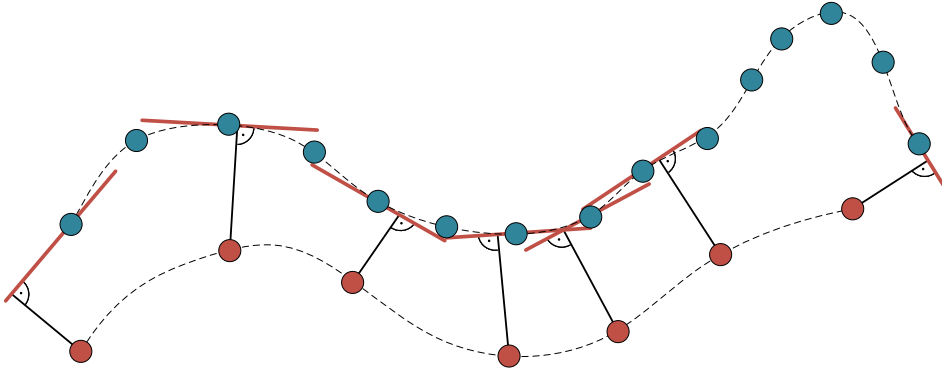


Abbildung 2.5: 2D-Beispiel der Punkt-zu-Ebene-Variante des ICP. Jedem Punkt des Templates (rot) wird genau der Punkt des Modells (blau) mit dem kleinsten euklidischen Abstand zugeordnet. Die Fehlerminimierung erfolgt anhand der kürzesten Abstände zwischen Template-Punkt und Tangentialgerade des korrespondierenden Modell-Punktes.

Wie beim Punkt-zu-Punkt-Abstand zuvor, muss auch hier  $\mathbb{F}$  minimiert werden. Da aber

$$\mathcal{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.18)$$

mit

$$\begin{aligned} r_{11} &= \cos(\theta_y) \cos(\theta_z) \\ r_{12} &= -\cos(\theta_y) \sin(\theta_z) \\ r_{13} &= \sin(\theta_y) \\ r_{21} &= \cos(\theta_x) \sin(\theta_z) + \sin(\theta_x) \sin(\theta_y) \cos(\theta_z) \\ r_{22} &= \cos(\theta_x) \cos(\theta_z) + \sin(\theta_x) \sin(\theta_y) \sin(\theta_z) \\ r_{23} &= -\sin(\theta_x) \cos(\theta_y) \\ r_{31} &= \sin(\theta_x) \sin(\theta_z) - \cos(\theta_x) \sin(\theta_y) \cos(\theta_z) \\ r_{32} &= \sin(\theta_x) \cos(\theta_z) + \cos(\theta_x) \sin(\theta_y) \sin(\theta_z) \\ r_{33} &= \cos(\theta_x) \cos(\theta_y) \end{aligned} \quad (2.19)$$

nicht linear ist, kann in diesem Fall das Optimierungsproblem nicht wie zuvor minimiert werden. Daher wird hier angenommen, dass zwischen den einzelnen Iterationen während der Registrierung nur sehr kleine Rotationsinkremente stattfinden, also für Winkel

$\theta_x \approx \theta_y \approx \theta_z \approx 0$  (vgl. [59]). Für Winkel  $\theta \approx 0$  gilt  $\sin(\theta) \approx \theta$  und  $\cos(\theta) \approx 1$ . Damit ergibt sich die Rotationsdarstellung

$$\mathcal{R} \approx \begin{bmatrix} 1 & -\theta_z & \theta_y \\ \theta_z & 1 & -\theta_x \\ -\theta_y & \theta_x & 1 \end{bmatrix}. \quad (2.20)$$

Jedes  $i$ -te Element des Fehlerfunktionals (2.17) lässt sich mit Hilfe der approximierten Rotation  $\mathcal{R}$  als linearer Ausdruck schreiben:

$$\begin{aligned} [\mathcal{R} \cdot y_i + \mathbf{t} - x_i] \circ \mathbf{n}_i &= [\theta_x (y_{2i}n_{2i} - y_{3i}n_{2i}) \\ &+ \theta_y (y_{3i}n_{3i} - y_{1i}n_{1i}) \\ &+ \theta_z (y_{1i}n_{1i} - y_{2i}n_{3i}) \\ &+ n_{1i}t_1 + n_{2i}t_2 + n_{3i}t_3] \\ &- [x_{1i}n_{1i} + x_{2i}n_{2i} + x_{3i}n_{3i} - y_{1i}n_{1i} - y_{2i}n_{2i} - y_{3i}n_{3i}]. \end{aligned} \quad (2.21)$$

Für die  $1 \leq i \leq N$  korrespondierenden Punkte lässt sich dies auch als lineares Gleichungssystem

$$A\mathbf{z} - \mathbf{b} \quad (2.22)$$

schreiben. Wobei hier

$$\mathbf{b} = \begin{bmatrix} x_{11}n_{11} + x_{21}n_{21} + x_{31}n_{31} - y_{11}n_{11} - y_{21}n_{21} - y_{31}n_{31} \\ x_{12}n_{12} + x_{22}n_{22} + x_{32}n_{32} - y_{12}n_{12} - y_{22}n_{22} - y_{32}n_{32} \\ \vdots \\ x_{1N}n_{1N} + x_{2N}n_{2N} + x_{3N}n_{3N} - y_{1N}n_{1N} - y_{2N}n_{2N} - y_{3N}n_{3N} \end{bmatrix}, \quad (2.23)$$

$$\mathbf{z} = (\theta_x, \theta_y, \theta_z, t_1, t_2, t_3)^\top \quad (2.24)$$

und

$$A = \begin{bmatrix} a_{11} & a_{21} & a_{31} & n_{11} & n_{21} & n_{31} \\ a_{12} & a_{22} & a_{32} & n_{12} & n_{22} & n_{32} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{1N} & a_{2N} & a_{3N} & n_{1N} & n_{2N} & n_{3N} \end{bmatrix}. \quad (2.25)$$

Die Elemente aus  $A$  sind ausgeschrieben

$$\begin{aligned} a_{1i} &= y_{2i}n_{2i} - y_{3i}n_{2i} \\ a_{2i} &= y_{3i}n_{3i} - y_{1i}n_{1i} \\ a_{3i} &= y_{1i}n_{1i} - y_{2i}n_{3i}. \end{aligned} \quad (2.26)$$

Daraus ergibt sich für die Minimierung des Fehlerfunctionals  $\mathbb{F}(\mathcal{R}, \mathbf{t})$ , dass dieses auch als Minimierung des linearen Gleichungssystems (2.22) und als

$$\mathbf{z}_{opt} = \arg \min_{\mathbf{z}} \|\mathbf{A}\mathbf{z} - \mathbf{b}\|^2 \quad (2.27)$$

bestimmt werden kann. Das geschieht zum Beispiel mit Hilfe der SVD aus  $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$  und deren Pseudoinversen  $\mathbf{A}^+ = \mathbf{V}\mathbf{\Lambda}^+\mathbf{U}^T$  (vgl. [59]). Daraus ergibt sich die Lösung zu

$$\mathbf{z}_{opt} = \mathbf{A}^+\mathbf{b}. \quad (2.28)$$

Dabei ist unbedingt zu beachten: Wenn die ersten drei Elemente  $\theta_x, \theta_y$  und  $\theta_z$  aus  $\mathbf{z}_{opt}$  in die approximierte Rotationsmatrix (2.20) eingesetzt werden, dann kann diese eventuell keine zulässige Rotationsmatrix bilden. Daher sollten diese lediglich mit der eigentlichen Rotationsdarstellung verwendet werden (Gleichungen (2.18) und (2.19)).

Da dieses Verfahren die Abstände zwischen Punkten im Template und den Tangentialebenen der korrespondierenden Punkte im Modell optimiert, ist es auch dann genau, wenn echte Punkt-zu-Punkt-Korrespondenzen zwischen den Wolken fehlen (siehe Abbildung 2.5). Andererseits setzt dieses Verfahren eine relativ hohe Punktwolkenauflösung und -dichte voraus, da sonst die Berechnung von Normalen, basierend auf Interpolation benachbarter Punkte, fehleranfällig ist. Hinzu kommt eine größere Anfälligkeit des Verfahrens bei verrauschten Punktwolken, denn das Rauschen wirkt sich ebenfalls stark auf die Normalberechnung der Tangentialebenen aus. Außerdem können aufgrund der approximierten Rotationsmatrix in (2.20), wahrscheinlicher Registrierungsfehler auftreten, wenn die anfänglichen Abstände (besonders hinsichtlich der Rotation) zwischen zu registrierenden Punktwolken zu groß sind.

**Merkmalerweiterung** In den meisten Varianten des ICP werden die Korrespondenzbestimmungen der ersten Phase lediglich über die räumliche Verteilung der Wolken gemessen, also über den kleinsten Abstand benachbarter Punkte. Dieses Verfahren ist zum Beispiel sehr anfällig für Bildrauschen. Um diesem Problem entgegenzusteuern und die Ausreißer zu minimieren, wurde der ICP um diverse Techniken erweitert, angepasst oder verändert. So werden zum Beispiel Punktkorrespondenzen je nach Zugehörigkeit unterschiedlich gewichtet oder gefiltert [27, 83, 88, 104].

Es ist naheliegend, dass nicht nur räumliche Merkmale wie Punktverteilungen oder lokale Oberflächenstrukturen als Punktkorrespondenzen berücksichtigt werden sollten. Sofern vorhanden, können auch weitere Informationen in die Registrierung einfließen. So werden mittlerweile von einigen Tiefenbildkameras zusätzlich auch Farb-, Helligkeits- oder Temperaturwerte simultan aufgezeichnet. Die einzelnen Punkte der Wolken können,

zusätzlich zu ihrer räumlichen Information, um diese Merkmale erweitert werden. Es hat sich gezeigt, dass Registrierungsverfahren, die ebenfalls solche Merkmale verwenden, die Registrierungsgenauigkeit steigern können [47].

Verfahren, die den ICP-Merkmalraum erweitern, verwenden bei der Korrespondenzbestimmung eine zusätzliche Gewichtung der Punktzuschreibungen. Diese richtet sich in der Regel nach Farb- oder Intensitätswerten. Dazu werden die Dimensionen der Punktwolken um die der zusätzlichen Farbinformationen erweitert. So erweitern einige Techniken den Suchraum um höchstens eine Dimension. Diese gewichteten Punktkorrespondenzen werden mit Hilfe des Farbwerts H aus dem HSL-Farbraum bestimmt<sup>7</sup> [64, 65]. Andere erweitern den Suchraum um insgesamt drei Dimensionen für den YIQ-Farbraum<sup>8</sup> [20, 40]. Des Weiteren sind Farbräume wie der  $L^*a^*b^*$ -Farbraum<sup>9</sup> möglich [54].

Ein anderes Verfahren unterteilt den Farbwert H des HSL-Raums in sechs diskrete Bereiche [22]. Basierend auf dieser Unterteilung sind sämtliche Punkte entsprechend ihrer Farbe einer Klasse zugeordnet und bewertet. Die Korrespondenzbestimmung erfolgt dann auf Grundlage dieser Bewertungen. Zudem ist es möglich, die Punktkorrespondenzen ausschließlich über den Farbwert zu bestimmen, ohne räumliche Korrespondenzen zu verwenden [41]. Dazu wurde die Methode von Johnson et al. [40] in der Art und Weise angepasst, dass lediglich die I- und Q-Farbdifferenzen des YIQ-Farbraums zur Bestimmung der kleinsten Abstände zwischen den Punktwolken verwendet werden.

---

<sup>7</sup>Dreidimensionaler Farbraum, bei dem H den Farbwert, S die Sättigung und L die Luminanz beschreiben.

<sup>8</sup>Dreidimensionaler Farbraum, der sich aus der Luminanz Y, dem Farbverhältnis von Cyan und Orange I sowie dem Farbverhältnis zwischen Magenta und Grün Q zusammensetzt.

<sup>9</sup>Dreidimensionaler Farbraum, bei dem die  $L^*$ -Achse die Luminanz angibt. Auf der  $a^*$ -Achse stehen sich Grün und Rot gegenüber und auf der  $b^*$ -Achse Blau und Gelb.



## 3 Algorithmenentwurf

Die Zusammenfassung über den Stand der Technik im vorherigen Kapitel hat gezeigt, dass es etliche Verfahren gibt, die effizient Punktwolken registrieren. Zudem wurden Methoden etabliert, die fortwährend weiterentwickelt und um neue Funktionalitäten ergänzt werden. Dazu gehört zum Beispiel auch der *ICP*, der gegenwärtig als Goldstandard angesehen werden kann. Tatsache ist aber, dass es kein Verfahren gibt, das in der Lage ist das Registrierungsproblem von Punktwolken jederzeit optimal zu lösen. Folglich ist es unabdingbar, sich weiterhin mit der Verbesserung bestehender Algorithmen zu befassen oder diese zumindest an spezielle Anwendungsfälle anzupassen. Aber auch die stete Entwicklung neuer Konzepte ist zwingend notwendig. Schließlich können nur so neue Lösungsvorschläge entstehen, die das Potential haben, bestehende Standards abzulösen.

Solch ein neues Konzept soll im Rahmen dieser Arbeit entwickelt und kritisch untersucht werden<sup>1</sup>. Das eigentliche Grundprinzip der Registrierung bleibt jedoch erhalten und lässt sich wie in Abbildung 3.1 darstellen. Im linken Teil des Flussdiagramms sind die beiden Beispielpunktwolken abgebildet, die registriert werden sollen. Die obere Wolke stellt das Modell dar und die untere das Template. In der Blackbox geschieht die eigentliche Registrierung. Als Ergebnis gibt der Algorithmus die rigide Transformation zurück, die dafür sorgt, dass sich beide Punktwolken sinngemäß und möglichst optimal überlagern. Optional kann eine Initialtransformation vorgegeben werden, die beide Punktwolken vor dem Start der Registrierung zueinander vorteilhaft ausrichtet. Idealerweise sollte der Algorithmus aber ohne diese initiale Ausrichtung der Punktwolken auskommen.

Das neue Konzept setzt sich aus mehreren Bestandteilen zusammen. In diesem Kapitel werden die Module sukzessiv hergeleitet und beschrieben. So wird der Algorithmus in den folgenden Abschnitten und Teilabschnitten um weitere Funktionalitäten erweitert. Das Flussdiagramm in Abbildung 3.1 unterstützt die Erklärung visuell, da auch dieses am Ende der einzelnen Teilabschnitte um die entsprechende Funktionalität ergänzt wird. Das Kapitel schließt mit einem funktionsfähigen Algorithmus, der das neue Konzept zur

---

<sup>1</sup>Teile dieses Kapitels wurden bereits vorab veröffentlicht [38].

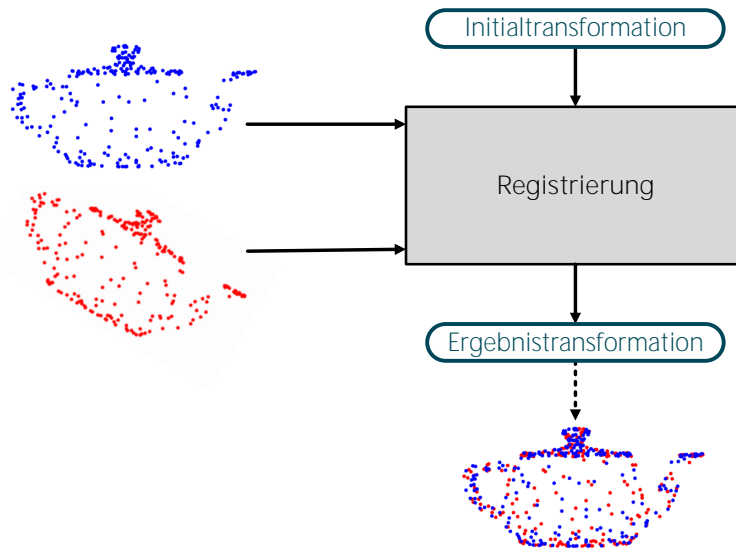


Abbildung 3.1: Flussdiagramm zum neuen Registrierungsalgorithmus. Abhängig von einer möglichen Initialregistrierung berechnet der Algorithmus die Transformation, die das Template (blau) auf das Modell (rot) registriert.

Punktwolkenregistrierung so implementiert, dass es in Kapitel 4 mit dem Goldstandard verglichen und einer umfassenden Analyse unterzogen werden kann.

Da das neue Konzept auf physikalischen Gesetzen basieren soll, werden zunächst entsprechende Analogien hergeleitet und auf das Registrierungsproblem von Punktwolken abgebildet. Das geschieht im folgenden Abschnitt 3.1. Es werden zwei Parallelen zu physikalischen Modellen gezogen, sodass sich eine Punktwolke rigide auf die andere zubewegen kann.

Anschließend werden in Abschnitt 3.2 die Metriken beschrieben, die den Abstand zwischen zwei Punktwolken messen. Daraus ergibt sich die Richtung, in der sich die eine Punktwolke bewegen muss, um eine optimale Überlagerung zu erreichen. Dabei sind auch die Metriken physikalischen Gesetzen nachempfunden und komplementieren die Analogie aus dem vorherigen Abschnitt.

Die ersten beiden Abschnitte beschreiben bereits den eigentlichen Registrierungsprozess: Eine Punktwolke bewegt sich in die Richtung, in der eine optimale Überlagerung beider zu erwarten ist. Das ist aus algorithmischer Perspektive unzureichend. Damit das Verfahren auch implementiert und mit anderen Algorithmen verglichen werden kann, müssen noch weitere Ergänzungen vorgenommen werden. Dazu gehört zum einen die Regularisierung der Punktwolkenbewegung. Diese verhindert, dass die Bewegungen zu

groß sind und über das Ziel hinausschießen bzw. zu klein sind und der Algorithmus zu weit vor dem Optimum terminiert. Zum anderen gehört die Reduktion der Komplexität zu den notwendigen Ergänzungen. Das Verfahren muss demnach so implementiert werden, dass es auch bei sehr hoch aufgelösten Punktwolken eine geringe Laufzeit besitzt. Eine dritte und letzte Ergänzung ist die Transformationsbeschreibung. Die richtige Wahl dieser Bewegungsrepräsentation hat gleichermaßen Einfluss auf die Genauigkeit und Geschwindigkeit der Registrierung. Diese drei Feinheiten zur Implementierung des Algorithmus werden im letzten Abschnitt 3.3 vorgestellt.

### 3.1 Punktwolken analog zur Physik

Festkörper, also Stoffe im festen Aggregatzustand, bestehen aus Teilchen, die dreidimensional und raumfest angeordnet sind. Abhängig von der Bindungsart und Anordnung der Einzelteilchen können Festkörper kaum bis gar nicht verformt werden. Zum Beispiel besitzen Metallgitter oder kristalline Anordnungen oftmals sehr starke Bindungen. Bewegt sich solch ein Festkörper im Raum, gilt dies für sämtliche Einzelteilchen gleichermaßen. Eine relative Bewegung der Teilchen im Körper selbst findet so gut wie nicht statt. So ist eine Deformation des Körpers ausgeschlossen und die Bewegung ist rigide. Bis auf die regelmäßige Anordnung kristalliner Stoffe lassen sich diese starren Verbindungen direkt mit rigiden Punktwolken vergleichen. Denn auch diese bestehen aus mehreren Punkten, die gewissermaßen eine feste Bindung zueinander haben.

Mit Hilfe der klassischen Mechanik kann die Bewegung von Festkörpern beschrieben werden. Immer dann, wenn Kräfte auf den Körper einwirken, verursacht dies eine Bewegungsänderung des Körpers. Auf kosmische Dimensionen bezogen ist es vorwiegend die Gravitationskraft, die zwischen massereichen Objekten anziehende Bewegungen hervorruft. Je geringer der Abstand zwischen zwei Objekten ist, desto stärker werden die Anziehungskräfte. Ähnliches gilt für einen kleinen Maßstab. Hier beschreibt zum Beispiel die Elektrostatik Kräfte, die gleichermaßen vom Abstand punktförmiger Ladungsträger abhängig sind. Aber an Stelle der Masse treten in diesem Fall Ladungen. Sind die Ladungsträger zweier Objekte verschieden, dann ziehen sich beide an. Bei gleichen Ladungen kommt es zu abstoßenden Kräften.

In beiden Beispielen sorgen die anziehenden bzw. abstoßenden Kräfte dafür, dass sich entweder eines der Objekte oder beide in Bewegung setzen. Sowohl in der klassischen Mechanik als auch in der Elektrostatik werden diese Objekte als punktförmig angenommen. Es sind demnach Kugeln, die theoretisch keine Größe besitzen. Außerdem lassen sich in beiden Fällen die Kräfte zusammenfassen, die durch den Einfluss

mehrerer Punkte gleichzeitig entstehen. Solche Kraftfelder können dann Bewegungen vollständiger Systeme bzw. ganzer Körper beschreiben.

Da die Punkte in einer Punktwolke ebenfalls kugelförmig sind und keine bestimmte Größe haben, bilden diese Überlegungen das Grundgerüst des neuen Registrierungskonzeptes. Die folgenden Teilabschnitte beschreiben sowohl das physikalische Modell des Mehrteilchensystems als auch das der Starrkörpertransformation. Basierend auf diesen beiden Modellen der Physik ist es schließlich möglich, ein Verfahren zu beschreiben, das eine Punktwolke rigide und unter dem Einfluss von Kräften bewegt. Die Ursache der Kräfte ist ebenfalls von physikalischen Phänomenen inspiriert und wird im folgenden Abschnitt 3.2 im Detail hergeleitet und bezüglich der Punktwolkenregistrierung umformuliert.

### 3.1.1 Mehrteilchensysteme

Das Mehrteilchensystem ist ein Modell der klassischen Mechanik. Mit dessen Hilfe können Systeme beschrieben werden, deren Einzelteilchen in Wechselwirkung stehen und somit das komplette System beeinflussen. Es bildet mikroskopische Prozesse auf makroskopische Systeme ab und kommt immer dann zum Einsatz, wenn die Zahl der Einzelteilchen so hoch ist, dass die Berechnung sämtlicher Individuen zu komplex und unzweckmäßig ist [70].

Angenommen eine Punktwolke ist rigide, d.h. die Punkte in der Wolke bewegen sich nicht relativ zueinander, sondern nur im Verbund des Gesamtsystems Wolke, dann ist die Punktwolke vergleichbar mit einem starren Körper, der aus mehreren Teilchen fester Zuordnung besteht. Dies ist vergleichbar mit der festen Gitterstruktur der Atome in Metallen, nur meist ohne die gleichmäßige Anordnung. Auch wenn die Teilchen keine Relativbewegungen innerhalb des Systems aufweisen, stehen diese in einer kontinuierlichen Wechselwirkung zueinander und zu ihrer Umwelt. Das Resultat sind Kräfte, die auf jedes Einzelteilchen einwirken.

Solche Kräfte haben oftmals eine Bewegungsänderung des Systems zur Folge. Das Modell des Mehrteilchensystems abstrahiert die einzelnen Wechselwirkungen, fasst diese bzw. ihre resultierenden Kräfte zusammen und bündelt sie in einem einzigen Massenschwerpunkt. Das heißt, ein Mehrteilchensystem besteht aus  $N$  Einzelteilchen. Das  $i$ -te Teilchen ist durch seine Masse  $m_i$  und seine Position  $\mathbf{y}_i \in \mathbb{R}^3$  charakterisiert. Auf jedes einzelne dieser Teilchen wirkt eine Kraft  $\mathbf{F}_i \in \mathbb{R}^3$ . Diese setzt sich aus einer externen

$\mathbf{F}_i^{(ex)}$  und einer internen Kraft  $\mathbf{F}_{ij}^{(in)}$  zusammen:

$$\mathbf{F}_i = \mathbf{F}_i^{(ex)} + \sum_{\substack{j=1 \\ i \neq j}}^N \mathbf{F}_{ij}^{(in)}. \quad (3.1)$$

$\mathbf{F}_{ij}^{(in)}$  beschreibt die Interaktion zwischen dem  $i$ -ten und dem  $j$ -ten Teilchen innerhalb desselben Systems. Die externe Kraft  $\mathbf{F}_i^{(ex)}$  wiederum wirkt von außen auf das  $i$ -te Teilchen.

Die Gleichung (3.1) kann mit Hilfe des dritten Newtonschen Axiom vereinfacht werden. Dieses besagt, dass Kräfte stets paarweise auftreten und auf jede Aktion eine Reaktion folgt:

$$\mathbf{F}_{ij}^{(in)} = -\mathbf{F}_{ji}^{(in)}. \quad (3.2)$$

Außerdem beeinflusst ein Teilchen sich nicht selbst, also

$$\mathbf{F}_{ii}^{(in)} = 0. \quad (3.3)$$

Sämtliche Teilchen des Gesamtsystems können nach dem Schwerpunktsatz durch

$$\sum_{i=1}^N \mathbf{F}_i = \sum_{i=1}^N \mathbf{F}_i^{(ex)} + \sum_{\substack{i=1 \\ i \neq j}}^N \sum_{\substack{j=1 \\ i \neq j}}^N \mathbf{F}_{ij}^{(in)} \quad (3.4)$$

zusammengefasst werden. Unter Berücksichtigung des dritten Newtonschen Axioms ist die interne Kraft des Gesamtsystems mit

$$\sum_{\substack{i=1 \\ i \neq j}}^N \sum_{\substack{j=1 \\ i \neq j}}^N \mathbf{F}_{ij}^{(in)} = \frac{1}{2} \sum_{\substack{i=1 \\ i \neq j}}^N \sum_{\substack{j=1 \\ i \neq j}}^N \left( \mathbf{F}_{ij}^{(in)} + \mathbf{F}_{ji}^{(in)} \right) = 0. \quad (3.5)$$

Das heißt, die interne Kraft hat keinen Einfluss mehr und die Gleichung (3.1) lässt sich vereinfachen. Die Gesamtkraft

$$\mathbf{F} = \sum_{i=1}^N \mathbf{F}_i = \sum_{i=1}^N \mathbf{F}_i^{(ex)}, \quad (3.6)$$

welche auf das System einwirkt, ist lediglich von der externen Kraft abhängig, die ihren Ursprung außerhalb des Systems hat (vgl. [70]).

Die Anzahl der Teilchen ist oft so hoch, dass die Berechnung aller Einzelkräfte zu rechenintensiv ist. Gerade für starre Körper, in denen keine Relativbewegung der Einzelteilchen stattfindet, wäre dies unzweckmäßig. So wäre es zum Beispiel unvorteilhaft die

Bewegung einer rollenden Kugel anhand jedes einzelnen Atoms zu beschreiben. Praktikabler ist es daher, die Teilchen mit Hilfe des Schwerpunktsatzes zusammenzufassen und die globale Position der Kugel als Ganzes anzugeben. So werden für das mikroskopische Mehrteilchensystem die Massenpunkte aller Teilchen in einem einzigen Massenschwerpunkt zusammengefasst. Hieraus lassen sich viele Aussagen bezüglich der Einzelteilchen auf das makroskopische Gesamtsystem ableiten.

Der Massenschwerpunkt des System ist durch

$$\mathbf{c} = \frac{1}{G} \sum_{i=1}^N m_i \cdot \mathbf{r}_i \quad (3.7)$$

definiert, wobei  $G = \sum_{i=1}^N m_i$  die Gesamtmasse des Systems beschreibt. In diesem Massenschwerpunkt greift nun die Gesamtkraft an, die sich aus der Superposition sämtlicher Einzelkräfte ergibt (siehe Gleichung(3.6)). Die Wirkung dieser Kraft hat wiederum einen Einfluss auf die Bewegung des Körpers. Da in dieser Arbeit ausschließlich rigide Systeme betrachtet werden, kommt es durch diese Kräfteinwirkungen weder zu einer Verformung noch zu einer Deformation des Körpers. In der klassischen Mechanik werden die Bewegungen eines rigidem Mehrteilchensystems durch das Modell der Starrkörpertransformation formuliert. So muss auch hier nicht die Bewegung jedes einzelnen Teilchens beschrieben werden, sondern lediglich die des Gesamtsystems.

### 3.1.2 Starrkörpertransformation

Ein Starrkörper ist ein fester bzw. rigider Körper bestehend aus mehreren Teilchen. Diese besitzen keine Relativbewegung im Bezugssystem des Körpers. Das heißt, der Körper kann nicht verformt werden. Wirkt eine Kraft  $\mathbf{F}$  auf den Körper, dann bewegt sich dieser als Gesamtsystem. Dabei setzt sich die Bewegung aus Translationen und Rotationen zusammen.

Die Kraft  $\mathbf{F}$ , die von außen auf den Körper wirkt, leitet sich aus der Gleichung (3.6) des Mehrkörpersystems ab. Sie greift im Massenschwerpunkt  $\mathbf{c}$  an, der infolgedessen eine lineare und rotatorische Beschleunigung erfährt. Die lineare Beschleunigung  $\mathbf{a}$  des Körpers ergibt sich aus dem zweiten Newtonschen Axiom und wird durch

$$\mathbf{F} = \mathbf{a} \cdot G, \quad (3.8)$$

beschrieben, wobei  $\mathbf{a}$  der Beschleunigungsvektor und  $G$  die Gesamtmasse des Körpers ist. Eingesetzt in die Funktion der gleichmäßig beschleunigten Bewegung

$$\mathbf{x}(t) = \frac{1}{2} \mathbf{a} \cdot t^2 + \mathbf{v}_0 \cdot t + \mathbf{x}_0 \quad (3.9)$$

ergibt sich der Weg bzw. Bewegungsvektor  $x(t)$  in Abhängigkeit der Zeit  $t$ . Hier ist  $v_0$  die Anfangsgeschwindigkeit und  $x_0$  der Anfangsweg, jeweils zum Zeitpunkt  $t_0 = 0$ .

Ähnliches gilt für die Rotationsbewegung. Mit Hilfe der Winkelbeschleunigung  $\alpha$  lässt sich der Winkel  $\sigma$  über die Gleichung der gleichmäßig beschleunigten Rotation berechnen:

$$\sigma(t) = \frac{1}{2}\alpha \cdot t^2 + \omega_0 \cdot t + \sigma_0. \quad (3.10)$$

Auch hier ist der Winkel  $\sigma(t)$ , mit dem sich der Körper um seinen Massenschwerpunkt  $c$  dreht, von der Zeit  $t$  abhängig.  $\omega_0$  sowie  $\sigma_0$  sind die Anfangswinkelgeschwindigkeit und die Anfangswinkel zum Zeitpunkt  $t_0 = 0$ .

Sowohl die lineare Bewegung als auch der Rotationswinkel sind Funktionen in Abhängigkeit der Zeit  $t$ . In realen Systemen ermöglicht dies eine Beschreibung kontinuierlicher Bewegungen. Das Registrierungsverfahren ist jedoch iterativ und somit diskret. Daher können die Bewegungsgleichungen nur diskret behandelt werden. Da später die algorithmische Komplexität möglichst gering gehalten werden soll, muss die Implementierung zum einen möglichst wenig Rechenoperationen ausführen, zum anderen den Speicherverbrauch gering halten. Das heißt, anstatt kontinuierliche Bewegungen zu beschreiben, sollen die Anfangszustände der Bewegungsgleichung vernachlässigt werden. Daraus ergibt sich eine iterative Bewegung, die in jedem Schritt aus der Ruhe beginnt. So müssen keine Gegenkräfte modelliert werden, wie zum Beispiel Reibungen, die dafür sorgen, dass die Punktwolken nicht bis ins Unendliche beschleunigen.

Alle Annahmen, die getroffen werden, um eine Punktwolkenbewegung von den realen physikalischen Modellen zu unterscheiden, sollen die algorithmische Komplexität reduzieren und können wie folgt zusammengefasst werden:

1. Die Bewegung pro Iterationsschritt hat stets die zeitliche Länge  $\Delta t = 1$ .
2. Zu Beginn einer Bewegung befindet sich das System in absoluter Ruhe am Nullpunkt. Das heißt,  $\{v_0, x_0, \omega_0, \sigma_0\} = \mathbf{0}$ .
3. Da es sich bei der Registrierung nicht um real-physikalische Vorgänge handelt, wird auf den Erhalt von physikalischen Einheiten verzichtet.

Unter Berücksichtigung dieser drei Annahmen lassen sich die beiden Bewegungsgleichungen (3.9) und (3.10) erheblich vereinfachen. Für die lineare Bewegung gilt dann

$$x = \frac{1}{2}a \quad (3.11)$$

und für die rotatorische Bewegung

$$\sigma = \frac{1}{2}\alpha. \quad (3.12)$$

Daraus folgt, dass die lineare Bewegung lediglich von der Beschleunigung  $\boldsymbol{a}$  abhängt. Nach der Gleichung (3.8) ist  $\boldsymbol{a}$  wiederum direkt proportional zur Kraft  $\boldsymbol{F}$ , die im Massenschwerpunkt angreift. Folglich ist die Translation des Systems bereits gegeben. Anders verhält es sich mit der Rotationsbewegung  $\boldsymbol{\sigma}$ . Diese ergibt sich aus den externen Kräften, die auf jedes einzelne Teilchen einwirken und kann folgendermaßen bestimmt werden:

Die Gesamtrotation eines starren Körpers hängt von sämtlichen externen Kräften  $\boldsymbol{F}_i^{(ex)}$  ab, die auf das jeweilige  $i$ -ten Teilchen wirken. Diese Kraftereinwirkungen haben einen Einfluss auf das externe Drehmoment  $\boldsymbol{\tau}_i^{(ex)}$ . Das Drehmoment wiederum bewirkt die Rotation des Körpers und ist durch

$$\boldsymbol{\tau}_i^{(ex)} = \boldsymbol{l}_i \times \boldsymbol{F}_i^{(ex)} \quad (3.13)$$

definiert, dem Kreuzprodukt aus externer Kraft und dem Hebelarm

$$\boldsymbol{l}_i = \boldsymbol{y}_i - \boldsymbol{c} \quad (3.14)$$

zwischen Massenschwerpunkt und  $i$ -tem Teilchen.

Die Superposition aller externen Drehmomente des Körpers hat eine direkte Auswirkung auf die erste Ableitung des Drehimpulses  $\boldsymbol{L}$  nach der Zeit

$$\frac{d}{dt} \boldsymbol{L} = \sum_{i=1}^N \boldsymbol{\tau}_i^{(ex)}. \quad (3.15)$$

Unter der Annahme, dass Bewegungen nur sukzessiv stattfinden und diese eine zeitliche Länge von  $\Delta t = 1$  besitzen, ergibt sich daraus der Drehimpuls

$$\boldsymbol{L} = \sum_{i=1}^N \boldsymbol{\tau}_i^{(ex)}. \quad (3.16)$$

Schließlich ist der Einheitsvektor

$$\hat{\boldsymbol{L}} = \frac{\boldsymbol{L}}{\|\boldsymbol{L}\|} \quad (3.17)$$

des Drehimpulses die Rotationsachse des starren Körpers, der sich auf den Massenschwerpunkt stützt.

Der noch fehlende Rotationswinkel lässt sich direkt aus den Trägheitsmomenten  $J_i$  der Einzelteilchen berechnen. Das Trägheitsmoment

$$J_i = m_i \|\boldsymbol{y}_i - \boldsymbol{c}\|^2 \quad (3.18)$$

des  $i$ -ten Teilchen hängt sowohl von der Masse  $m_i$  als auch von dem Euklidischen Abstand zwischen Teilchen und Massenschwerpunkt des Systems ab. Nach dem Superpositionsprinzip ergibt sich das Trägheitsmoment des Gesamtsystems zu

$$J = \sum_i J_i. \quad (3.19)$$

Aus dem zweiten Newtonschen Gesetz berechnet sich die Winkelbeschleunigung zu

$$\alpha = \frac{J}{L}. \quad (3.20)$$

Diese kann schließlich in die rotatorische Bewegungsgleichung (3.12) eingesetzt werden. Dann ist

$$\theta = \|\sigma\| \quad (3.21)$$

der Winkel, der die Rotation um die Drehachse  $\hat{L}$  beschreibt.

Hiermit sind alle Komponenten, die eine schrittweise Transformation des rigiden Mehrteilchensystems beschreiben, durch die lineare Bewegung  $x$  und die Rotation mit dem Winkel  $\theta$  um die Achse  $\hat{L}$  gegeben. Diese Parameter können schließlich in eine entsprechende Transformationsdarstellung übertragen werden.

Die schrittweise Bewegungsbeschreibung und die generelle Notwendigkeit mehrerer Iterationen pro Registrierung fordert, dass die Einzeltransformationen  ${}^{k-1}\mathcal{T}_k$  miteinander verknüpft werden müssen.  $k$  ist der Index der aktuellen Iteration und  $K$  die Anzahl aller Iterationen. Mit Hilfe der zweiten Annahme, dass sich das System zu Beginn jeder Bewegung in absoluter Ruhe befindet, kann die Gesamttransformation der Punktwolke diskret durch

$${}^0\mathcal{T}_K = {}^0\mathcal{T}_1 \otimes {}^1\mathcal{T}_2 \otimes \dots \otimes {}^{K-1}\mathcal{T}_K \quad (3.22)$$

beschrieben werden. Der Operator  $\otimes$  definiert die Verknüpfung zweier aufeinander folgender rigider Transformationen. Dieser ist abhängig von der jeweiligen Transformationsdarstellung. Welche Repräsentationsart schlussendlich gewählt wird, ist von der Implementierung abhängig und kann prinzipiell frei gewählt werden. Möglich sind alle Bewegungsrepräsentationen, die sich direkt oder über Umwege aus einem Translationsvektor und der Achse-Winkel-Beschreibung einer Rotation darstellen lassen.

Schlussendlich sind durch die Adaption der beiden physikalischen Modelle Mehrteilchensystem und Starrkörpertransformation die ersten Bestandteile des neuen Verfahrens gegeben. Beide bilden die essentielle Grundlage der Methodik. Das Flussdiagramm in Abbildung 3.2 zeigt, wo diese beiden Module im Algorithmus einzuordnen sind. Gegeben sind die beiden Punktwolken und eine optionale Initialtransformation. Pro Iteration werden zwischen den einzelnen Punkten beider Wolken Kräfte berechnet. Diese hängen von der jeweilig aktuellen Lagebeziehung beider Punktwolken zueinander ab. Das Modell  $\mathcal{X}$  bleibt stets ortsfest und erfährt somit keine Bewegung. Die Kräfte wirken nur auf das Template  $\mathcal{Y}$  ein. Mit Hilfe des Mehrteilchensystems werden die externen Kräfte zusammengefasst und im Massenschwerpunkt der Punktwolke zentriert. Die resultierende Kraft bewirkt eine Bewegung der Punktwolke, die analog zur Starrkörpertransformation

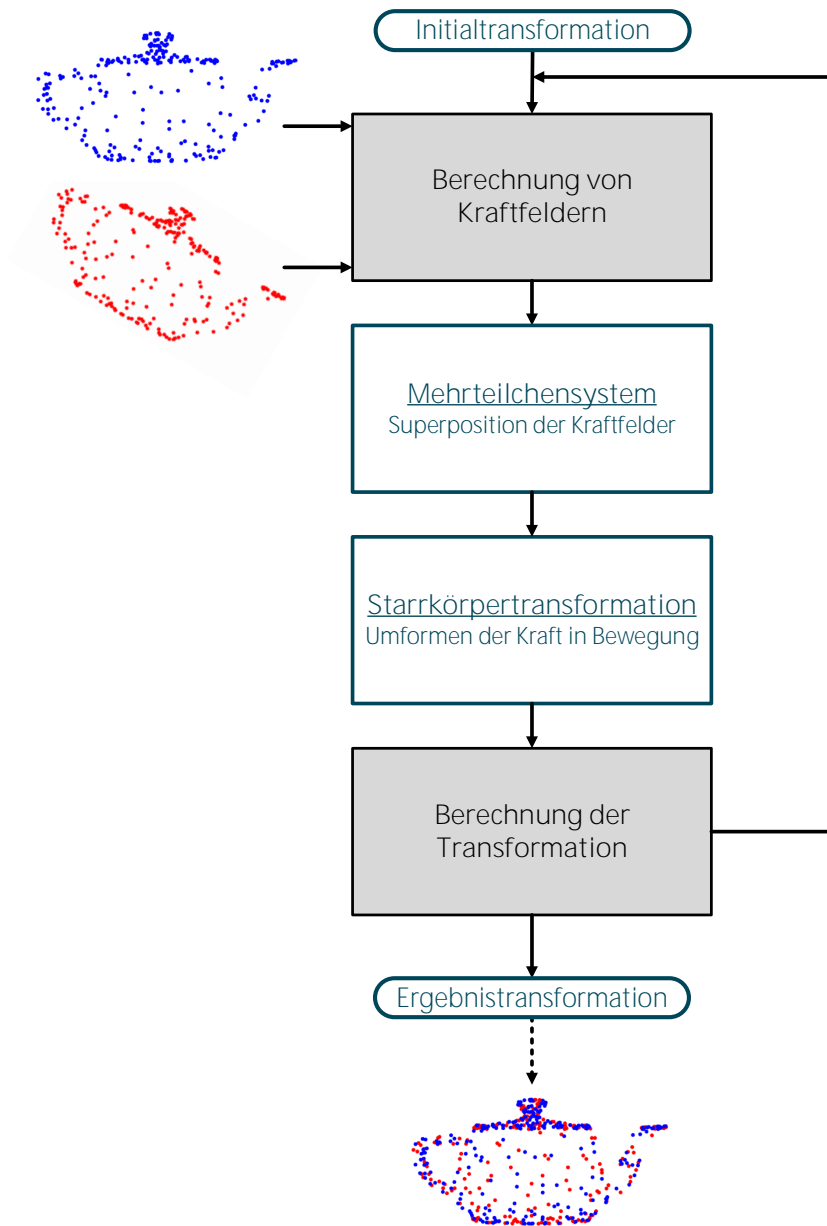


Abbildung 3.2: Das Flussdiagramm des neuen Registrierungsalgorithmus erweitert um zwei neue Bestandteile. Das Mehrteilchensystem berechnet die Superposition zwischen den Kraftfeldern beider Punktwolken. Anhand der resultierenden Kraft, die auf den Massenschwerpunkt des Templates wirkt, wird analog zur Starrkörpertransformation eine Bewegung der Punktwolke berechnet.

beschrieben wird. Das Ergebnis ist eine Transformation pro Iterationsschritt. Diese lässt sich mit den vorhergegangenen verknüpfen, sodass eine sukzessiv zunehmende Bewegung des Templates in Richtung des Modells entsteht.

Damit die Registrierung gelingen kann, müssen noch die beiden ausgegrauten Bestandteile des Flussdiagramms aus Abbildung 3.2 erarbeitet werden. Das ist zum einen die Beschreibung der Kraftmodelle, das heißt die Metriken der Registrierung. Diese geben an, in welche Richtung sich das Template bewegen muss. Die Metriken sind ebenfalls physikalischen Gesetzen nachempfunden und werden im folgenden Teilabschnitt 3.2 beschrieben. Zum anderen steht noch die Berechnung der Transformation aus. Da letztere jedoch stark von der Repräsentationsart abhängt und thematisch zum Teil der algorithmischen Implementierung gehört, wird diese erst in Abschnitt 3.3 genauer thematisiert.

## 3.2 Metriken analog zu Kraftfeldern

Im vorherigen Abschnitt wurde eine Analogie zwischen Modellen der klassischen Mechanik und der Punktwolkentransformation hergeleitet. Die adaptierte Beschreibung des Mehrteilchensystems und die der Starrkörpertransformation ermöglichen eine rigide Bewegung des Templates mit Hilfe von Kraftfeldern. Diese Punktwolke kann sich solange auf das Modell zubewegen, bis sich beide möglichst optimal überlagern. Solch eine Bewegung beinhaltet sowohl Translation als auch Rotation und geschieht schrittweise. Nur die Richtung, in der sich das Template bewegen muss, ist bis hierhin nicht geklärt. Das ist die Aufgabe von Metriken, die im Folgenden beschrieben werden.

Das neue Konzept sieht vor, dass die Metrik den Abstand zur optimalen Punktwolkenüberlagerung misst und diesen wiederum als externe Kraft  $\mathbf{F}_i^{(ex)}$  repräsentiert. Diese Kraft wirkt auf das entsprechende  $i$ -te Teilchen des Templates  $\mathcal{Y}$  ein. Grundlegend ist diese Kraftmodellierung frei wählbar und kann dementsprechend an unterschiedliche Anwendungen angepasst werden, ohne den kompletten Algorithmus neu entwerfen zu müssen. Für den Anwender bleibt das Grundprinzip der Registrierung transparent und die Effizienz des Algorithmus ändert sich nur marginal. Beim Entwurf einer neuen Metrik müssen lediglich zwei notwendige Bedingungen eingehalten werden:

1. Ein anziehende Kraft  $\mathbf{F}_i^{(ex)}$ , die auf einen Punkt  $\mathbf{y}_i$  im Template wirkt, muss stets in Richtung der bestmöglichen Übereinstimmungen in  $\mathcal{X}$  zeigen. Ferner sind auch abstoßende Kräfte zulässig. Diese müssen von möglichst ungleichen Punktepaarungen weg zeigen.
2. Der Betrag des Kraftvektors  $\|\mathbf{F}_i^{(ex)}\|$  muss konstant sein oder mit der Registrie-

rungsqualität zunehmen.

Das bedeutet, Metriken können ebenfalls von der Physik inspiriert sein, müssen es aber nicht. Möglich sind auch Abstandsmaße, die keine oder nur eine teilweise Ähnlichkeit zu Realweltphänomenen haben, solange diese sich durch Kraftvektoren darstellen lassen.

In dieser Arbeit werden drei Beispielmetriken vorgestellt. Genauso wie die Punktwolkentransformation beruhen auch diese auf physikalischen Gesetzen. Eine der Metriken ist vom Gravitationsgesetz inspiriert und modelliert Kräfte ausschließlich über die räumlich Lagebeziehung beider Punktwolken. Die anderen beiden Metriken basieren auf dem Coulombschen Gesetz. Neben der räumlichen Information werden hier auch weitere Merkmale gewichtend hinzugezogen. Dies ermöglicht zusätzlich abstoßende Kräfte zwischen den Punktwolken.

### 3.2.1 Newtons Gravitationsgesetz

Die erste Metrik ergibt sich ebenfalls aus der Analogie zur klassischen Mechanik. In diesem Fall ist es das Newtonsche Gravitationsgesetz. Dieses Gesetz beschreibt Anziehungskräfte zwischen Objekten in Abhängigkeit ihrer Masse und deren Abstand zueinander, wie zum Beispiel die Anziehungskräfte zwischen einem Planeten und der Sonne. Je näher sich die Himmelskörper kommen, desto mehr nimmt die Anziehung zwischen ihnen zu. Gleichermaßen steigt auch die Anziehungskraft mit zunehmender Masse. Jeden Himmelskörper umgibt demnach ein Gravitationsfeld, das mit zunehmender Entfernung zum Objekt abnimmt. Sind zwei oder mehrere Objekte nah genug beieinander, akkumulieren sich deren Gravitationsfelder zu einem Gesamtkraftfeld.

Das Newtonsche Gesetz modelliert diese Anziehungskräfte und vereinfacht die Realität so, dass diese berechenbar wird. Denn unabhängig von der Art der Himmelskörper sowie deren Größe, Form und chemischer Zusammensetzung reduziert das Gesetz diese auf einzelne Punktmassen. Diese Punkte haben lediglich eine Masse und eine Ortskoordinate, an der sie sich im Raum aufhalten. Somit werden die Kraftfelder und Bewegungen der Himmelskörper lediglich durch punktförmige Objekte beschrieben. Auch an dieser Stelle sind die Parallelen zur Punktwolkenregistrierung offensichtlich. Schließlich werden beim Registrierungsproblem von Punktwolken ebenfalls Punkte ohne Form und Größe betrachtet. Außerdem sollen sich die korrespondierenden Punkte zunehmend stark anziehen, sobald sich diese näher kommen, bis sie sich optimal überlagern.

Ausgehend von dieser Analogie werden alle Punkte der Wolken  $\mathcal{X}$  und  $\mathcal{Y}$  als Punktmassen im gleichen Bezugssystem betrachtet. Die Gravitationskräfte zwischen Punkten derselben Wolken sind gleich Null. Somit wird jeder Punkt des Templates  $y_i$  nur durch

das externe Gravitationsfeld beeinflusst, welches von  $\mathcal{X}$  erzeugt wird. Dem Newtonschen Gesetz zu folge ist

$$\mathbf{F}_i^{(ex)} = -\gamma \sum_{j=1}^M \frac{m_i m_j}{\|\mathbf{y}_i - \mathbf{x}_j\|^3} (\mathbf{y}_i - \mathbf{x}_j) \quad (3.23)$$

die externe Kraft, die auf jeden Punkt  $\mathbf{y}_i$  wirkt.  $\gamma$  ist die Gravitationskonstante.  $m_i$  bzw.  $m_j$  steht für die jeweiligen Massen der Punkte  $\mathbf{y}_i$  und  $\mathbf{x}_j$ .

Für das Registrierungsproblem wird angenommen, dass die Massen aller Punkte stets  $m_i = m_j = 1$ . Folglich sind die Massen im Vergleich zu kosmologischen Gravitationsfeldern sehr klein. Gleiches gilt auch für die räumliche Ausdehnung bzw. den Abstand beider Punktwolken zueinander. Aus diesen Gründen wird die Gravitationskonstante ebenfalls auf  $\gamma = 1$  gesetzt. Das vereinfacht die Berechnung des Kraftfelds zu

$$\mathbf{F}_i^{(ex)} = - \sum_{j=1}^M \frac{(\mathbf{y}_i - \mathbf{x}_j)}{\|\mathbf{y}_i - \mathbf{x}_j\|^3}. \quad (3.24)$$

Die adaptierte Beschreibung des Newtonschen Gesetzes, angewandt auf Punktwolken führt dazu, dass Kräfte auf alle einzelnen Punkte des Templates wirken. Diese sind so gerichtet, dass sich nahe liegende Punktepaare beider Wolken einander stärker anziehen als solche, die weiter voneinander entfernt sind. Diese Korrespondenzbeschreibung ist somit ähnlich der des ICP. Doch im Gegensatz zum neuen Ansatz wird beim ICP pro Iteration jedem Punkt aus dem Template genau ein Modell-Punkt zugeordnet, nämlich der jeweils Nächstliegende. Die Punktkorrespondenz des ICP ist somit Eins-zu-Eins. Bei der hier vorgestellten Metrik beeinflussen sämtliche Punkte des Modells jeden einzelnen Template-Punkt. Folglich ist die Punktkorrespondenz Eins-zu-Viele. Das soll lokale Optima während des Registrierungsprozesses umgehen und schließlich die Registrierungsgenauigkeit steigern. Zudem besteht die Möglichkeit, dass sich durch den Einfluss mehrerer Punkte auf einen einzelnen, Ausreißer und Rauschen weniger stark auf die Registrierung auswirken und das Verfahren robuster als der ICP wird.

### 3.2.2 Coulombs Gesetz

Die zuvor beschriebene Metrik basierend auf dem Newtonschen Gesetz berücksichtigt ausschließlich die räumlichen Strukturen der Punktwolken. Das heißt, nah beieinander liegende Punkte beider Wolken ziehen sich stärker an, als Punktepaare, die weiter voneinander entfernt sind. Dabei spielen weder Form, Größe oder Maße eine Rolle. Merkmalserweiterungen wie Helligkeitswerte oder Farben werden nicht berücksichtigt. Solche Merkmale sind aber besonders dann von Bedeutung, wenn die Punktwolken wenig

räumliche Strukturen aufweisen. Genau dann ermöglichen die zusätzlichen Informationen trotzdem differenzierbare Korrespondenzen zwischen den Punktwolken.

Auch uneindeutige räumliche Zuordnungen können mit Hilfe von Merkmalen vermieden werden. Ein Minimalbeispiel in Abbildung 3.3 zeigt genau solch ein Verhalten. Hier sind zwei Punktwolken vereinfacht und zweidimensional dargestellt. Die Abbildung 3.3 a) zeigt die Punktwolken vor der Registrierung, wobei die Modell-Punkte anhand von Kreisen und die Template-Punkte durch Sterne repräsentiert sind. Beide Wolken weisen nur wenig räumliche Strukturen auf und überlagern sich teilweise. Werden diese Punktwolken ausschließlich mittels ihrer Strukturen registriert, sieht das zu erwartende Ergebnis wie in Abbildung 3.3 b) aus. Strukturell überlagern sich die beiden Punktwolken annähernd optimal. Doch die Überlagerung ist sinngemäß falsch und spiegelt sich in der falschen Zuordnung der Farbwerte wieder. Werden die Farben bei der Registrierung mitberücksichtigt, könnte das Ergebnis wie in Abbildung 3.3 c) aussehen. Folglich sind sowohl die räumlichen Strukturen als auch die erweiterten Merkmale der Wolke für eine optimale Registrierung notwendig.

Die beiden folgenden Beispielmetriken berücksichtigen die räumliche Verteilung der Punkte sowie deren zusätzlichen Merkmale. Beide Metriken leiten sich vom Coulombschen Gesetz ab und sind daher ebenfalls physikalisch motiviert. Für die erste Metrik wird das Coulombsche Gesetz so interpretiert, dass lediglich Anziehungskräfte beschrieben werden. Die zweite ähnelt eher dem physikalischen Vorbild, da hier die Merkmale, neben den Anziehungskräften, auch abstoßende Kräfte erzeugen können.

Das Coulombsche Gesetz beschreibt die elektrostatische Kraft zwischen zwei punktförmigen Ladungsträgern bzw. die Kraftfelder zwischen mehreren Ladungsträgern. Ähnlich den Gravitationsfeldern berücksichtigt das Coulombsche Gesetz die Abstände zwischen den Ladungsträgern. Der Unterschied besteht darin, dass die Massen durch Ladungen der Punkte und die Gravitationskonstante durch die elektrische Konstante ersetzt werden. Im Gegensatz zu den Massen im Newtonschen Gesetz können Ladungen auch negativ sein. Deshalb ist es möglich, dass sich Ladungsträger voneinander abstoßen [21].

Auch hier können wieder die zu registrierenden Punktwolken mit Wolken von Ladungsträgern gleichgesetzt und das Modell der Elektrostatik auf die Punktwolkenregistrierung angewandt werden. In diesem Fall wirkt das elektrostatische Feld, das durch die Modell-Punktwolke  $\mathcal{X}$  erzeugt wird, auf jeden  $i$ -ten Punkt des Templates  $\mathcal{Y}$ . Die dabei entstehen-

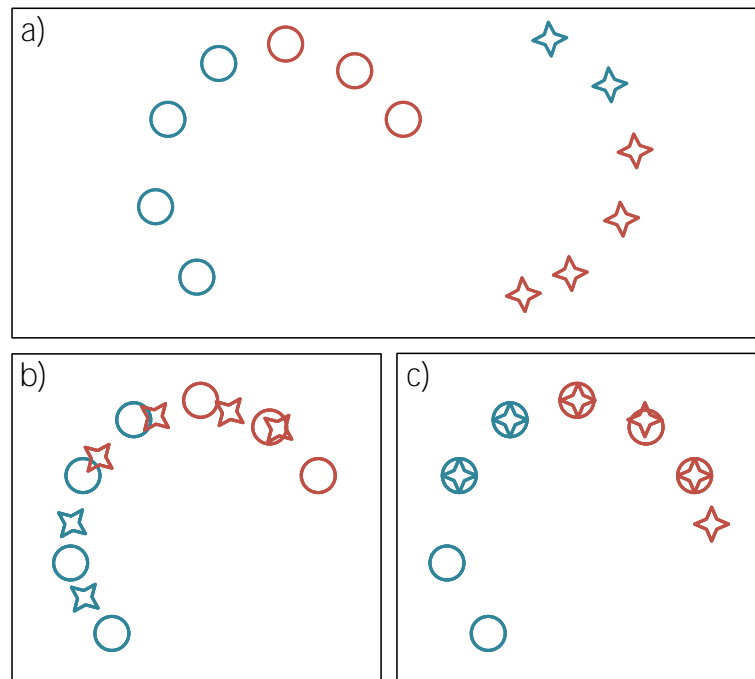


Abbildung 3.3: 2D Minimalbeispiel, das die Notwendigkeit von Merkmalerweiterungen bei der Punktwolkenregistrierung demonstriert. a) Zwei eingefärbte Punktwolken Modell (Kreise) und Template (Sterne) sollen registriert werden. b) Werden lediglich räumliche Strukturen berücksichtigt, liefert die Registrierung ein sinngemäß falsches Ergebnis. c) Verwendet die Registrierung zusätzliche Farbmerkmale, ist ein optimales Ergebnis möglich.

de Kraft berechnet sich aus

$$\mathbf{F}_i^{(ex)} = \kappa \sum_{j=1}^M \zeta_i \zeta_j \frac{(\mathbf{y}_i - \mathbf{x}_j)}{\|\mathbf{y}_i - \mathbf{x}_j\|^3}. \quad (3.25)$$

Hier ist  $\kappa = \frac{1}{4\pi\epsilon_0}$  und  $\epsilon_0$  die elektrische Konstante. Das heißt,  $\kappa$  ist konstant. Da die räumliche Verteilung der Punktwolken  $\mathcal{X}$  und  $\mathcal{Y}$  im Verhältnis zu realen elektrischen Ladungswolken sehr groß ist, wird für das Registrierungsproblem  $\kappa = 1$  angenommen.  $\zeta_i$  und  $\zeta_j$  beschreiben jeweils die Ladungen der Punkte  $\mathbf{y}_i$  bzw.  $\mathbf{x}_j$ . Nach dem Gesetz von Coulomb sind Ladungen entweder positiv oder negativ. Das heißt, Punkte mit gleichen Ladungen stoßen einander ab. Punkte mit unterschiedlicher Ladung ziehen sich wiederum an. Der Betrag der anziehenden bzw. abstoßenden Kraft ist, wie beim Gravitationsgesetz, indirekt proportional zum Abstand.

Damit die Merkmale  $\mathcal{F}_\mathcal{X}$  und  $\mathcal{F}_\mathcal{Y}$  als Ladungen interpretiert und miteinander verglichen werden können, wird der Wertebereich der Merkmalswerte stets auf den Bereich

$[0, 1]$  skaliert. Außerdem sollten beide Merkmalsräume die gleiche Dimension besitzen. Mit diesen Forderungen kann das Coulombsche Gesetz als folgende Metrik interpretiert werden:

$$\mathbf{F}_i^{(ex)} = \sum_{j=1}^M \Delta_i \frac{(\mathbf{y}_i - \mathbf{x}_j)}{\|\mathbf{y}_i - \mathbf{x}_j\|^3}. \quad (3.26)$$

Der Vorfaktor  $\Delta_i$  bewertet die Gleichheit der Merkmale beider Punktwolken. Aber im Gegensatz zu den physikalischen Ladungen im Coulombschen Gesetz sollten Merkmale, die sich ähnlich sind, stärker anziehen als solche, die verschieden sind. Hieraus lassen sich zwei Varianten dieser Metrik ableiten:

In der ersten Variante werden die Kräfte entsprechend des Merkmalsabstands skaliert. So treten nur Anziehungskräfte auf und es gilt:

$$\Delta_i = 1 - \frac{\|\mathbf{f}_i^y - \mathbf{f}_j^x\|}{\sqrt{D}}, \quad (3.27)$$

mit  $\|\mathbf{f}_i^y - \mathbf{f}_j^x\|$  dem Euklidischen Abstand zwischen den Merkmalen  $\mathbf{f}_i^y, \mathbf{f}_j^x \in [0, 1]^D$ . Es folgt, dass auch  $\Delta_i \in [0, 1]$  ist und der räumlichen Abstand zwischen den entsprechenden Punktkorrespondenzen über den Merkmalsunterschied skaliert wird. Das heißt, stimmen die Merkmale zweier Punkte überein, wird der räumliche Abstand höher gewichtet und fließt dementsprechend stärker in die Kraftberechnung ein. Bei Punkten, deren Merkmale verschieden sind, hat die räumliche Korrespondenz weniger bis keinen Einfluss auf die Anziehungskraft.

Die zweite Variante beschreibt mit

$$\Delta_i = 2 \left( 0.5 - \frac{\|\mathbf{f}_i^y - \mathbf{f}_j^x\|}{\sqrt{D}} \right) \quad (3.28)$$

auch abstoßende Kräfte. Wie bei der ersten Variante ist  $\|\mathbf{f}_i^y - \mathbf{f}_j^x\|$  die Euklidische Distanz zwischen den Merkmalen  $\mathbf{f}_i^y \in [0, 1]^D$  und  $\mathbf{f}_j^x \in [0, 1]^D$ . Gegensätzlich ist aber, dass  $\Delta_i \in [-1, 1]$ . Folglich werden die räumlichen Abstände zwischen korrespondierenden Punkten nicht nur skaliert. Punktkorrespondenzen, bei denen sich die Merkmale sehr stark unterscheiden, modellieren abstoßende Kräfte.

Mit Hilfe dieser drei Beispielmetriken lässt sich der Algorithmus zur Punktwolkenregistrierung weiter vervollständigen. Wie im Flussdiagramm der Abbildung 3.4 zu erkennen ist, können nun durch diese Metriken Kräfte generiert werden, welche die sinngemäßen Abstände zwischen Punktwolken als Kraftfelder darstellen. Im zweiten Schritt werden die Kräfte mit Hilfe des Mehrteilchensystems im Massenschwerpunkt des Templates gebündelt. Die daraus resultierende Kraft bewirkt eine Bewegung der Punktwolke. Diese wird

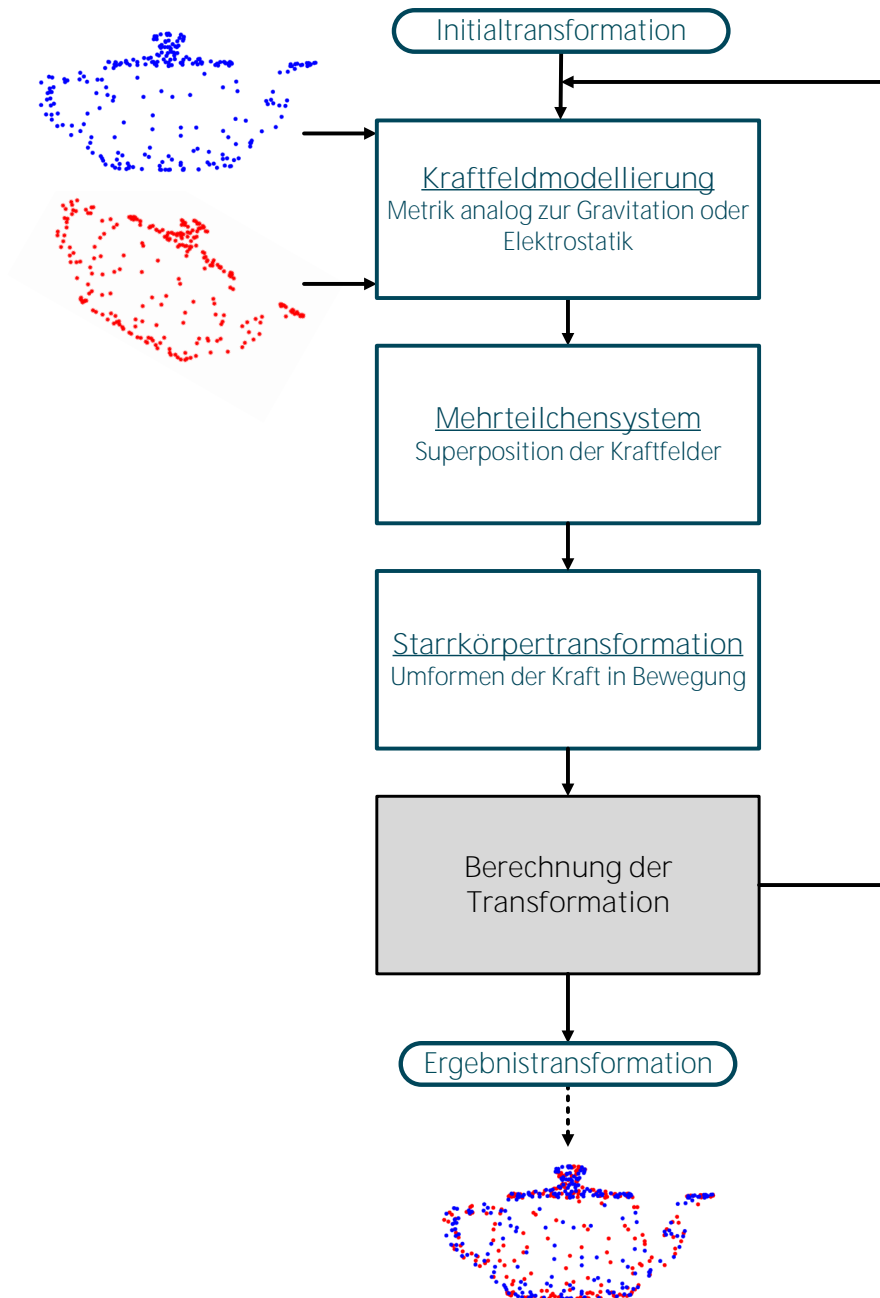


Abbildung 3.4: Das Flussdiagramm des neuen Registrierungsalgorithmus erweitert um die Kraftfeldmodellierung. Die Kraftfelder sind die Metrik der Registrierung und berechnen sich analog zum Newtonschen Gravitationsgesetz bzw. Coulombschen Gesetz der Elektrostatik.

im dritten Bestandteil des Algorithmus anhand der Starrkörpertransformation berechnet. Wird diese Befehlsabfolge über mehrere Iterationen wiederholt, nähert sich das Template allmählich dem Modell an.

Inhalt des Abschnittes 3.3 ist die detaillierte Beschreibung der Bewegung sowie die Bedingung, bei der das iterative Verfahren terminiert. Neben diesen beiden Aspekten wird die Reduktion der algorithmischen Komplexität erläutert.

### 3.3 Implementierung

Bis hierhin wurde die Punktwolkenregistrierung als Analogie zu physikalischen Kraftfeldern abgeleitet. Dieser Zusammenhang modelliert Punktwolken als Mehrteilchensysteme zwischen denen Anziehungs- und Abstoßungskräfte herrschen. Die Kräfte wiederum hängen sowohl von der räumlichen Verteilung als auch von den zusätzlichen Merkmalen der einzelnen Punkte ab und greifen im Massenschwerpunkt der Template-Wolke an. Daraus folgt eine Bewegung des Templates in Richtung des statischen Modells. Diese Bewegung ist rigide und leitet sich aus dem Modell der Starrkörpertransformation ab. Die Anziehungs- und Abstoßungskräfte ergeben sich aus den Abstandsmaßen, die sämtliche Punkte beider Wolken berücksichtigen. Diese Metriken können beliebig modelliert werden und müssen sich demnach auch nicht zwingend aus physikalischen Gesetzmäßigkeiten ergeben. Wichtig ist nur, dass zwei Forderungen an die Kraftmodellierung eingehalten werden: Erstens müssen die Kraftbeträge mit der Qualität der Überlagerung zunehmen oder zumindest konstant bleiben. Zweitens müssen anziehende Kräfte stets in Richtung hoher Punktkorrespondenzen zeigen. Werden zudem abstoßende Kräfte berücksichtigt, dann müssen diese von schlechten Punktkorrespondenzen weg zeigen.

Es wurden drei Beispielmetriken beschrieben. Eine beruht auf dem physikalischen Prinzip der Newtonschen Gravitationsfelder. Hier werden lediglich die räumlichen Strukturen der Punktwolken berücksichtigt. Die anderen beiden Abstandsmaße schließen zusätzlich Merkmalerweiterungen mit ein. Beide sind eine Interpretation des Coulombschen Gesetzes, wobei erstere nur den Betrag der Anziehungskräfte zwischen korrespondierenden Punkten skaliert. Bei hoher Merkmalsgleichheit werden die Abstandsbeziehungen der Punkte stärker gewichtet. Sind die Merkmale verschieden, verringert sich der Betrag bzw. wird sogar Null. Die letztere Metrik erlaubt neben den gewichteten Anziehungskräften auch abstoßende Kräfte. Das heißt, bei Ungleichheit der Merkmale wirkt die Kraft umso stärker in entgegengesetzte Richtung je ungleicher die Merkmale sind.

Damit ist das grundlegende Prinzip des neuen Konzeptes zur Punktwolkenregistrierung erklärt. Um den Algorithmus aber implementieren zu können und ihn später in Ka-

pitel 4 mit dem Goldstandard zu vergleichen, ist diese Beschreibung ungenügend. Aus algorithmischer Sicht sind noch drei entscheidende Probleme zu lösen:

Erstens steigen die Kräfte direkt proportional zur Qualität der Registrierung oder sind zumindest gleichbleibend. Da, im Gegensatz zu den realen physikalischen Systemen die Punkte der Wolken nicht aus Materie bestehen, kommt es zu keiner Kollision. Das wiederum bedeutet, dass bei zu großen Kräften die Punktwolkenbewegung über das Ziel hinausschießt. Im extremsten Fall führt das zu einer fortwährenden Oszillation um das Optimum. Außerdem fehlt dem Algorithmus ein Abbruchkriterium. Da das Verfahren iterativ ist, sollte die Registrierung zu dem Zeitpunkt gestoppt werden, wenn die Überlagerung möglichst optimal ist. Idealerweise stellen Metriken die nötigen Informationen bereit, um einen optimalen Abbruch zu gewährleisten. Hier ist das nicht der Fall, da die Abstandsmaße durch Kraftfelder repräsentiert werden. Denn entweder wachsen die Beträge der Kraftfelder bis ins Unendliche oder bleiben konstant. Somit kann aus den Metriken kein direktes Distanz- oder Gütemaß für die Registrierungsqualität berechnet werden.

Zweitens steigt die Komplexität des Algorithmus mit zunehmender Punktwolkenauflösung rapide an. Das heißt, je mehr Punkte in den beiden Wolken vorhanden sind, umso größer werden auch die zu berechnenden Kraftfelder. Das führt zu einem extremen Anstieg der einzelnen zu berechnenden Kräfte und schließlich zu hohen Berechnungszeiten.

Letztlich fehlt noch eine geeignete Repräsentationsform der Punktwolkenbewegung. Mit Hilfe der Starrkörpertransformation können Kräfte, die auf das Template einwirken, in eine Bewegung umgeformt werden. Das Ergebnis sind ein Translationsvektor, um den die Punktwolke linear verschoben wird, und eine Drehachse inklusive passendem Winkel, um den die Punktwolke rotiert. Diese Bewegungsrepräsentation würde sogar genügen. Jedoch lassen sich Transformationen mit Vektor- und Achse-Winkel-Darstellung nicht direkt verknüpfen. Das heißt, es kann keine rigide Gesamtbewegung der Punktwolke über alle Iterationen auf direktem Weg berechnet werden. Daher ist es notwendig, eine andere, nach Möglichkeit auf den Algorithmus angepasste Transformationsbeschreibung zu finden, die zudem kompakt und effizient ist.

In den folgenden drei Teilabschnitten werden Lösungen für diese Probleme vorgestellt. Zunächst wird ein Verfahren beschrieben, das ebenfalls durch physikalische Modelle inspiriert ist. Mit dessen Hilfe kann die Punktwolkenbewegung reguliert werden. Das bedeutet, einerseits werden die Oszillationen um das Optimum vermieden, andererseits stellt dieses Verfahren auch gleichzeitig ein Abbruchkriterium zur Verfügung, bei dem die Registrierung als beendet gilt und der Algorithmus terminiert. Danach folgt Teilabschnitt 3.3.2. Dieser beschäftigt sich mit der Reduktion der Komplexität. Im letzten Teil-

abschnitt 3.3.3 wird schließlich eine Transformationsdarstellung vorgestellt, die sich zwar vom Standard in der Registrierung unterscheidet, aber Einzelbewegungen in kompakter Form miteinander verknüpft und einen weiteren entscheidenden Vorteil bietet: Bewegungen können geglättet werden.

### 3.3.1 Regularisierung

Simulated Annealing ist ein stochastisches Verfahren, das ursprünglich eingeführt wurde, um kombinatorische Optimierungsprobleme zu lösen [12, 49]. Wie der Name bereits sagt, werden hier physikalische Abkühlungsprozesse simuliert, um so iterative Optimierungsprozesse zu lösen.

Die Funktionsweise des Algorithmus ist demnach an die Lehre der statistischen Thermodynamik angelehnt. U. a. besagt diese, dass Systeme, die erhitzt werden, einen hohen energetischen Zustand besitzen und dadurch sehr instabil sind. Wird zum Beispiel ein starrer Körper aus Metall sehr stark erwärmt, beginnt die feste Gitterstruktur der Atome sich zu lösen. Die Bewegung der Atome wird zunehmend zufälliger und die Struktur des Metalls instabiler, je mehr Energie hinzugefügt wird. Nach dem Erhitzen beginnt sich das Material langsam abzukühlen und das System wird allmählich stabil. Die Atome des Metalls ordnen sich wieder in einer Gitterstruktur an. Das geschieht so lange, bis der ursprüngliche und optimale Zustand unter Berücksichtigung der äußeren Einflüsse wieder hergestellt ist. In einem idealen System wäre das bei einer Temperatur von  $0\text{ K}$  der Fall. Das ist theoretisch der absolute Nullpunkt, bei dem keine Bewegungen mehr stattfinden.

Das Simulated Annealing kann somit als Gradientenabstiegsverfahren interpretiert werden, das zusätzlich gewisse Unsicherheiten zwischen den Zustandsübergängen simuliert. Das heißt, jeder Zustand des Systems besitzt eine messbare Energie. Das ist vergleichbar mit der Kostenfunktion eines Optimierungsprozesses. Eine Übergangswahrscheinlichkeit gibt an, ob der aktuelle Zustand verlassen werden kann, um in einen neuen Zustand mit geringerer Energie zu wechseln. Diese Übergangswahrscheinlichkeit wird durch die Boltzmann-Gibbs-Verteilung beschrieben und hängt sowohl von der aktuellen Temperatur als auch von der derzeitigen Energie des Systems ab. Damit ist der Unterschied zu klassischen Gradientenabstiegsverfahren, dass beim Simulated Annealing ebenso unerwartete Zustandsübergänge möglich sind. Das führt wiederum dazu, dass lokale Optima mit einer gewissen Wahrscheinlichkeit auch wieder verlassen werden können.

Da das Simulated Annealing die Energie eines Systems als Kostenfunktion verwendet und die Temperatur als Abbruchkriterium gilt, kann dies ohne großen Aufwand auf das

neue Konzept der Punktwolkenregistrierung adaptiert werden:

Der Registrierungsprozess startet mit einer beliebigen Anfangstemperatur  $T_0 > 0$  und kühlt sich in jedem Iterationsschritt um einen bestimmten Faktor ab. Dieser ist konstant und kann daher als Abkühlungskonstante  $c \in ]0, 1[$  bezeichnet werden. Dann ist

$$T_k = T_{k-1} \cdot c \quad (3.29)$$

die Temperatur des  $k$ -ten Registrierungsschrittes.

Die Kostenfunktion des  $k$ -ten Zustands kann über dessen aktuelle kinetische Energie beschrieben werden. In der klassischen Mechanik wird diese über

$$E = \frac{1}{2} M v^2 + \frac{1}{2} J \omega^2 \quad (3.30)$$

berechnet, wobei  $M$  die Gesamtmasse des Systems ist. Im Folgenden wird angenommen, dass  $M = 1$  ist. Da die Punktwolkenbewegung bereits als schrittweise definiert wird und stets in absoluter Ruhe beginnt, kann die Geschwindigkeit  $v$  mit dem Betrag der linearen Bewegung  $x$  aus Gleichung (3.11) gleichgesetzt werden. Ebenfalls gegeben ist die Winkelgeschwindigkeit  $\omega$ , die dem Winkel  $\theta$  aus Gleichung (3.21) entspricht. Das Drehmoment  $J$  ist aus der Gleichung (3.19) bekannt. Zudem gilt auch hier, dass aus Gründen der algorithmischen Vereinfachung auf den Erhalt der physikalischen Einheiten verzichtet wird.

Die kinetischen Energien, die während der Punktwolkenregistrierung auftreten, sind in den meisten Fällen sehr groß. Das liegt zum einem daran, dass Kräfte aus der Superposition von Kraftfeldern entstehen und daher stark von den hohen Auflösungen der Punktwolken abhängen. Zum anderen ist in vielen Anwendungen der anfängliche Abstand beider Wolken nicht groß. Demnach herrschen bereits von Beginn der Registrierung an hohe Kräfte. Aus diesem Grund werden im Folgenden kinetische Energien, anders als beim physikalischen Mechanikmodell, um den Logarithmus zur Basis 10 skaliert. Daraus ergibt sich die kinetische Energie  $E_k$  im  $k$ -ten Registrierungsschritt wie folgt:

$$E_k = \log_{10} \left( \frac{1}{2} \|\mathbf{x}_k^2\| + \frac{1}{2} J_k \omega_k^2 \right). \quad (3.31)$$

In dem physikalisch motivierten Registrierungsverfahren sind die Kräfte direkt proportional zur Qualität der Überlagerung oder zumindest gleichbleibend groß. Daraus folgt, dass aus algorithmischer Sicht ein Maximierungsproblem gelöst werden muss. Die kinetische Energie steigt kontinuierlich an. Aus steigenden Energien resultiert aber auch stets eine größere Punktwolkenbewegung. Das führt zu oszillierenden Bewegungen um das Optimum, sodass dieses nie erreicht werden kann. Aus diesem Grund wird an dieser

Stelle auf ein weiteres physikalisches Phänomen Bezug genommen: Der theoretische absolute Nullpunkt der Temperatur, Null Kelvin. In diesem Zustand kommt es in den meisten Materien zum Stillstand sämtlicher Bewegung. In Analogie zu dieser Theorie wird im Folgenden die Bewegung zum Zeitpunkt  $k$  ebenfalls in Abhängigkeit zu der Temperatur  $T_k$  gesetzt. Das geschieht durch die direkte Skalierung der Linear- und Rotationsbewegung um den Faktor Temperatur  $T_k$ .

Die zunehmenden Kräfte während des Registrierungsprozesses verursachen ein weiteres Problem: Angenommen es kommt zu irgendeinem Zeitpunkt  $k$  zu einer perfekten Überlagerung beider Punktwolken, dann würde der Betrag der Anziehungskräfte ins Unendliche steigen. Das bedeutet, der Algorithmus muss sehr große Gradientenanstiege zwischen zwei aufeinander folgenden Iterationen unterdrücken, vor allem gegen Ende der Registrierung. In Folge der zunehmenden Genauigkeiten steigen die Kräfte extrem stark an und verursachen immer größere lineare und rotatorische Bewegungen. Jedoch unmittelbar zu diesem Zeitpunkt der Registrierung ist es von Vorteil, wenn die Bewegungen möglichst klein und fein werden, um sich kontinuierlich dem Optimum anzunähern.

Zum Teil reguliert das bereits die konstante Abkühlung und die damit verbundene Skalierung der Linear- und Rotationsbewegung. Jedoch können zu große Energiesprünge nicht immer abgefangen werden. Hinzu kommt, dass es in wenigen Fällen auch von Vorteil sein kann, dass genau solche großen Punktwolkenbewegungen ausgeführt werden. Das ist vor allem dann der Fall, wenn der Registrierungsprozess in einem lokalen Optimum festhängt. Dann ist eine große Bewegung hilfreich, obgleich diese vom globalen Optimum weg zeigt und sich beide Punktwolken weiter voneinander entfernen.

Solche Phänomene können mit Hilfe der Übergangswahrscheinlichkeiten reguliert werden. In dieser Arbeit wird die Übergangswahrscheinlichkeit definiert als

$$\rho_T = e^{\left(-\frac{\Delta E}{T_k}\right)}. \quad (3.32)$$

Somit ist  $\rho_T$  die Wahrscheinlichkeit, die angibt, ob ein Energieübergang vom vorherigen Energiezustand  $E_{k-1}$  zum aktuellen Zustand  $E_k$  möglich ist. Wobei  $\Delta E = E_k - E_{k-1}$  der Energieunterschied beider Zustände ist und dieser von der aktuellen Temperatur  $T_k$  abhängt. Die Berechnung der Wahrscheinlichkeit beruht auf der Boltzmann-Gibbs-Verteilung [12]. Sie entscheidet, ob der aktuelle Zustand akzeptiert wird, also die Punktwolkenbewegung wie berechnet stattfinden kann. Andernfalls wird das Template in diesem Schritt um den Winkel der vorherigen Iteration rotiert, aber mit neuer Achse. Ähnliches gilt für die Linearbewegung. Hier wird bei Ablehnung des Zustandsübergangs zwar die normierte Richtung des aktuellen Bewegungsvektors verwendet, aber die Länge des Vektors entspricht der des vorherigen Zustands.

---

**Algorithm 1** Pseudocode zur Regularisierung der Punktwolkenbewegung mittels Simulated Annealing.

---

**Phase 1: Übergangswahrscheinlichkeiten**

$$\Delta E \leftarrow E_k - E_{k-1}$$

**if**  $\Delta E \geq 0$  **then**

$$\rho_T \leftarrow e^{\left(-\frac{\Delta E}{T_k}\right)}$$

$$p \leftarrow \text{rand}(0, 1)$$

**if**  $(\rho_T < p)$  **then**

$$\mathbf{x}_k \leftarrow \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|} \cdot \|\mathbf{x}_{k-1}\|$$

$$\theta_k \leftarrow \theta_{k-1}$$

$$E_k \leftarrow E_{k-1}$$

**Phase 2: Regulation**

$$T \leftarrow T \cdot c$$

$$\mathbf{x}_k \leftarrow \mathbf{x}_k \cdot T$$

$$\theta_k \leftarrow \theta_k \cdot T$$


---

Der Pseudocode in Algorithmus 1 fasst das adaptierte Simulated Annealing für die Registrierung von Punktwolken zusammen. In jeder Iteration  $k$  des Registrierungsverfahrens werden am Ende der Kraftfeld- und Bewegungsberechnung, jedoch vor der eigentlichen Umsetzung der Punktwolkenbewegung, zwei zusätzliche Phasen eingeschoben. In der ersten Phase werden die Übergangswahrscheinlichkeiten berechnet. Anhand derer wird entschieden, wie die Punktwolkenbewegung auszusehen hat. Die zweite Phase reguliert daraufhin den Betrag der Linear- und Rotationsbewegung.

Genauer heißt das: In der ersten Phase wird entschieden, ob der aktuelle Zustand akzeptiert wird. Das gilt ausnahmslos für alle  $\Delta E < 0$ , also immer dann, wenn die aktuelle kinetische Energie  $E_k$  kleiner ist als die  $E_{k-1}$  des vorherigen Zustands. Das bedeutet, Zustandswechsel sind immer dann erlaubt, wenn die Bewegungen im Betrag kleiner werden und somit feiner. Für den Fall, dass  $E_k \geq E_{k-1}$ , muss zusätzlich entschieden werden, ob der Energieanstieg und somit die Bewegung der Punktwolke zu groß sind. Diese Entscheidung hängt im großen Maße von der Übergangswahrscheinlichkeit  $\rho_T$  ab (vgl. Gleichung (3.32)). Ist  $\rho_T$  größer oder gleich einer gleichverteilten Zufallsvariable  $p \in [0, 1]$ , dann wird der Zustandsübergang ebenfalls akzeptiert. Andernfalls werden die aktuell berechneten Bewegungen um die Beträge der vorherigen Iteration ersetzt.

Die Wahrscheinlichkeit, dass  $\rho_T \geq p$ , ist am Anfang der Registrierung noch hoch. Das liegt daran, dass sich die Temperatur zum Anfang der Registrierung ebenfalls in einem

relativ hohen Wertebereich befindet. So können größere Energiezustandswechsel zugelassen werden, die eine Punktwolke über eventuelle lokale Optima hinwegbewegen, auch wenn dies bedeuten sollte, dass sich die Punktwolke kurzzeitig vom globalen Optimum wegbewegt. Extreme Energieanstiege zwischen den Zustandsübergängen werden jedoch mit hoher Wahrscheinlichkeit unterdrückt. So wird vermieden, dass sich beide Punktwolken zu weit voneinander entfernen.

Mit abnehmender Temperatur sinkt auch die Wahrscheinlichkeit, dass  $\rho_T \geq p$ . Was wiederum dazu führt, dass auch die Wahrscheinlichkeit, hohe Energieanstiege zuzulassen, sinkt. Folglich finden zum Ende der Registrierung, vor allem wenn  $T_k \rightarrow 0$ , nur noch feine Punktwolkenbewegungen statt. So kann der Algorithmus allmählich auf das aktuelle Optimum hin iterieren, ohne sich zu weit darüber hinwegzubewegen.

Sollte nun  $\rho_T < p$  sein, werden sowohl der Betrag der linearen Bewegung als auch der Winkel des aktuellen Zustands durch die des Vorherigen ersetzt. Das Gleiche gilt auch für die Energie. Die Richtung der Linearbewegung sowie die Rotationsachse bleiben erhalten. Das ist zulässig, da dieser Fall nur bei sehr hohen Energiedifferenzen eintritt und diese durch eine sehr große Kraft zwischen den Punktwolken verursacht wird. Solche hohen Kräfte wiederum sind ein Anzeichen dafür, dass in der berechneten Bewegungsrichtung eine hohe Registrierungsgenauigkeit zu erwarten ist. Um solch eine optimale Überlagerung beider Punktwolken nicht durch zu große Bewegungen zu weit zu überschreiten, werden die Beträge der Transformation auf die kleineren des Vorgängerzustandes reguliert.

In der zweiten Phase werden dann die akzeptierten Linear- und Rotationsbewegungen aus Phase 1 in Abhängigkeit der Zeit  $T_k$  regularisiert. Das heißt, zu Beginn der Registrierung ist die Bewegung noch grob. Mit  $T_k \rightarrow 0$  wird die Punktwolkenbewegung ebenfalls reduziert. Folglich werden, zusätzlich zu den Kontrollmechanismen der ersten Phase, auch hier starke Oszillationen gegen Ende der Registrierung vermieden.

Abbildung 3.5 zeigt das Flussdiagramm des Registrierungsverfahrens. Der Algorithmus berechnet in jeder  $k$ -ten Iteration Kraftfelder zwischen den beiden Punktwolken, die den aktuellen Abstand beider Wolken zueinander modellieren. Aus den Kraftfeldern resultiert nach dem Superpositionsprinzip eine Kraft, die im Massenschwerpunkt des Templates angreift. Analog zur Starrkörpertransformation der klassischen Mechanik führt das zu Rotations- und Linearbewegungen der rigiden Punktwolke. Die stets größer werdenden bzw. konstant bleibenden Bewegungen tendieren dazu, weit über das Optimum hinaus zu schießen oder um das Ziel zu oszillieren. Um das zu unterbinden, wurde in diesem Teilabschnitt ein neuer Bestandteil entwickelt und dem Algorithmus hinzugefügt.

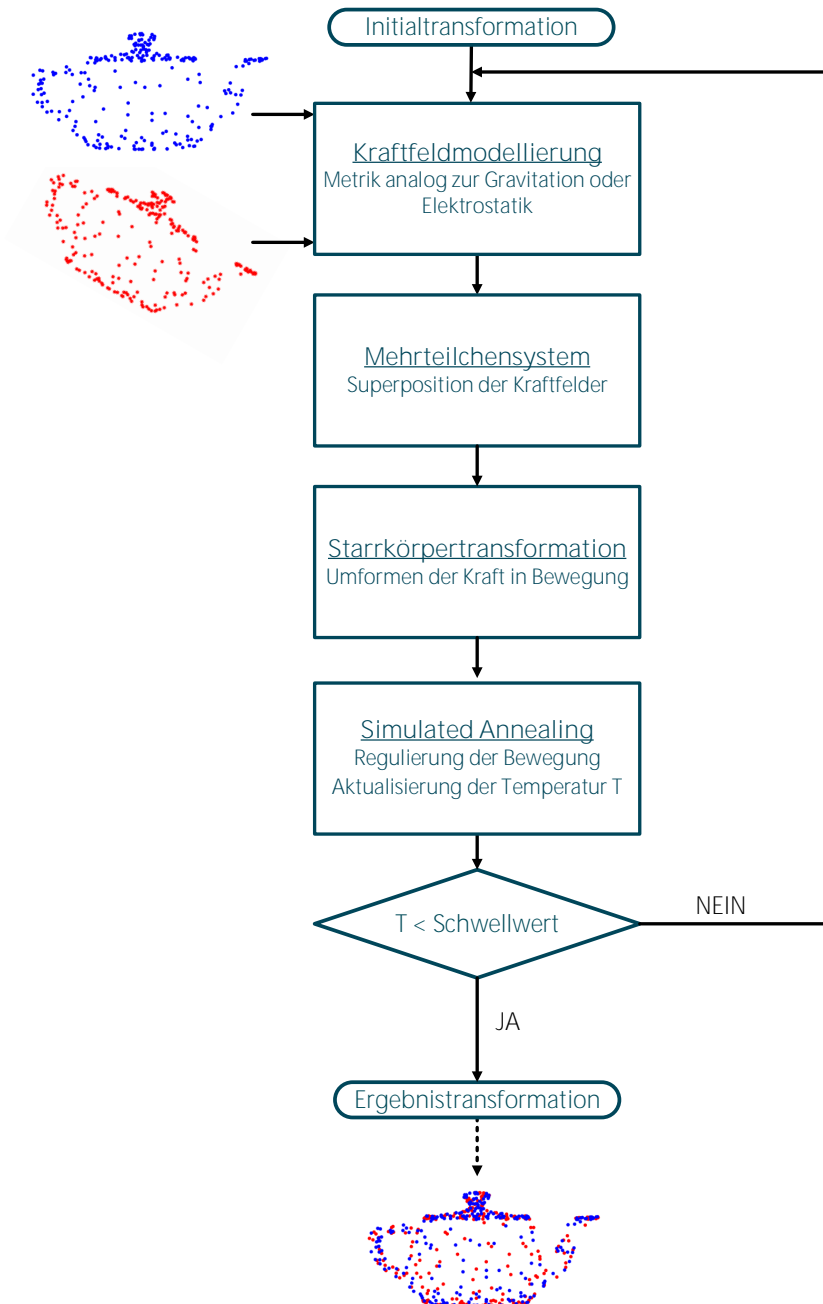


Abbildung 3.5: Das Flussdiagramm des neuen Registrierungsalgorithmus erweitert um einen Regularisierer. Mit Hilfe des Simulated Annealing werden die Punktwolkenbewegungen regularisiert und die Terminierung des iterativen Algorithmus ermöglicht.

Mit Hilfe eines angepassten Simulated Annealing werden die Bewegungen der Punktwolke regularisiert, sodass grobe Transformationen nur zu Beginn der Registrierung zulässig sind und mit einer hohen Wahrscheinlichkeit zugelassen werden. In späteren Iterationen nimmt die simulierte Temperatur des Systems kontinuierlich ab. Das führt wiederum dazu, dass auch die Punktwolkenbewegung feiner wird und große Sprünge nur noch mit sehr geringer Wahrscheinlichkeit geschehen. Ist die Temperatur nach der  $k$ -ten Iteration kleiner als ein frei wählbarer Schwellwert  $\varepsilon > 0$ , dann terminiert der Algorithmus. In diesem Fall wird die Transformation  ${}^0\mathcal{T}_K$  zurückgegeben, die das Template so bewegt, dass es das Modell möglichst optimal überlagert. Andernfalls wird, basierend auf der aktuellen Transformation  ${}^0\mathcal{T}_k$ , eine weitere Iteration durchgeführt.

Die Abkühlungskonstante  $c$  und der Temperaturschwellwert  $\varepsilon$  sind notwendige Bestandteile des Simulated Annealing und vom Anwender frei wählbar. Inwiefern sich die richtige Wahl dieser Parameter auf die Qualität und Geschwindigkeit der Registrierung auswirkt, wird in Kapitel 5 ausführlicher analysiert.

### 3.3.2 Reduktion der Komplexität

Bisher ist der Algorithmus so konzipiert, dass Kräfte modelliert werden können, die als Abstandsmaß zwischen den zu registrierenden Punktwolken dienen. Diese werden in jeder Iteration neu berechnet und sind demnach abhängig vom aktuellen Versatz beider Punktwolken. Der entscheidende Vorteil hierbei ist, dass mit Hilfe des Mehrteilchensystems diese Kräfte zusammengefasst und im Massenschwerpunkt des Templates gebündelt werden können. Daraus folgt eine schrittweise rigide Bewegung analog zur Starrkörpertransformation.

Nachteilig an dieser Herangehensweise ist, dass für die Kraftfeldberechnung alle Punkte des Modells mit denen des Templates verglichen werden. Folglich ist die Komplexität des Algorithmus direkt von den Auflösungen beider Punktwolken abhängig. Das heißt, dass bei großen Punktzahlen die Laufzeit des Algorithmus unverhältnismäßig stark ansteigt. Genauer kann die Zeitkomplexität zur Kraftfeldberechnung mit

$$\mathcal{O}(M \cdot N \cdot K) \tag{3.33}$$

abgeschätzt werden, wobei  $M$  die Anzahl der Punkte im Modell und  $N$  die Punktzahl im Template angeben.  $K$  ist die Anzahl an Iterationen, die für die gesamte Registrierung notwendig ist.

Effizienter wäre eine Lösung, bei der die Laufzeit des Algorithmus nicht von der Auflösung der Punktwolken abhängt oder zumindest nur in einem sehr geringen Maße. Idealerweise ist die Komplexität dann konstant. Hierfür werden im Folgenden zwei Anpassungen

des bisherigen Algorithmus vorgenommen. Zuerst wird das Zufallsprinzip angewandt, um die eigentliche Anzahl an Berechnungen zu senken. Danach wird die Kraftfeldmethode parallelisiert, sodass diese performant auf Mehrkernrechnerarchitekturen berechnet werden kann, wie zum Beispiel Grafikprozessoren.

**Monte-Carlo-Verfahren** Das Monte-Carlo-Verfahren beruht auf dem Zufallsprinzip und kommt immer dann zum Einsatz, wenn das zu berechnende Problem zu komplex ist, um es vollständig zu berechnen. Es beschreibt die Wirklichkeit durch eine geringe Anzahl an zufällig gewählten Stichproben. Das Prinzip wurde von Stan Ulam und John von Neumann so formuliert, dass es für die Berechnung auf Computern angewandt werden kann [68]. Seitdem sind vielzählige und spezielle Varianten der Monte-Carlo-Simulationen entwickelt worden. U. a. lösen diese komplexe Problem der Physik, Mathematik, Biologie, Chemie und Materialwissenschaft, aber auch der Ökonomie und Soziologie [56].

Die Monte-Carlo-Methode kann auch auf das Registrierungsproblem angewandt werden. Anstatt alle möglichen Punktkorrespondenzen zwischen Modell und Template zu berücksichtigen, werden jeweils nur zufällig gewählte Stichproben aus beiden Punktwolken für die Kraftfeldberechnung verwendet. Im besten Fall wird mit den wenigen Stichproben ein möglichst genaues Abbild der Wirklichkeit geschaffen, welches die Kraftfelder ausreichend approximiert. Hierfür werden, pro Iteration, gleichverteilt  $\mathcal{N}$  Punkte aus dem Template gewählt. Dann werden zwischen jeden der  $\mathcal{N}$  Template-Punkte die Kräfte zu  $\mathcal{M}$  gleichverteilten Stichproben aus dem Modell berechnet. Die Zeitkomplexität der gesamten Registrierung lässt sich schließlich mit dem konstanten Wert

$$\mathcal{O}(\mathcal{M} \cdot \mathcal{N} \cdot K) \leq \mathcal{O}(M \cdot N \cdot K) \quad (3.34)$$

abschätzen.  $K$  ist auch hier die Anzahl an benötigten Iterationen.  $\mathcal{M}$  und  $\mathcal{N}$  können frei gewählt werden und müssen nicht zwangsläufig gleichgroß sein, solange  $1 \leq \mathcal{M} \leq M$  und  $1 \leq \mathcal{N} \leq N$  gilt.

Wichtig ist, dass in jedem Iterationsschritt jeweils eine neue Stichprobe aus den Punktwolken gewählt wird. Erst unter dieser Voraussetzung ist sichergestellt, dass die Varianz der Stichproben während des kompletten Registrierungsprozesses hoch genug ist. In diesem Fall hängt das Ergebnis nicht nur von einer möglicherweise besonders schlechten Auswahl ab. Das unterscheidet die Kraftfeldmethode von anderen Verfahren, wie beispielsweise dem ICP. Denn um die Komplexität der anderen Registrierungsalgorithmen zu minimieren, werden die Punktwolkenauflösungen lediglich vor der Registrierung reduziert. Grund hierfür sind vor allem die verwendeten Datenstrukturen, die der zusätzlichen Beschleunigung des jeweiligen Algorithmus dienen.  $k$ D-Bäume benötigen zum Beispiel

für die Konstruktion erheblich mehr Zeit als für die Suche. Daher ist es in diesen Fällen unzumutbar, in jeder Iteration die Stichproben und somit die Baumstrukturen stets neu zu berechnen.

Für den Algorithmus, der im Rahmen dieser Arbeit entwickelt wird, sind solche Datenstrukturen nicht vorgesehen. Die Verwendung des Monte-Carlo-Prinzips ist somit zulässig. Außerdem ist die Laufzeit nicht mehr von den Punktwolkenauflösungen abhängig, sondern lediglich von den Stichprobengrößen  $\mathcal{M}$  und  $\mathcal{N}$ . Daraus ergibt sich eine annähernd konstante Laufzeitkomplexität für die Registrierung. Es gilt: Je größer die Stichproben sind, desto höher ist die Registrierungsgenauigkeit und folglich die Komplexität. Wenn weniger Punkte für die Berechnung des Kraftfeldes einbezogen werden, dann sinkt die Zuverlässigkeit der berechneten Abstandsmaße und die Punktwolkenanpassung weicht zunehmend stark vom Optimum ab. Inwiefern die Stichprobengröße die Genauigkeit und Geschwindigkeit der Registrierung beeinflusst wird im Evaluierungskapitel 5 ausführlicher untersucht.

**Parallelisierung** Mit Hilfe der Monte-Carlo-Methode kann die Komplexität des Algorithmus und somit die Laufzeit auf einem annähernd konstanten Wert gehalten werden, unabhängig von der Punktwolkenauflösung. Die Genauigkeit und Geschwindigkeit der Registrierung hängt hierbei von der Stichprobengröße ab. Für eine bessere Qualität der Registrierungsergebnisse sollten die Stichproben möglichst groß gewählt werden. Das wiederum wirkt sich negativ auf die Geschwindigkeit des Algorithmus aus. Das gilt aber nur solange alle Berechnungen seriell sind, also nacheinander ausgeführt werden. Da die Kraftfeldberechnungen der einzelnen Punkte im Template unabhängig voneinander sind, können diese gleichzeitig bzw. parallel ausgeführt werden.

Demnach können, unabhängig von den anderen Punkten im Template, die gesamten externen Kräfte des Punktes  $\mathbf{y}_i \in \mathcal{Y}$  zu sämtlichen  $\mathcal{M}$  Punkten des Modells berechnet werden. Gebündelt durch die Superposition geben die Kräfte deshalb den Einfluss des Modells auf den jeweiligen  $i$ -ten Template-Punkt an. Sobald die  $\mathcal{N}$  Kraftfelder für eine Template-Stichprobe berechnet und gebündelt wurden, können diese mit Hilfe des Mehrteilchensystems auf den Massenschwerpunkt des gesamten Templates angewendet werden. Zusammengefasst heißt das, würden die Kraftfelder des Templates komplett parallel ausgeführt werden, dann würde sich die Abschätzung der Zeitkomplexität aus Gleichung (3.34) auf

$$\mathcal{O}(\mathcal{M} \cdot K) \tag{3.35}$$

reduzieren lassen.

Das Algorithmen-Design lässt folglich die Verwendung von Parallelarchitekturen zu. Diese sind Systeme, die in der Lage sind, mehrere Berechnungen im gleichen Zeitschritt auszuführen, wie zum Beispiel die Grafikprozessoreinheiten (GPUs).

Eine GPU ist, ähnlich dem Hauptprozessor (CPU) eines Computers, ein einzelnes physisches Rechenwerk mit mehreren Recheneinheiten. Doch im Gegensatz zur CPU sind in einer GPU mehrere hundert Rechenkerne in einer einzelnen Hardwareeinheit verbaut. Bei modernen CPUs existieren ebenfalls Lösungen mit mehreren Rechenkernen pro Hardwareeinheit. Doch hier liegt die Maximalzahl noch immer im zweistelligen Bereich. Eine GPU kann daher eine weitaus höhere Zahl an Operationen nebenläufig ausführen als eine CPU. Beispielsweise erreicht die GPU einer NVIDIA Tesla K40 bis zu 5 Terra-FLOPs<sup>2</sup>. Ein aktueller Intel Core i7 Prozessor mit 3.46 GHz hingegen bewältigt bis zu 55.36 Giga-FLOPs [78].

Das ist einer der Gründe, warum die GPUs für die Berechnung von komplexen Problemen zunehmend an Bedeutung gewonnen haben, denn eigentlich ist deren ursprüngliche Aufgabe die effiziente Verarbeitung und Ausgabe von Computergrafiken. Aber auch der Zugang zu den GPUs wurde den Anwendern erleichtert. Dies ermöglichen u. a. High-Level Programmiersprachen, wie beispielsweise CUDA C/C++ [17, 48, 73, 78].

Zu beachten ist, dass GPUs ihr Potential nur bei der parallelen Ausführung gleicher Aufgaben optimal entfalten können. Lang andauernde sequentielle Berechnungen oder stets wechselnde Funktionen werden von der CPU effektiver gelöst. Daher sollte die Verwendung von GPUs nicht in jedem Fall der CPU vorgezogen werden. Grundsätzlich sollte ein Algorithmus die folgenden drei Bedingungen erfüllen, damit dieser auf der GPU effizient gelöst werden kann [78]:

1. Die Berechnung kann in hunderte oder tausende von einander unabhängige Einheiten aufgeteilt werden. Am effizientesten sind GPUs, wenn die Berechnung auf alle Recheneinheiten gleichmäßig aufgeteilt wird und diese möglichst zur selben Zeit ausgelastet sind.
2. Die Rechenzeit sollte signifikant größer sein als die Zeit, die benötigt wird, um Daten für die Berechnung von der CPU zur GPU zu transferieren und wieder zurück. Gleichwohl sollte die Rechenzeit klein genug sein, um das Problem in angemessener Zeit lösen zu können.
3. Letztlich sollte auch der Transfer von Daten zwischen CPU und GPU minimal gehalten werden.

---

<sup>2</sup>Floating point Operation per Second

Die Berechnung der Kraftfelder lässt sich so formulieren, dass diese drei Kriterien erfüllt sind. Zunächst können die  $\mathcal{N}$  einzelnen Kraftfeldberechnungen der Template-Punkte unabhängig voneinander berechnet werden. Auf jedes dieser Kraftfelder haben  $\mathcal{M}$  Einzelkräfte ihren Einfluss. Das heißt, grundsätzlich werden  $\mathcal{N}$  viele Berechnungen auf der GPU ausgeführt. Wenn  $\mathcal{N}$  groß genug ist, wird eine effektive Ausnutzung aller Recheneinheiten der GPU gewährleistet.  $\mathcal{M}$  hingegen kann so gewählt werden, dass die Berechnungszeiten pro Recheneinheit der GPU angemessen sind.

Auch der Datentransfer zwischen CPU und GPU kann für das Registrierungsverfahren klein gehalten werden. Den größten Anteil hieran haben die beiden zu registrierenden Punktwolken. Diese können, je nach Auflösung, extrem groß sein, vor allem dann, wenn sie mit zusätzlichen Merkmalen erweitert werden. Sie müssen aber nur ein einziges Mal vor Beginn der Registrierung von der CPU an die GPU übertragen werden. Während der Berechnung müssen keine weiteren Daten übertragen werden. Nach Abschluss reicht es die Punktwolkendaten zu löschen. Lediglich die berechnete Transformation muss am Ende der Registrierung von der GPU zur CPU übermittelt werden. Die Datengröße der Transformation ist zwar von der Art der Transformationsdarstellung abhängig, aber in diesem Zusammenhang vernachlässigbar klein.

Einzig die Wahl der zufälligen Stichproben kann dazu führen, dass der Algorithmus gegen eines der drei Kriterien verstößt. Um entsprechende Punkte aus den Wolken wählen zu können, müssen pro Iteration  $\mathcal{M}$  und  $\mathcal{N}$  viele Zufallszahlen generiert werden. Grundsätzlich können Computer keine echten Zufallszahlen generieren [52]. Daher muss auf sogenannte Pseudozufallszahlen zurückgegriffen werden. Um solche Pseudozufallszahlen auf der CPU zu berechnen, gibt es eine Vielzahl von Methoden. Indes lassen sich davon nur die wenigsten effizient auf der GPU verwenden [69]. Das liegt hauptsächlich an der hohen Anzahl von parallel ausgeführten Berechnungsschritten. Denn die meisten Zufallszahlengeneratoren verwenden den aktuellen Zeitzyklus einer Recheneinheit, um möglichst gleichverteilt Zahlen zu generieren. Auf der GPU ist es sehr wahrscheinlich, dass mehrere parallele Recheneinheiten in demselben Rechenzyklus starten. Das führt wiederum dazu, dass die Unterschiedlichkeit der Zufallszahlen, und daher eine zuverlässige Gleichverteilung der Stichproben, nicht gewährleistet werden kann.

Die einfachste Lösung des Problems ist, die Zufallszahlen auf der CPU zu erzeugen und entsprechende Listen mit  $\mathcal{M}$  und  $\mathcal{N}$  vielen Zufallszahlen an die GPU zu transferieren. Damit wäre eine zuverlässige Gleichverteilung der Pseudozufallszahlen möglich. Das Konzept sieht aber vor, dass in jeder Iteration neue Stichproben gewählt werden, um das Ergebnis nicht nur von einer einzigen, möglicherweise schlechten Stichprobe abhängig zu machen. Um dies zu vermeiden, müssten vor jeder Iteration neue Zufallszahlen an

die GPU übermittelt werden. Mit wachsenden  $\mathcal{M}$  und  $\mathcal{N}$  ist demnach auch ein hoher Datentransfer verbunden und der Algorithmus arbeitet nicht mehr ausreichend effizient.

Eine weitere Möglichkeit ist die Verwendung des sogenannten Shared Memory auf der GPU. Dieser gemeinsame Speicher kann von allen Recheneinheiten der GPU lesend und schreibend genutzt werden. Das bedeutet, die Informationen zu den aktuellen Rechenzyklen sämtlicher Recheneinheiten könnten ausgetauscht, abgeglichen und manipuliert werden. Dieser Vorgang könnte verhindern, dass gleiche Zufallszahlen in demselben Zeitabschnitt generiert werden. Jedoch ist der Schreibzugriff auf diesen Speicher nicht gleichzeitig möglich. So kann jederzeit lediglich eine Recheneinheit pro Zeiteinheit auf dem Speicher schreiben. Zudem müsste stets sichergestellt werden, dass die Lese- und Schreibzugriffe in der richtigen Reihenfolge stattfinden. Das bedeutet, dass die jeweiligen Recheneinheiten wieder aufeinander warten müssten. Die Berechnungen werden quasi seriell ausgeführt und die Recheneinheiten der GPU sind nicht zu jeder Zeit voll ausgelastet.

Daher muss eine Methode verwendet werden, die einzelne Pseudozufallszahlen direkt auf der jeweiligen Recheneinheit generiert und das ohne ausschließlich den Prozesszyklus zu verwenden. Die Methode sollte wenig Rechenleistung benötigen und den Speicherbedarf bzw. den Datentransfer niedrig halten. Einige solcher Methoden werden u. a. von Howes und Thomas vorgestellt und verglichen [36].

Für den Registrierungsalgorithmus wird ein Ansatz verwendet, der ursprünglich für die Echtzeit-Arbeitsraum-Analyse von Linearkinematiken entwickelt wurde [39]. Das Verfahren beruht auf dem linearen Kongruenzgenerator (LCG) von Knuth [52]. Der LCG hat die Form

$$p_k = (a \cdot s_k + c) \bmod m, \quad k \geq 0, \quad (3.36)$$

mit

$$\begin{aligned} m, & \quad m > 0 \\ a, & \quad 0 \leq a < m \\ c, & \quad 0 \leq c < m \\ s_k, & \quad 0 \leq s_k < m. \end{aligned} \quad (3.37)$$

Dabei ist  $m$  der Modulo, der die maximale Größe der Periodenlänge des Generators definiert. Dieser gibt an, in welchem Wertebereich die erzeugten Zufallszahlen liegen. Je größer  $m$  ist, desto gleichmäßiger sind die Zufallszahlen verteilt. Wird  $m$  aber zu groß gewählt, dann ist es wahrscheinlicher, dass bestimmte Zahlen nie gewählt werden.  $a$  und  $c$  heißen Faktor bzw. Inkrement. Diese können nach bestimmten Regeln frei gewählt

werden (vgl. [52], Seite 9-24). Dann ist

$$p_k = \begin{cases} p_k/m & , p_k \neq 0 \\ (m-1)/m & , \text{sonst} \end{cases} \quad (3.38)$$

eine Pseudozufallszahl mit  $p_k \in [0, 1]$ .

In der ursprünglichen Variante des LCG entspricht der Saatpunkt  $s_k$  dem Wert des vorherigen Zustands  $p_{k-1}$ . Das würde aber bedeuten, wenn alle Recheneinheiten der GPU mit dem gleichen Startwert  $p_0$  beginnen, diese auch permanent dieselben Zufallszahlen in den folgenden Zyklen berechnen. Das bedeutet, der LCG muss für die Verwendung auf der GPU abgewandelt werden.

Auf der GPU hat jede gestartete Recheneinheit eine eindeutige Identifikationsnummer mit bis zu drei Dimensionen. Angenommen die Identifikationsnummer, ist eindimensional, dann lassen sich deren Indizes als Vektor  $\mathbf{d}_1 = (0, 1, \dots, \mathcal{N} - 1)^T$  zusammenfassen. Dieser Vektor hat demnach die Größe der Stichprobe des Templates. Ein zweiter Vektor  $\mathbf{d}_2 = (0, 1, \dots, \mathcal{M} - 1)^T$  beinhaltet eindeutige Indizes für das Modell, unabhängig von der jeweiligen Recheneinheit. Ferner sind  $v_1 \in [0, 1]$  und  $v_2 \in [0, 1]$  zwei Pseudozufallszahlen. Diese sind für alle Recheneinheiten gleich und werden ein einziges Mal vor der Registrierung auf der CPU erzeugt. Daraufhin können vor der ersten Iteration des Registrierungsalgorithmus zwei Saatpunktvektoren

$$\mathbf{s}_{1_0} = (\mathbf{d}_1 \cdot v_1) \cdot (m - 1) \quad (3.39)$$

und

$$\mathbf{s}_{2_0} = (\mathbf{d}_2 \cdot v_2) \cdot (m - 1) \quad (3.40)$$

erzeugt werden. Hier ist  $m$  wieder der Modulo. Eingesetzt in die vektoriellen Varianten

$$\mathbf{p}_{1_0} = (a \cdot \mathbf{s}_{1_0} + c) \bmod m \quad (3.41)$$

bzw.

$$\mathbf{p}_{2_0} = (a \cdot \mathbf{s}_{2_0} + c) \bmod m \quad (3.42)$$

der Gleichung (3.36) und der jeweilig elementweisen Anwendung von Gleichung (3.38) ergeben sich  $\mathcal{N}$  Pseudozufallszahlen in  $\mathbf{p}_{1_0}$  und  $\mathcal{M}$  in  $\mathbf{p}_{2_0}$ .

Für die nachfolgenden  $k$ -ten Iterationen ist  $\mathbf{s}_{1_k} = \mathbf{p}_{1_{k-1}}$  und  $\mathbf{s}_{2_k} = \mathbf{p}_{2_{k-1}}$ . Schließlich ergeben sich pro  $k$ -ter Iteration die Stichprobenindizes der Template-Punkte  $y_i$  aus sämtlichen Elementen des Vektors

$$\mathbf{i} = \lfloor \mathbf{p}_{1_k} \cdot N \rfloor. \quad (3.43)$$

Die jeweiligen Indizes der Punkte  $x_j$  des Modells ergeben sich aus den Elementen des Vektors

$$\mathbf{j} = \lfloor \mathbf{p}_{2_k} \cdot M \rfloor. \quad (3.44)$$

Dieses Vorgehen erzeugt annähernd gleichverteilte Pseudozufallszahlen und verletzt nicht die drei Kriterien, die an parallele Algorithmen gestellt sind, welche auf der GPU ausgeführt werden. Sowohl  $v_1$  als auch  $v_2$  werden auf der CPU generiert und müssen nur ein einziges Mal vor dem Start der Registrierung auf die GPU kopiert werden. Diese Zufallszahlen stellen sicher, dass in zwei aufeinander folgenden Registrierungen mit einer hoher Wahrscheinlichkeit unterschiedliche Stichproben gewählt werden. Im Vergleich zum Transfervolumen der Punktwolken ist dieser Übertragungsaufwand vernachlässigbar. Die Komplexität der Berechnung von  $\mathbf{p}_{1_k}$  und  $\mathbf{p}_{2_k}$  ist ebenfalls sehr gering und sorgt trotzdem für eine annähernde Gleichverteilung beider Stichproben.

Das Flussdiagramm in Abbildung 3.6 zeigt, dass in diesem Teilabschnitt kein neues Modul hinzugekommen ist, aber die Kraftfeldmodellierung in  $\mathcal{N}$  Einzelberechnungen aufgeteilt wurde. In jedem dieser Teilmodule wird ein Kraftfeld berechnet, das den  $i$ -ten Template-Punkt beeinflusst. Die Kraftfelder setzen sich aus den Lagebeziehungen zu  $\mathcal{M}$  Modell-Punkten zusammen. Dabei sind die  $\mathcal{N}$  einzelnen Kraftfeldberechnungen komplett unabhängig voneinander. Das ermöglicht den Einsatz von Mehrkernrechnerarchitekturen wie zum Beispiel der GPU. Werden in jeder Iteration alle Kraftfelder parallel berechnet, wird die Laufzeit des Algorithmus im Gegensatz zur seriellen Ausführung bis zu  $\mathcal{N}$ -fach reduziert.

Zusätzlich zur Laufzeitminimierung durch die Parallelisierung wird das Monte-Carlo-Verfahren angewandt. So werden nicht mehr alle Punkte beider Wolken zur Kraftfeldberechnung hinzugezogen sondern nur Stichproben. Das heißt, pro Iteration werden zufällig  $\mathcal{M}$  und  $\mathcal{N}$  viele Punkte aus dem Modell bzw. aus dem Template für die Korrespondenzberechnung verwendet. Eigens hierfür wurde ein Zufallszahlengenerator aus dem LCG abgeleitet, der speziell an dieses Problem angepasst und für die effiziente Zufallszahlengenerierung auf der GPU konzipiert ist.

Die  $\mathcal{N}$  Kraftfelder, die die einzelnen Recheneinheiten der GPU berechnen, werden anschließend mit Hilfe des Mehrteilchensystems zusammengefasst und im Massenschwerpunkt der Template-Punktwolke gebündelt. Daraus ergibt sich eine rigide Bewegung, die analog zu kinematischen Starrkörpertransformationen beschrieben wird. Anschließend wird diese Transformation durch ein angepasstes Simulated Annealing reguliert. Gleichzeitig kühlt sich das System mit jedem Iterationsschritt zunehmend ab. Unterschreitet die Systemtemperatur einen Schwellwert, wird die Registrierung beendet und

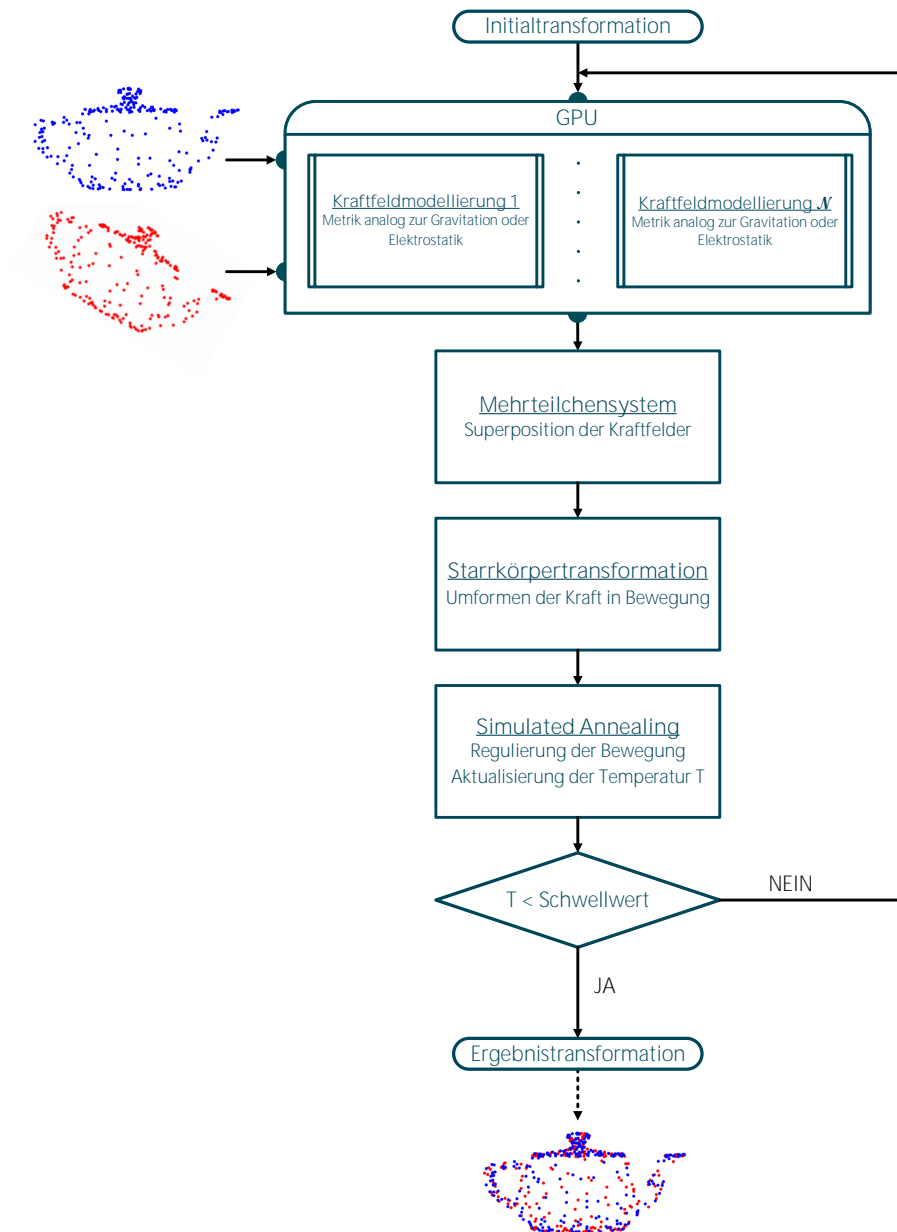


Abbildung 3.6: Das Flussdiagramm des neuen Registrierungsalgorithmus erweitert um das Monte-Carlo-Verfahren. Mit dessen Hilfe werden Stichproben beider Punktwolken für die Kraftfeldberechnung gewählt. Das senkt die Laufzeitkomplexität auf einen konstant niedrigen Wert und ermöglicht zudem die Berechnung von  $N$  Kraftfeldern parallel auf Mehrkernrechnerarchitekturen wie z.B. GPUs.

das Ergebnis in Form einer Gesamttransformation zurückgegeben. Andernfalls wird die aktuell errechnete Transformation als Initialwert für die nächste Iteration verwendet.

### 3.3.3 Transformationsbeschreibung

Nach der Gleichung (3.22) ist die Repräsentation der Punktwolkenbewegung frei wählbar. In jeder Iteration wird mit Hilfe der Starrkörpertransformation eine neue Bewegung der Punktwolke berechnet. Obwohl diese aus absoluter Ruhe, also ohne Anfangsgeschwindigkeit startet, ist diese Transformation von den hervorgegangenen Bewegungen abhängig, weil Kraftfelder, die eine solche Bewegung verursachen, die aktuelle Lagebeziehung beider Punktwolken zueinander voraussetzen. Demnach müssen die Einzeltransformationen am Ende einer Iteration miteinander verknüpft werden.

Bisher wurde für das Registrierungsverfahren eine entkoppelte Transformationsdarstellung beschrieben: die Rotation über eine Achse-Winkel-Darstellung (vgl. Gleichung (3.17) bzw. (3.21)) und die Translation über eine einfache lineare Vektorverschiebung (vgl. Gleichung (3.11)). Dies hat den Vorteil, dass diese Form der Bewegung direkt aus der Kraftfeldmethode folgt und keine weiteren Umrechnungen notwendig sind. Jeder Punkt des Templates kann auf sehr einfache Weise verschoben werden. Somit ist die Bewegungsrepräsentation direkt und intuitiv. Doch die entscheidenden Nachteile dieser Beschreibungsform sind die, vor allem im Vergleich zu anderen Rotationsdarstellungen, hohe Komplexität und numerische Instabilität beim Verknüpfen von Rotationen mit Hilfe der Achse-Winkel-Darstellung [24, 45].

Es gibt mehrere alternative Transformationsbeschreibungen [18, 24, 45]. Bezogen auf Registrierungsalgorithmen ist die Transformationsdarstellung durch homogene  $4 \times 4$ -Matrizen ein geläufiges Mittel. Das liegt vor allem daran, dass sich diese direkt aus den Optimierungsschritten der meisten Algorithmen ableiten lässt und mithin keine Konvertierung von bzw. zu anderen Darstellungen notwendig ist. Besonders gut lässt sich das am Beispiel des ICP erkennen (siehe Kapitel 2 Gleichung (2.13) bis (2.16)). Aber auch in anderen Fachgebieten wie der Robotik oder Computergrafik sind Transformationsmatrizen ein oft eingesetztes Werkzeug, um Bewegungen und Lagebeziehungen zu beschreiben. Daher werden sie oft gelehrt und sind in diverse Programmiersprachen bzw. Software integriert [45].

Die Achse-Winkel-Darstellung und der Translationsvektor ergeben sich aus den Kraftfeldern und lassen sich ebenfalls direkt in eine Transformationsmatrix übertragen (vgl. [24, 45]). Die Verknüpfung der Einzeltransformationen sowie die Bewegung der Punktwolke können direkt über Matrizenmultiplikation bzw. Matrix-Vektor-Multiplikation gelöst

werden. Diese Operationen sind nur wenig rechen- und zeitintensiv. Aus diesen Gründen und wegen der hohen Verfügbarkeit sind auch in dem hier vorliegenden Algorithmus homogene  $4 \times 4$ -Matrizen als eine mögliche Transformationsdarstellung implementiert. Schließlich ist mit dieser Transformationsbeschreibung der Algorithmus zur physikalisch motivierten Punktwolkenregistrierung komplett und es kann in einer ersten Machbarkeitsstudie gezeigt werden, ob das Verfahren valide ist und zum Stand der Technik vergleichbare Ergebnisse liefert. Diese Analyse ist im ersten Teil des Evaluierungskapitels 5 aufbereitet.

Unabhängig davon, ob die Machbarkeitsstudie zeigt, dass das neue Registrierungskonzept vergleichbar gute Ergebnisse zum Goldstandard liefert, sollte eine alternative Transformationsbeschreibung implementiert und untersucht werden. Die Transformationsmatrizen bringen nämlich einige Nachteile mit sich. Ein entscheidender negativer Aspekt ist deren numerische Instabilität. Es ist sehr wahrscheinlich, dass bei der Konvertierung von der Achse-Winkel-Darstellung in eine entsprechende Matrix geringfügige Rundungsfehler auftreten. In diesen Fällen ist die Orthogonalität des Rotationsanteils nicht länger gewährleistet und die Bewegung nicht mehr rigide. Daraus folgen unerwünschte Punktwolkendehformationen. Im Regelfall sind diese Fehler so klein, dass sie vernachlässigbar sind. Doch bei der Verknüpfung mehrerer Transformationen werden diese Fehler akkumuliert. Sind die ursprünglichen Abstände zwischen zwei zu registrierenden Punktwolken eher klein, fällt auch das nur geringfügig ins Gewicht. Sobald jedoch die Abstände größer werden, kann das schnell zu größeren Registrierungsfehlern führen. Obgleich es Methoden wie das Gram-Schmidt-Verfahren gibt, die die nötige Orthogonalität wieder herstellen können, steigern diese die Komplexität des Algorithmus [45]. Daher wird an dieser Stelle oftmals zwischen Genauigkeit und Geschwindigkeit abgewogen oder zumindest eine grobe Initialregistrierung verwendet.

Im Hinblick auf den hier zu entwickelnden Algorithmus haben die Transformationsmatrizen noch einen weiteren gravierenden Nachteil: Bewegungen lassen sich nur auf komplexe Weise interpolieren bzw. glätten. Diese Aspekte sind aber wichtig, da das neue Registrierungsverfahren auf dem Monte-Carlo-Prinzip beruht. Demnach sind, je nach Stichprobengröße, unterschiedlich große Varianzen in den Registrierungsergebnissen zu erwarten. Um diese Varianzen möglichst gering zu halten, könnten pro Iteration mehrere Punktwolkenbewegungen gleichzeitig berechnet werden. Anschließend können die Teiltransformationen gemittelt und zu einer Gesamtbewegung zusammengefasst werden. Solch eine geglättete Bewegung kann dazu führen, dass zu große Ausreißer eliminiert werden und die Genauigkeit sowie die Reproduzierbarkeit des Verfahrens steigt. Mit Hilfe von Transformationsmatrizen ist das nur schwer bzw. über Umwege möglich und ist

oftmals mit sehr viel höheren Laufzeiten verbunden [43].

**Duale Quaternionen** Wie auch die Transformationsmatrizen beschreiben Duale Quaternionen eine Bewegung in kompakter Form, also Rotation und Translation in einem Konstrukt. Anders als bei den homogenen  $4 \times 4$ -Matrizen, die mindestens zwölf Parameter zur Bewegungsbeschreibung benötigen, verwenden Duale Quaternionen lediglich acht Elemente. So lassen sich die Bewegungen mit weniger Speicher- und Rechenaufwand interpretieren. Außerdem sind sie, im Vergleich zu den Matrizen, numerisch stabiler und weisen keine Singularitäten auf. Besonders interessant an Dualen Quaternionen ist die Tatsache, dass es mehrere Verfahren zur Interpolation bzw. Glättung gibt [43].

Im Gegensatz zu den Transformationsmatrizen finden Duale Quaternionen in der Punktwolkenregistrierung nur wenig Anwendung und sind dementsprechend auch weniger bekannt. Da sie allerdings für das vorliegende Registrierungsverfahren notwendig sind, werden sie und deren Interpolation im Folgenden detailliert beschrieben.

Duale Quaternionen basieren auf der Theorie der Dualen Zahlen und erweitern die Rotationsdarstellung der Quaternionen um einen Translationsanteil. Eine Quaternion  $q$  setzt sich aus zwei Bestandteilen zusammen, einem realen Anteil  $q_0$  und einem dreikomponentigen Imaginärteil  $\mathbf{q}$ :

$$\begin{aligned} q &= q_0 + \mathbf{q} \\ &= q_0 + q_1 i + q_2 j + q_3 k. \end{aligned} \quad (3.45)$$

Die Elemente  $q_0, q_1, q_2$  und  $q_3$  sind reelle Zahlen. Die Elemente  $i, j$  und  $k$  sind imaginäre Zahlen mit den Eigenschaften

$$\begin{aligned} i^2 &= j^2 = k^2 = ijk = -1, \\ ij &= -ji = k, \\ jk &= -kj = i, \\ ki &= -ik = j. \end{aligned} \quad (3.46)$$

Quaternionen beschreiben eine Rotation über

$$q_0 = \cos\left(\frac{\theta}{2}\right) \quad (3.47)$$

und

$$\mathbf{q} = \mathbf{k} \cdot \sin\left(\frac{\theta}{2}\right). \quad (3.48)$$

Wie auch bei der Achse-Winkel-Darstellung, ist  $\mathbf{k}$  die Drehachse und  $\theta$  der Winkel einer Rotation. Außerdem ist für die rigide Rotationsbeschreibung gefordert, dass  $q$  stets die

Einheitsquaternion ist, also  $\|q\| = 1$ . Bildlich gesprochen repräsentiert die Quaternion  $q$  aufgrund dessen die Rotation als Punkt auf der Oberfläche einer vierdimensionalen Hypersphäre.

Eine Einheitsquaternion ist die normierte Quaternion

$$q = \frac{1}{\|q\|}, \quad (3.49)$$

wobei die Länge der Quaternion mit  $\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$  berechnet wird. Durch die Normierung ergibt sich die Inverse der Quaternion

$$q^{-1} = q^* \quad (3.50)$$

und entspricht ihrer Konjugierten. Die konjugierte Quaternion ist wiederum definiert als

$$q^* = (q_0, -\mathbf{q}). \quad (3.51)$$

Die sequenzielle Verknüpfung mehrerer Rotationen erfolgt durch die Multiplikation der jeweiligen Quaternionen. So ist die Verknüpfung zweier Quaternionen  $p$  und  $q$  definiert als

$$\begin{aligned} pq &= (p_0q_0 - p_1q_1 - p_2q_2 - p_3q_3) \\ &+ (p_0q_1 + p_1q_0 + p_2q_3 - p_3q_2) i \\ &+ (p_0q_2 - p_1q_3 + p_2q_0 + p_3q_1) j \\ &+ (p_0q_3 + p_1q_2 - p_2q_1 + p_3q_0) k. \end{aligned} \quad (3.52)$$

Ferner berechnet sich das Skalarprodukt zweier Einheits-Quaternionen  $p$  und  $q$  aus

$$p \circ q = p_0q_0 + p_1q_1 + p_2q_2 + p_3q_3. \quad (3.53)$$

Die Rotation eines Vektors kann dann direkt über die Einheitsquaternion  $q$  und ihrer Inversen bzw. Konjugierten berechnet werden. Dazu wird der Vektor  $y_i$  in eine korrespondierende Quaternion  $y_i = (0, \mathbf{y}_i^T)$  überführt. Der rotierte Vektor ist ebenfalls eine Quaternion und ergibt sich aus

$$y'_i = qy_iq^*. \quad (3.54)$$

Alternativ könnte die Quaternion ebenfalls in eine Rotationsmatrix konvertiert werden. Der Punkt würde demnach über das Matrix-Vektor-Produkt rotiert werden. Insgesamt ist diese zweite Variante etwas weniger komplex und kann insofern mit geringer Laufzeit gelöst werden [24]. Hierbei ist jedoch zu beachten, dass die Konvertierung wiederum zu Rundungsfehlern führen kann und somit die Orthogonalität der Matrix geprüft und

gegebenenfalls wieder hergestellt werden muss. Das erhöht erneut die Komplexität und der Laufzeitvorteil wird nichtig.

Die Verknüpfung von Quaternionen ist deutlich weniger rechenintensiv als die Orthogonalitätssicherung der Rotationsmatrizen. Ergo ist die Rotationsberechnung und das Verknüpfen mehrerer Rotationen mittels Quaternion insgesamt weniger komplex. Hinzu kommt, dass mit der Rotationsbeschreibung durch Quaternionen das Problem des Gimble Lock vermieden wird [46]. Die Rotationen sind stets eindeutig. In der Registrierung finden die Quaternionen eher weniger Einsatz. Es ist schwieriger Quaternionen als Zielfunktion für numerische Optimierungsverfahren anzugeben als Matrizen. Der Grund ist die geforderte Einheitslänge der Quaternion, welche schließlich ein quadratisches Optimierungsproblem darstellt (siehe Gleichung (3.49)).

Die Dualen Quaternionen sind eine Erweiterung der Quaternionen und basieren auf der Theorie der Dualen Zahlen. Eine Duale Zahl ist definiert als

$$\hat{a} = a + \epsilon a_0. \quad (3.55)$$

Hierbei wird  $a$  als realer Teil bezeichnet und  $a_0$  als dualer Teil mit  $a, a_0 \in \mathbb{R}$ .  $\epsilon$  ist die sogenannte duale Einheit oder auch duale Basis. Sie hat die Eigenschaft  $\epsilon^2 = 0$ . Die für die Registrierung notwendigen Grundoperationen auf den dualen Zahlen sind die Addition

$$\begin{aligned} \hat{a} + \hat{b} &= (a + \epsilon a_0) + (b + \epsilon b_0) \\ &= (a + b) + \epsilon (a_0 + b_0), \end{aligned} \quad (3.56)$$

die Multiplikation

$$\begin{aligned} \hat{a}\hat{b} &= (a + \epsilon a_0)(b + \epsilon b_0) \\ &= ab + a\epsilon b_0 + b\epsilon a_0 + a_0 b_0 \epsilon^2 \\ &= ab + \epsilon (ab_0 + ba_0) \end{aligned} \quad (3.57)$$

und die Division

$$\begin{aligned} \frac{\hat{a}}{\hat{b}} &= \frac{(a + \epsilon a_0)}{(b + \epsilon b_0)} = \frac{(a + \epsilon a_0)(b - \epsilon b_0)}{(b + \epsilon b_0)(b - \epsilon b_0)} \\ &= \frac{ab + \epsilon (ba_0 - ab_0)}{b^2} \\ &= \frac{a}{b} + \epsilon \frac{(ba_0 - ab_0)}{b^2}. \end{aligned} \quad (3.58)$$

Differenziertere Ausführungen zu Dualen Zahlen und deren Operationen finden sich u. a. in [26, 45, 76].

Duale Quaternionen sind eine Erweiterung der Dualen Zahlen um Quaternionen und wurden von Clifford [16] unter dem Namen Biquaternionen eingeführt. Sie sind definiert als

$$\hat{q} = q_r + \epsilon q_d. \quad (3.59)$$

$q_r$  ist der Realteil und  $q_d$  der Dualteil von  $\hat{q}$ .  $\epsilon$  besitzt auch hier die Eigenschaft  $\epsilon^2 = 0$ . Insofern gelten die gleichen Grundrechenoperationen wie bei den Dualen Zahlen.

Neben diesen Basisoperationen gibt es noch weitere Berechnungsvorschriften auf den Dualen Quaternionen, die für die Glättung von Bewegungen nützlich sind. So ist die Multiplikation einer Dualen Quaternion  $\hat{q}$  mit einem Skalar  $s \in \mathbb{R}$  definiert als

$$s\hat{q} = sq_r + s\epsilon q_d. \quad (3.60)$$

Ferner ist die Konjugierte einer Dualen Quaternion

$$\hat{q}^* = q_r^* + \epsilon q_d^*. \quad (3.61)$$

Sie besteht also den Konjugierten der beiden Quaternionen  $q_r$  und  $q_d$ . Mit Hilfe der Konjugierten wiederum kann der Betrag wie folgt berechnet werden:

$$\|\hat{q}\| = \hat{q}\hat{q}^*. \quad (3.62)$$

Ist der Betrag einer Dualen Quaternion

$$\|\hat{q}\| = 1, \quad (3.63)$$

wird diese Einheits-Duale Quaternion genannt. Ist die Einheitsbedingung erfüllt, repräsentiert  $\hat{q}$  eine rigide Transformation. In diesem Fall wird die Rotation durch die Quaternion  $q_r$  beschrieben. Die translatorische Information ist in

$$q_d = \frac{1}{2}\tau q_r \quad (3.64)$$

enthalten.  $\tau$  ist ebenfalls eine Quaternion, die den Translationsvektor  $t$  mit  $\tau = (0, t^T)$  beschreibt.

Mit diesen Grundrechenregeln auf den Dualen Quaternionen können die Punkte des Templates bewegt werden. Die Überführung der Achse-Winkel-Darstellung sowie die des Translationsvektors in eine Duale Quaternion ist direkt möglich und wenig komplex. Auch ist die Transformation jedes einzelnen Punktes einer Wolke mit wenigen Rechenoperationen verbunden. Das heißt, die Transformationsbeschreibung des neuen Registrierungsverfahrens kann auch mit Hilfe der Dualen Quaternionen implementiert werden. Ob

einzig das Ersetzen der Transformationsmatrizen durch Duale Quaternionen bereits zu einer Verbesserung des Registrierungsverfahrens beiträgt und dadurch die Fehlervarianz gesenkt werden kann, wird in Kapitel 5 geklärt.

Die Dualen Quaternionen bieten einen wesentlichen Vorteil, der für den neuen Registrierungsalgorithmus von Nutzen ist: das sogenannte Blending. Dieser Begriff beschreibt die gewichtete Mittlung von Transformationen. Das Verfahren wird vor allem in der Computergrafik eingesetzt, um beispielsweise Animationen von Textilien und Haut zu interpolieren. Derart kommt es nicht zu unerwünschten und unrealistischen harten Verformungen [11, 43, 44, 95]. Das Blending kann ebenfalls dafür eingesetzt werden, um Trajektorien zu interpolieren oder biomechanische Prozesse abzubilden [42, 46, 76, 77].

Für das Blending gibt es diverse Ansätze, wie das sogenannte Screw Linear Interpolation (ScLERP) [95], das Dual Quaternion Iterative Blending (DIB) [11] oder das Dual Quaternion Linear Blending (DLB) [44]. Eine detailliertere Auflistung und Beschreibung der Verfahren finden sich u. a. in [43, 77].

Mit Dualen Quaternionen können also Trajektorien und Transformationen gemittelt werden. Mithin ist dieses Verfahren dazu geeignet, die Punktwolkenbewegungen des Templates zu glätten. In jeder Iteration wird nicht nur eine einzelne Transformation berechnet, sondern  $n$ -viele. Zur Laufzeitoptimierung wird das Monte-Carlo-Verfahren verwendet, sodass nur Stichproben aus den Punktwolken zur Kraftfeldberechnung herangezogen werden. Demnach ist es sehr wahrscheinlich, dass sich auch die resultierenden Bewegungsrichtungen geringfügig unterscheiden. Aus diesem Grund werden die  $n$  Transformationen pro Iteration gemittelt. Daraus ergibt sich eine neue geglättete Bewegung.

Aufgrund des Monte-Carlo-Ansatzes können jedoch in einigen Fällen Ausreißer auftreten. Diese sorgen dafür, dass auch die gemittelten Transformationen zu stark beeinflusst werden und dementsprechend weit von der optimalen Bewegung abweichen. Obwohl solche Ausreißer sehr unwahrscheinlich sind, müssen diese berücksichtigt werden. Daher wird in den Glättungsprozess ein zusätzliches Filter integriert. Zunächst wird aus allen  $n$  Transformationen die gemittelte Bewegung berechnet. Anschließend wird die Transformation von den  $n$  ursprünglichen herausgesucht, die am weitesten vom Mittel abweicht. Diese wird eliminiert. Aus den verbleibenden  $(n - 1)$  ursprünglichen Transformationen wird erneut das Mittel berechnet. Die daraus resultierende Transformation ist folglich die geglättete Bewegung der Punktwolke. Diese bildet wiederum die Grundlage für die nächsten Iterationen bzw. stellt das Ergebnis der Registrierung dar.

Um die jeweiligen Transformationen zu mitteln, wird in dieser Arbeit das DLB Verfahren verwendet [44]. Das Verfahren ist im Gegensatz zu anderen nicht iterativ. Es hat eine

annähernd konstante Komplexität und ist somit gut an die Problemstellung dieser Arbeit angepasst [43, 77]. Das DLB beschreibt eine gewichtete Mittlung von  $n$  Transformationen durch

$$DLB(w_1, \dots, w_n; \hat{q}_1, \dots, \hat{q}_n) = \frac{w_1 \hat{q}_1 + \dots + w_n \hat{q}_n}{\|w_1 \hat{q}_1 + \dots + w_n \hat{q}_n\|}. \quad (3.65)$$

$\hat{q}_1, \dots, \hat{q}_n$  sind die  $n$  Einheits-Dual Quaternionen, welche die  $n$  aktuellen Transformationen beschreiben, und  $w_1, \dots, w_n$  sind die dazugehörigen Gewichtungen.

Um die Bewegungen zu mitteln, müssen alle Transformationen gleich gewichtet sein. Zusätzlich ist gefordert, dass die Gewichte konvex sind. Das heißt,

$$\sum_{i=1}^n w_i = 1. \quad (3.66)$$

Daraus ergeben sich für den ersten Schritt der Glättung die Gewichtungen

$$w_1 = \dots = w_n = \frac{1}{n}. \quad (3.67)$$

Das Ergebnis des DLB ist eine Einheits-Duale Quaternion. Den größten Einfluss auf die Bewegungsabweichungen bzw. unglatte Transformationen haben fehlerhafte Rotationen. Daher wird der Rotationsanteil des Blendings mit den Rotationsanteilen der  $n$  Individualergebnisse verglichen. Die Duale Quaternion, deren Rotation zu weit vom Durchschnitt abweicht, wird eliminiert. Da die Informationen zur Rotation im Realteil der Dualen Quaternion enthalten sind, müssen für eine Abstandsmessung nur diese verglichen werden. Der Abstand zwischen zwei Quaternionen  $p$  und  $q$  kann mit Hilfe der Abstandsfunktion

$$\text{dist}(p, q) = 1 - (p \circ q)^2 \quad (3.68)$$

gemessen werden. Für Einheits-Quaternionen gilt  $\text{dist}(p, q) \in [0, 1]$ . Für identische Quaternionen gilt  $\text{dist}(p, q) = 0$ . Mit zunehmendem Abstand nimmt auch der Wert der Abstandsfunktion zu.

Nachdem der entfernteste Ausreißer aus den  $n$  Transformationen bestimmt und eliminiert ist, kann erneut das Mittel über die verbleibenden Transformationen berechnet werden. Hierfür wird wiederum das  $DLB(w_1, \dots, w_n; \hat{q}_1, \dots, \hat{q}_{(n-1)})$  berechnet. Dazu müssen die Gewichtungen  $w_1 = \dots = w_{(n-1)} = \frac{1}{(n-1)}$  entsprechend angepasst werden. Die resultierende Duale Quaternion ist schließlich die geglättete Bewegungsrichtung für die aktuelle Iteration.

Unter Berücksichtigung der geglätteten Transformationsdarstellung durch Duale Quaternionen muss das Flussdiagramm zum Algorithmus angepasst werden. In Abbildung 3.7 ist zu erkennen, dass pro Iteration die Kraftfeldberechnung  $n$ -mal ausgeführt wird.

Aus diesen  $n$  Kraftfeldern ergeben sich dementsprechend auch  $n$  Transformationen. Diese werden durch Duale Quaternionen repräsentiert. Mit deren Hilfe wird in dem neu hinzugefügten Bestandteil des Algorithmus die Punktwolkenbewegung geglättet. Das Ergebnis ist wiederum eine Duale Quaternion, die mit den Transformationen aus den vorherigen Iterationen verknüpft werden kann. Ist das System ausreichend abgekühlt, gibt der Algorithmus die Gesamttransformation zurück und sämtliche Punkte des Templates können entsprechend dieser Transformation bewegt werden.

Mit dem Algorithmus nach Abbildung 3.7 ist nun auch das letzte Modul des neuen Konzeptes zur Punktwolkenregistrierung implementiert. Aufgrund dessen kann dieser in Kapitel 5 mit dem Goldstandard verglichen werden. Zusätzlich werden auch einige der veränderbaren Parameter des Simulated Annealing ausführlicher untersucht. Zudem wird die Glättung der Punktwolkenbewegung durch das Blending der Dual Quaternionen analysiert.

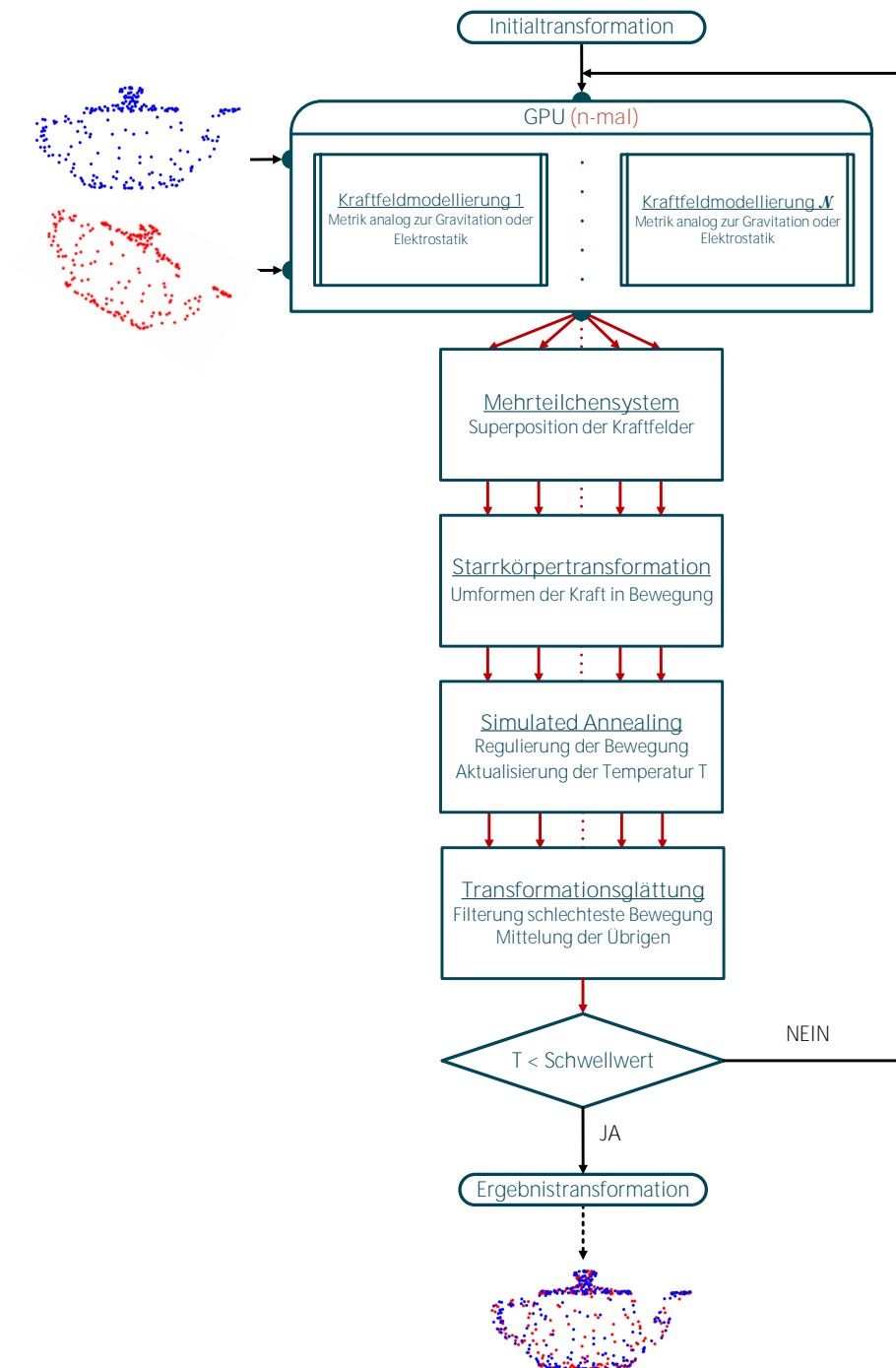


Abbildung 3.7: Das abschließende Flussdiagramm des neuen Registrierungsalgorithmus erweitert um die Bewegungsbeschreibung durch Duale Quaternionen. Mit deren Hilfe werden  $n$  Bewegungen unabhängig voneinander berechnet, gefiltert und gemittelt.

## 4 Evaluierung

Das Ziel dieser Dissertation ist es, ein neues Konzept zu entwerfen, das effizient Punktwolken miteinander registriert. Alternativ zu existierenden Ansätzen orientiert sich dieses Verfahren an verschiedenen physikalischen Gesetzmäßigkeiten. Um zu prüfen, ob das neue Paradigma eine konkurrenzfähige Alternative darstellt, muss es mit dem aktuellen Stand der Technik verglichen werden. Als Grundlage für diesen Vergleich dient der Algorithmus aus dem vorherigen Kapitel 3.

Das Verfahren basiert auf Kraftfeldern, die den Abstand zwischen zwei Punktwolken modellieren. Die Kraftfelder werden durch Metriken erzeugt, die unter Einhaltung spezieller Regeln frei gestaltet werden können. Möglich sind sowohl anziehende als auch abstoßende Kräfte mit steigenden oder konstanten Beträgen. Das vorherige Kapitel stellt drei verschiedene Beispielmetriken vor. Eine basiert auf dem Newtonschen Gravitationsgesetz und modelliert die Kraftfelder allein auf Basis der räumlichen Punktverteilung beider Wolken. Die anderen beiden Metriken sind vom Coulombschen Gesetz inspiriert und beziehen demnach zusätzliche Merkmale mit ein, um welche die Punktwolken erweitert werden können. Mit Hilfe der Merkmale werden die räumlichen Punktkorrespondenzen so gewichtet, dass die Kräfte entsprechend skaliert oder gar abstoßend wirken.

Analog zum Mehrteilchensystem der klassischen Kinematik werden die berechneten Kraftfelder, die auf sämtlichen Punkte des Templates wirken, in einem einzigen Massenschwerpunkt gebündelt. Die Krafteinwirkung wiederum verursacht gemäß der Starrkörpertransformation eine rigide Bewegung des Templates in Richtung des statischen Modells. Da sowohl die Kräfte mit steigender Registrierungsqualität zunehmen bzw. mindestens konstant bleiben, als auch die Punkte beider Wolken nicht aus Materie bestehen, kommt es vor allem am Ende der Registrierung zu größeren oszillierenden Punktwolkenbewegungen um das Optimum. Ein angepasstes Simulated Annealing regularisiert diese Bewegungen und sorgt zusätzlich dafür, dass die Registrierung nach einer bestimmten Anzahl an Iterationen terminiert. Auch dieser Bestandteil des Algorithmus ist von der Physik motiviert und leitet sich von thermodynamischen Prozessen ab.

Zusätzlich sind Strategien implementiert, die die Laufzeitkomplexität des Algorithmus senken. Basierend auf dem Monte-Carlo-Prinzip werden nicht alle Punkte aus beiden

Wolken für die Kraftfeldgenerierung berücksichtigt, sondern lediglich zufällige Stichproben. Obgleich kleinere Stichprobengrößen dafür sorgen, dass die Laufzeit des Algorithmus sinkt, können sie die Wirklichkeit lediglich näherungsweise abbilden. Da in diesem Fall nur wenige zufällig gewählte Punktkorrespondenzen die Kraftfelder generieren, ist es sehr wahrscheinlich, dass die Genauigkeit der Registrierung abnimmt. Dem wirken zwei weitere Strategien entgegen. Zunächst ist das Verfahren so formuliert, dass es effizient auf Mehrkernrechenarchitekturen berechnet werden kann. Das ermöglicht eine bedingte Erhöhung der Stichprobengröße, ohne zwingend die Laufzeitkomplexität zu steigern. Außerdem beschreiben neben homogenen Transformationsmatrizen auch Duale Quaternionen die Bewegungen der Punktwolke. Anders als mit den  $4 \times 4$ -Matrizen können mit der Hilfe Dualer Quaternionen mehrere Bewegungen pro Iteration berechnet und anschließend effizient zu einer Gesamttransformation gemittelt werden.

Im Folgenden wird untersucht, ob das neue Konzept ein valides Verfahren für die Punktwolkenregistrierung darstellt und der dazugehörige Algorithmus effizient arbeitet. Hierfür werden die grundlegenden Funktionalitäten des Algorithmus untersucht und mit dem aktuellen Stand der Technik verglichen.

Zunächst analysiert eine Machbarkeitsstudie, ob das neue Konzept vergleichbare Ergebnisse zum Goldstandard liefert. In dieser Machbarkeitsstudie 4.1 werden die zu vergleichenden Algorithmen anhand folgender vier Kriterien untersucht:

A Genauigkeit und Reproduzierbarkeit:

Wie präzise können zwei Punktwolken überlagert werden und wie oft kann das reproduziert werden?

B Effizienz:

Wie hoch ist die benötigte Zeit, um zwei Punktwolken miteinander zu registrieren und inwiefern beeinflusst die Auflösung sowohl Laufzeit als auch Genauigkeit?

C Robustheit:

Wie hoch ist die Registrierungsgenauigkeit bei unterschiedlich stark verrauschten oder fehlerhaften Punktwolken?

D Merkmalerweiterungen:

Kann der Algorithmus mit Merkmalerweiterungen umgehen? Wenn dies zutreffen sollte, um welches Maß kann die Genauigkeit dadurch gesteigert werden?

Neben eventuellen anwendungsspezifischen Kriterien stellen diese vier Punkte die grundlegenden Anforderungen dar, die ein Algorithmus zur Punktwolkenregistrierung erfüllen muss, wobei die vierte Anforderung D gesondert zu betrachten ist. Da Punktwolken nicht

zwingend mit Merkmalen erweitert sind, ist diese Anforderung kein zwangsläufiges Qualitätsmerkmal für alle Registrierungsalgorithmen, sondern fokussiert sich eher auf spezifische Anwendungen. Doch zunehmend mehr Sensor- und Kamerasysteme stellen solche erweiterten Informationen bereit, sodass diese Anforderung zukünftig unabdingbar sein wird und die Algorithmen entsprechend angepasst werden müssen.

Der ICP hat sich in den letzten Jahrzehnten als Goldstandard ausgezeichnet. Durch permanente Verbesserung und zahlreichen Erweiterungen erreicht der ICP in den meisten Anwendungsfällen bzw. Testszenarien überdurchschnittliche Ergebnisse. Aus diesem Grund ist es der ICP, an dem sich nahezu alle neu entwickelten Registrierungsalgorithmen zu messen haben. Ein Vergleich mit möglichst verschiedenen Registrierungsverfahren ist im Rahmen dieser Machbarkeitsstudie nicht möglich und sollte daher in zukünftigen Arbeiten sowie anwendungsspezifisch geschehen. Um jedoch möglichst objektiv zu bleiben, werden verschiedene Umsetzungen des ICP hinzugezogen.

Der Übersicht halber wird lediglich eine einzige Variante des neuen Verfahrens getestet. Veränderbare Parameter wie die Stichprobengröße des Monte-Carlo-Ansatzes oder die Einstellungsmöglichkeiten des Simulated Annealing werden festgesetzt und bleiben über die Machbarkeitsstudie hinweg gleich. Zudem wird nur die Variante des Algorithmus untersucht, welche die Punktwolkenbewegung mit Hilfe von homogenen Transformationsmatrizen beschreibt. Der Vergleich zu Dualen Quaternionen erfolgt im zweiten Abschnitt 4.2. Dort wird sich zeigen, ob die geglättete Bewegung einen entscheidenden Effizienzvorteil bringt. Außerdem wird im zweiten Abschnitt analysiert, wie groß der Einfluss aller frei wählbaren Parameter auf Genauigkeit und Laufzeit des Algorithmus ist.

Die objektive Grundwahrheit bilden Punktwolken sowohl aus standardisierten Realweltdaten als auch aus synthetisch generierten Datensätzen. In jedem der folgenden Experimente besteht ein Datensatz aus zwei Punktwolken, dem Modell  $\mathcal{X}$  und dem Template  $\mathcal{Y}$ . Die Transformation  $\mathcal{T}_S$  ist bekannt und überlagert beide Punktwolken optimal. In den einzelnen Versuchen eines Experiments wird das Template  $\mathcal{Y}$  vom Optimum weg bewegt. Jeder Algorithmus registriert  $\mathcal{Y}$  auf  $\mathcal{X}$  und gibt die jeweilige Transformation  $\mathcal{T}_R$  als Ergebnis zurück. Je kleiner der Abstand zwischen den beiden Transformationen  $\mathcal{T}_S$  und  $\mathcal{T}_R$  ist, umso kleiner ist auch der jeweilige Registrierungsfehler.

Als Abstandsmaß für den Registrierungsfehler dient der Root Mean Square Error (RMSE). Demnach wird das Template  $\mathcal{Y}$  sowohl um  $\mathcal{T}_S$  als auch um  $\mathcal{T}_R$  transformiert. Die beiden transformierten Wolken heißen  $\mathcal{Y}_S$  sowie  $\mathcal{Y}_R$ . Da beide Punktwolken gleich groß sind und identische Strukturen abbilden, sind auch die jeweiligen  $i$ -ten Punkte  $\mathbf{y}_i^S$  und  $\mathbf{y}_i^R$  direkt miteinander vergleichbar. Das  $i$ -te Punktepaar  $\mathbf{y}_i^S$  und  $\mathbf{y}_i^R$  sollte sich nach einer optimalen Registrierung exakt überlagern. Der Abstand zwischen den beiden Punkten

wäre folglich Null. Mit Hilfe des Euklidischen Abstands zwischen den korrespondierenden Punktpaaren kann demnach die Qualität der Registrierung gemessen werden. Der **RMSE** fasst die Abstände aller Punktpaare mit

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N \|\mathbf{y}_i^S - \mathbf{y}_i^R\|^2}{N}} \quad (4.1)$$

zusammen. Das heißt, je kleiner der **RMSE** ist um so höher ist die Genauigkeit des Registrierungsverfahrens.

Sämtliche Experimente in diesem Kapitel wurden auf derselben Hardware ausgeführt. Dabei handelt es sich um einen handelsüblichen Desktop PC, der mit einem INTEL Core i7-6700, 16 GB RAM Arbeitsspeicher und einer NVIDIA GeForce GTX 970 ausgestattet ist. Das Betriebssystem ist Microsoft Windows 10 64bit. Matlab wurde in der Version R2016a verwendet und **CUDA** mit Version 8.0.

## 4.1 Machbarkeitsstudie

Zunächst untersucht eine Machbarkeitsstudie<sup>1</sup>, ob das neu entwickelte Registrierungskonzept valide ist. Anhand verschiedener Varianten und Implementierungen des **ICP** wird dessen Konkurrenzfähigkeit zum Goldstandard geprüft. Überprüft werden die allgemeinen Kriterien eines Registrierungsalgorithmus: A die Genauigkeit sowie Reproduzierbarkeit der Registrierung, B die Effektivität des Algorithmus und C die Robustheit bei vertauschten Punktwolken. Das vierte Kriterium D untersucht den Einfluss von zusätzlichen Merkmalen auf die Registrierungsgenauigkeit.

Das neue Verfahren wurde hierfür in Matlab [63] implementiert. Zur Übersicht ist die Punktwolkenbewegung allein durch homogene Transformationsmatrizen repräsentiert. Eine Glättung der Einzelbewegung ist mithin noch nicht umgesetzt. Ferner sind alle Metriken aus Kapitel 3.2 integriert. Trotzdem wird in den Experimenten A bis C lediglich die Metrik analog zum Newtonschen Gravitationsgesetz (**KNG**) untersucht. Denn bei Punktwolken ohne Merkmalerweiterungen verhalten sich die drei Beispielmetriken gleich und müssen nicht separat betrachtet werden. Lediglich im vierten Experiment D zur markmbasierten Punktwolkenregistrierung werden die anderen beiden Metriken gesondert betrachtet, die sich aus dem Coulombschen Gesetz der Elektrostatik ableiten und Kräfte sowohl ohne Abstoßung (**KCO**) als auch mit (**KCM**) modellieren.

Verglichen wird der neue Algorithmus mit dem **ICP**. Um eine möglichst hohe Objektivität zu gewähren, werden insgesamt drei verschiedene Varianten getestet. Der erste **ICP**

<sup>1</sup>Die Ergebnisse dieser Machbarkeitsstudie wurden teilweise in ähnlicher Form bereits in [38] veröffentlicht.

wird von MATLAB R2016a bereitgestellt und ist in der Computer Vision System Toolbox integriert [63]. Hierbei handelt es sich um eine Punkt-zu-Punkt-Variante (IMP) und eine Punkt-zu-Ebene-Variante (IME). Eine weitere gängige Version des ICP wurde von Kjer und Wilm ebenfalls in MATLAB implementiert [50, 51]. Auch hier geschieht die Fehlerminimierung mittels Punkt-zu-Punkt- (IKP) und Punkt-zu-Ebene-Methode (IKE). Die letzte ICP-Implementierung stammt aus der Point Cloud Library<sup>2</sup> (PCL) [35, 89]. Diese ist eine der bekanntesten und am weitesten verbreitete Softwarebibliotheken zur Ver- und Bearbeitung von Punktwolkendaten in C++. Die hier verwendeten Varianten basieren gleichermaßen auf der Punkt-zu-Punkt- (IPP) und Punkt-zu-Ebene-Metrik (IPE) des ICP.

Sämtliche zuvor genannten Implementierungen des ICP verwenden lediglich räumliche Strukturen für die Registrierung. Um auch Punktwolken zu registrieren, deren Punkte um zusätzliche Merkmale erweitert sind, wird eine weitere Variante des ICP zur Evaluation hinzugezogen. Diese basiert auf IKP bzw. IKE und erweitert diese um einen zusätzlichen Farbraum. Bezüglich der Arbeiten [40, 54] werden anstelle eines dreidimensionalen Farbraums dynamisch dimensionierte Farbräume mit  $D \geq 1$  berücksichtigt. Voraussetzung dafür ist, dass die zu registrierenden Punktwolken die gleiche Dimension  $D$  besitzen. In diesem Fall werden die Punktkorrespondenzen nicht ausschließlich über die 3D-Rauminformationen ermittelt, sondern im  $(3 + D)$ -dimensionalen Raum. Für die Analyse werden ebenfalls die Punkt-zu-Punkt- (IFP) und Punkt-zu-Ebene-Varianten (IFE) verwendet. Da sich diese beiden Implementierungen bezüglich Punktwolken ohne Merkmalerweiterungen nicht von deren ursprünglichen Vertretern IKP und IKE unterscheiden, werden IFP bzw. IFE ausschließlich im vierten Experiment D analysiert.

#### 4.1.1 A: Genauigkeit und Reproduzierbarkeit

Das erste Experiment ist so gestaltet, dass sowohl die Genauigkeit als auch die Reproduzierbarkeit von Registrierungsergebnissen evaluiert werden können. Hierfür wird der Datensatz eines Rekonstruktionsbeispiels verwendet. Das heißt, ein Objekt wird von mehreren Seiten mittels Tiefenbildsensoren aufgenommen. Die Registrierung überlagert die einzelnen Teilaufnahmen miteinander, sodass schlussendlich das vollständige Objekt als eine einzige Punktwolke repräsentiert wird. Datenbanken wie das Stanford 3D Scanning Repository stellen solche standardisierten Punktwolkendaten für Evaluationszwecke zur Verfügung [58]. Bei den dort gegebenen Datensätzen ist die exakte Transformation zwischen den einzelnen Aufnahmen stets bekannt. Somit können neue Registrierungsverfahren bzw. Erweiterungen etablierter Algorithmen objektiv getestet und verglichen

---

<sup>2</sup>Version 1.8

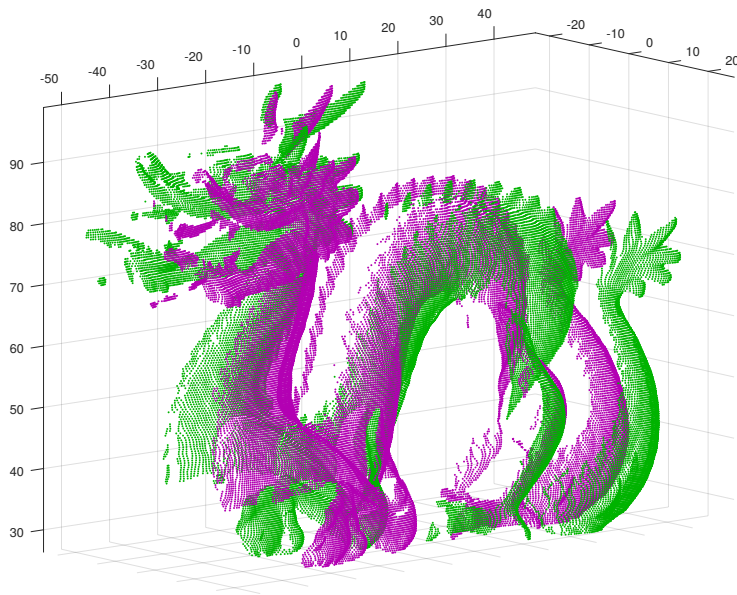


Abbildung 4.1: Dragon-Datenansatz aus Experiment A. Dieser repräsentiert eine Rekonstruktion von Punktwolken, bei der beide Punktwolken im selben Bezugssystem der Kamera abgebildet werden und das Template (magenta) auf das Modell (grün) registriert werden muss.

werden.

In diesem Experiment werden zwei Punktwolken des sogenannten Dragon-Datenansatzes verwendet. Dieser Datensatz bildet eine Drachenstatue ab, die mit Hilfe eines Cyberware 3030 MS Laserscanners aufgenommen wurde. Insgesamt besteht der Datensatz aus 70 Aufnahmen diverser Blickwinkel auf die Drachenbüste. Da in diesem Experiment nicht die Rekonstruktion vollständiger Objekte im Vordergrund steht, sondern die Registrierungsgenauigkeit zweier Punktwolken im Allgemeinen, werden lediglich zwei dieser Aufnahmen aus dem Datensatz verwendet. Beide Punktwolken bilden den Drachen mit einem Blickwinkelunterschied von ungefähr  $24^\circ$  ab. Demzufolge gibt es in beiden Punktwolken viele Strukturen, die sich eindeutig überlagern. Aber es gibt auch Regionen in beiden Wolken, die sich nicht decken. Durch die Aufnahme aus verschiedenen Blickwinkeln ist es sehr unwahrscheinlich, dass zwischen den beiden Punktwolken direkte Punkt-zu-Punkt-Korrespondenzen existieren. Das heißt, sogar bei sinngemäß optimaler Deckung beider Punktwolken liegen nur wenige Punkte exakt übereinander.

In der Abbildung 4.1 sind beide Punktwolken abgebildet. Diese sind im Bezug zum

Koordinatensystem des Laserscanners dargestellt und demnach um ca.  $24^\circ$  verschoben. Hierbei repräsentiert die grüne Punktwolke das Modell. Es besteht aus 41,841 Punkten und hat eine Ausdehnung von ca. (102, 73, 36) Einheiten. Das Template besteht aus 34,836 Punkten und hat eine ungefähre räumliche Ausdehnung von (98, 72, 50) Einheiten.

Der Datensatz stellt die nötige Transformation  $\mathcal{T}_S$  zur Verfügung, die das Template in einer Art und Weise bewegt, dass sich beide Punktwolken optimal überlagern. Diese Transformation bildet demnach die Grundwahrheit des Experiments. Das Experiment besteht aus jeweils 1,000 Versuchen. Zu Beginn eines Versuchs wird das Template um die Inverse Transformation  $\mathcal{T}_S^{-1}$  verschoben. Aus diesem Grund muss jeder Algorithmus die beiden Punktwolken miteinander registrieren und für sich die Transformation  $\mathcal{T}_R$  berechnen. Zwischen dem Template, das mit  $\mathcal{T}_S$  transformiert ist und dem, welches um  $\mathcal{T}_R$  verschoben ist, wird pro Algorithmus der **RMSE** berechnet. Zusätzlich wird für jeden Algorithmus die Laufzeit gemessen, die er für eine Registrierung benötigt.

Die Ergebnisse des Experiments sind in Tabelle 4.1 zusammengetragen. In der ersten Zeile sind die jeweiligen Algorithmen benannt. Die zweite Zeile gibt die durchschnittliche Laufzeit der einzelnen Algorithmen in Sekunden an. In den drei darauffolgenden Zeilen werden die wichtigen Aspekte des **RMSE** zusammengefasst: Der Median (**Med**), der Interquartilabstand (**IQA**) und die gesamte Bandbreite (**Bbr**) der 1,000 Versuche. Zur Visualisierung der **RMSE** dienen die jeweiligen Boxplots in der letzten Zeile. Aufgrund der geringen Fehlervarianzen und für den besseren optischen Vergleich ist die Achse des Boxplot logarithmisch skaliert.

Die Ergebnisse dieses Experiments zeigen, dass alle Algorithmen eine hohe Genauigkeit bei der Registrierung erreichen. Der Median des Registrierungsfehlers liegt bei sämtlichen Verfahren unter 0.4 Einheiten und das obwohl die räumliche Ausbreitung beider Punktwolken bis zu 100 Einheiten beträgt. Die **ICP**-Varianten weisen stets den gleichen Fehler auf, haben also einen **IQA** und eine Fehlerbandbreite von jeweils Null. Das bedeutet, die Reproduzierbarkeit dieser Registrierungsverfahren ist optimal. Das liegt an den gleichbleibenden Ausgangsvoraussetzungen bei der Registrierung. Weder die Struktur der Punktwolken noch die Anzahl der Punkte pro Wolke ändern sich. Auch der Versatz zwischen den Punktwolken ist zu Beginn der Registrierung identisch. Dementsprechend liefern **ICP**-Verfahren, aufgrund ihres deterministischen Verhaltens, ein gleichbleibendes Registrierungsergebnis.

Der **KNG** hingegen weist eine höhere Fehlervarianz bei der Registrierung auf. Ein Grund dafür sind die zufällig generierten Kraftfelder. Der Monte-Carlo-Ansatz wählt in

Tabelle 4.1: Die Ergebnisse zum Experiment A. Zusammengefasst sind die durchschnittliche Laufzeit der Algorithmen in  $s$  und der Registrierungsfehler gemessen am RMSE. Der RMSE wiederum wird durch den Median, den Interquartilabstand, der Bandbreite und den logarithmisch skalierten Boxplot repräsentiert. Die Werte ergeben sich aus 1,000 Registrierungen zweier Punktwolken des Dragon-Datensatzes.

Alg.	KNG	IMP	IME	IKP	IKE	IPP	IFE
Durchschnittliche Laufzeit							
in $s$	0.63	0.10	0.05	1.08	1.42	1.72	0.78
RMSE							
Med	0.07	0.26	0.30	0.26	0.21	0.21	0.26
IQA	0.03	0.00	0.00	0.00	0.00	0.00	0.00
Bbr	44.21	0.00	0.00	0.00	0.00	0.00	0.00
RMSE-Boxplot							

jeder Iteration unterschiedliche Punktkorrespondenzen für die Kraftfeldgenerierung aus. Daraus folgen leichte Variationen in der Bewegung Templates, was wiederum dazu führt, dass der Algorithmus nicht deterministisch ist. Ein weiterer Grund für die hohe Fehlervarianz ergibt sich aus der Transformationsdarstellung durch homogene  $4 \times 4$ -Matrizen. Um die Laufzeit gering zu halten, werden notwendige Orthogonalitätsbedingungen beim Umwandeln von Achse-Winkel-Darstellung in Rotationsmatrizen vernachlässigt. Obwohl die damit verbundenen Rotationsfehler meist gering ausfallen, können sich diese über mehrere Iterationen zu größeren Registrierungsfehlern akkumulieren.

Andererseits verhindert der Monte-Carlo-Ansatz, dass der Algorithmus nicht in lokalen Optima hängen bleibt. Das führt u. a. dazu, dass der Median im Vergleich zu den anderen Verfahren sehr viel kleiner ist. So haben die ICP-Algorithmen durch ihr deterministisches Verhalten zwar keine Varianzen in den Ergebnissen, können aber auch nicht falschen lokalen Optima ausweichen. Dass sogar der Median und Interquartilabstand zusammen genommen weit unter den Medianen der ICP-Varianten liegt, zeugt dafür, dass das neue Konzept ein Registrierungsverfahren mit hoher Genauigkeit ist. Negativ sind die vereinzelt auftretenden großen Ausreißer. Gleichwohl liegt die Spannweite des RMSE bis auf zwei sehr große Fehler unter einer Einheit und ist in den meisten Fällen vergleichbar mit den Medianen der ICP-Varianten. Aufgrund dieser geringen Anzahl an Ausreißern besteht die Hoffnung, dass die Reproduzierbarkeit der Kraftfeldmethode gesteigert werden kann, wenn die Punktwolkenbewegung zusätzlich gefiltert und geglättet wird.

Schließlich liegt die Laufzeit des KNG im Mittel und ist vergleichbar mit der Punkt-zu-Ebenen-ICP der PCL. Sowohl KNG als auch IPE liegen knapp unter einer Sekunde. Besonders schnell registrieren die beiden ICP-Implementierungen, die von Matlab bereitgestellt werden. Die Punkt-zu-Ebenen-Variante liegt sogar bei rund 0.05 s. Alle anderen Algorithmen benötigen bei solch einer Punktwolkenauflösung im Schnitt mehr als eine Sekunde.

#### 4.1.2 B: Effizienz

In diesem Experiment wird die Effektivität der Registrierungsalgorithmen analysiert. Diese stellt das Verhältnis zwischen Genauigkeit und Laufzeit in Abhängigkeit der Punktwolkenauflösung dar. Das heißt, die Effektivität bewertet, wie viel Zeit ein Algorithmus benötigt, bis er ein Ergebnis mit angemessener Genauigkeit berechnet hat.

Für dieses Experiment wird die Punktwolke eines Handskeletts verwendet (siehe Abbildung (4.2)). Der entsprechende Datensatz wird vom Large Geometric Model Archive des Georgia Institute of Technology bereitgestellt [101]. Das Abbild des Handskeletts

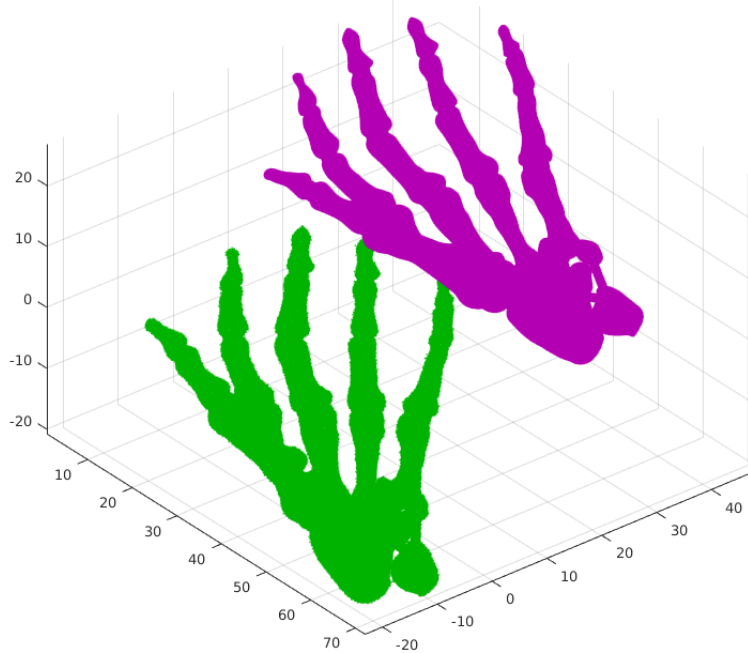


Abbildung 4.2: Sehr hoch aufgelöste Punktwolken mit jeweils 327,323 Punkten. Für das Experiment sind Modell (grün) und Template (magenta) um die dargestellte zufällige Transformation verschoben. Dieser Abstand ist für das gesamte Experiment gleichbleibend.

besteht aus insgesamt 327,323 Punkten und hat eine räumliche Ausdehnung von ca. (66,45,23) Einheiten. Bei so hohen Auflösungen stoßen viele Registrierungsalgorithmen an ihre Grenzen. Damit die Algorithmen dennoch auf solch hoch aufgelöste Datensätze angewandt werden können, wird die Punktwolkenauflösung oftmals direkt vor der Registrierung reduziert. Dies kann dazu führen, dass eindeutige Strukturen gelöscht werden. Ohne diese eindeutigen Zuordnungen wiederum steigt die Fehlerrate bei der Korrespondenzbestimmung pro Iteration. Diese Fehler oder Ungenauigkeiten ziehen sich dann durch den gesamten Registrierungsprozess. Andernfalls könnte auch die Auflösung in den einzelnen Registrierungsschritten erneut und jedes Mal im unterschiedlichen Ausmaß verringert werden. In diesem Fall steigt jedoch oftmals die Laufzeit der Algorithmen sehr stark an. Folglich ist es wichtig zu wissen, ab wann ein Algorithmus nicht mehr effizient arbeitet.

In diesem Experiment wird die gleiche Punktwolke des Handskeletts sowohl für das Modell als auch für das Template verwendet. Somit bilden beide Punktwolken exakt die gleichen Strukturen ab. Der einzige Unterschied zwischen beiden entsteht durch die je-

weilige Wahl der Punktstichproben. Ausgehend von einer sehr kleinen Stichprobe von je 327 zufällig gewählten Punkten pro Wolke, werden in 26 Durchläufen zunehmend mehr Punkte zur Stichprobe hinzugefügt. Im letzten Durchlauf bestehen Modell und Template schließlich aus jeweils 327,323 Punkten. Dabei geschieht die Wahl der Punkte für das Modell und für das Template separat voneinander und zufällig. Folglich ist es anfänglich sehr unwahrscheinlich, dass es zwischen dem Modell und Template direkte Punktkorrespondenzen gibt. Diese Wahrscheinlichkeit wächst jedoch stetig mit der Stichprobengröße beider Wolken.

Jeder Durchlauf besteht aus 100 Einzelversuchen, in denen jeder Algorithmus das Template auf das Modell registriert. Da sowohl Template als auch Modell aus der gleichen Punktwolke bestehen und sich demzufolge beide optimal überlagern, wurde vor dem Experiment eine zufällige Transformation für das Template generiert. Diese ist in dem gesamten Experiment gleich und beschreibt einen linearen Versatz von (-2, -19, -14) Einheiten pro Raumachse sowie eine Rotation von  $22.44^\circ$  um die zufällige Rotationsachse (-0.50, 0.26, -0.82). Diese Transformation ist in Abbildung 4.2 dargestellt, wobei das Modell grün und das Template magenta eingefärbt sind. In dieser Darstellung haben beide Wolken ihre volle Auflösung. Demnach sind in der Abbildung die einzelnen Punkte kaum auszumachen.

Die Ergebnisse dieses Experiments sind in Abbildung 4.3 zusammengefasst. In beiden Graphen sind auf der x-Achse die verschiedenen Auflösungen der 26 Durchläufe aufgetragen. Der obere Graph stellt die durchschnittlichen Laufzeiten für jeden Algorithmus dar. Im unteren Graph ist der RMSE-Median über die jeweils 100 Versuche pro Auflösung aufgetragen. Zur Übersicht ist die Achsenskalierung der RMSE-Median-Werte logarithmisch. Sobald einer der Algorithmen für einen Durchlauf durchschnittlich länger als 15 s für die Registrierung benötigt, wird dieser für die nachfolgenden Durchläufe nicht mehr berücksichtigt. Algorithmen, die soviel Zeit für die Registrierung benötigten, sind hinsichtlich vieler Anwendungen nicht mehr effizient genug.

Das Experiment zeigt deutlich, dass sämtliche ICP-Implementierungen einen exponentiellen Anstieg der Laufzeit bezüglich der Punktwolkenauflösung haben. Jedoch sind diese unterschiedlich stark ausgeprägt. Den steilsten Anstieg der Laufzeit haben die beiden ICP-Implementierungen IKP und IKE, gefolgt von IPP und IPE. Am effizientesten sind die beiden Implementierungen IMP und IME, die von Matlab bereitgestellt werden. Obgleich sämtliche ICP-Varianten anfänglich eine signifikant niedrige Laufzeit von weit weniger als einer Sekunde haben, können sie diese gute Performanz nicht bei höhere Auflösungen halten. Bereits bei einer Auflösung von 33,060 Punkten pro Wolke benöti-

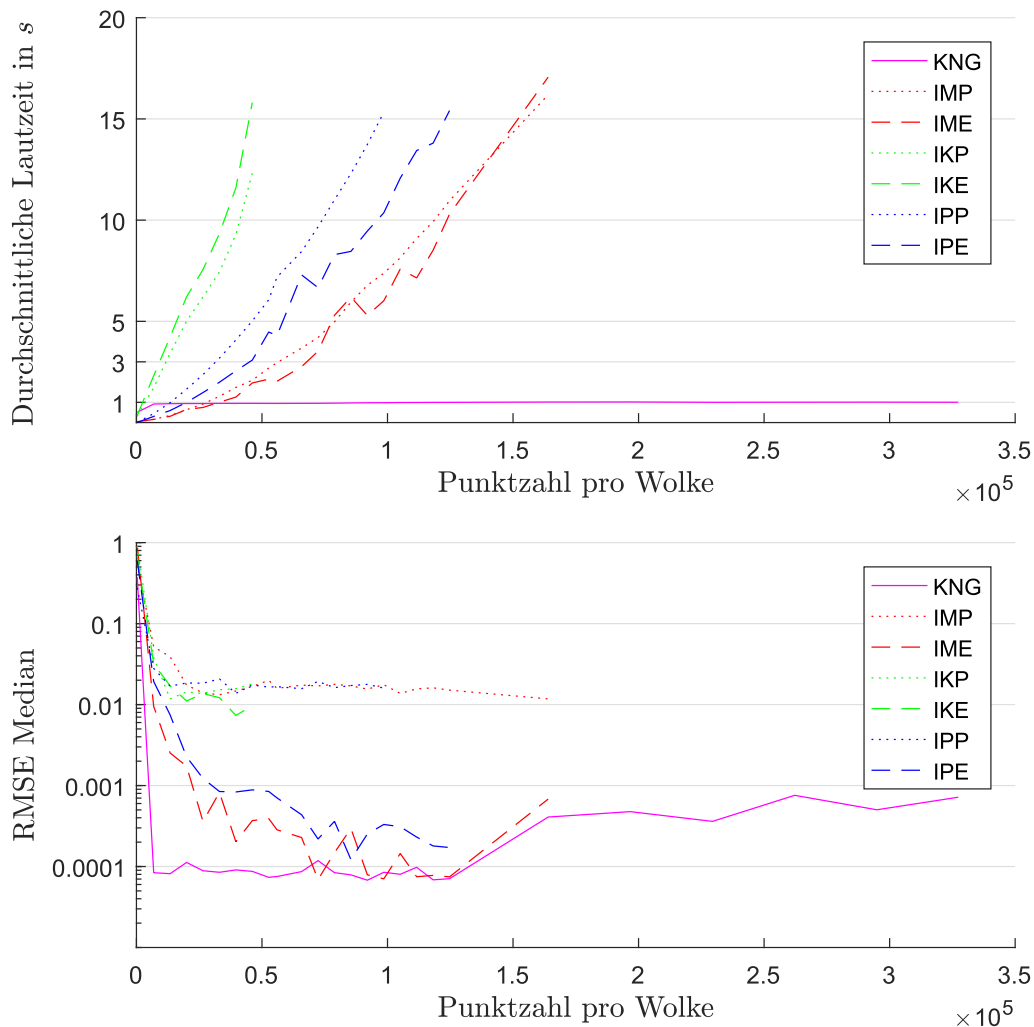


Abbildung 4.3: Ergebnisse zum Effektivitätsexperiment B: Der obere Graph zeigt die Abhängigkeiten von Punktzahl zu Laufzeit, der untere die Abhängigkeit von Punktzahl zu logarithmisch dargestellten RMSE-Median. Jeder Algorithmus musste pro Punktzahl 100 Registrierungen durchführen.

gen ausschließlich alle ICP-Algorithmen mehr als 1 s. Spätestens bei einer Auflösung von 163,989 Punkten brauchen sämtliche ICP-Varianten sogar mehr als 15 s. Lediglich der KNG registriert Punktwolken mit fast doppelt so hoher Auflösung bei ungefähr gleichbleibender Geschwindigkeit.

Die Laufzeit ist annähernd konstant und liegt bei ca. einer Sekunde. Dieses Ergebnis ist jedoch zu erwarten, denn das Verfahren basiert auf dem Monte-Carlo-Ansatz und

verwendet für die Registrierung permanent dieselbe Stichprobengröße pro Iteration. In dieser Machbarkeitsstudie sind das jeweils 1,024 Punkte pro Wolke. Der leichte Anstieg von rund 0.09 s über alle Durchläufe entsteht vorwiegend aus notwendigen Vorverarbeitungsschritten. Aber auch der Datentransfer zwischen CPU und GPU vor Beginn einer Registrierung steigt mit zunehmender Punktwolkengröße an und verursacht geringe Laufzeitanstiege.

Die Genauigkeit der Registrierung ist bei sämtlichen Algorithmen sehr hoch. Besonders stechen der KNG und die beiden Punkt-zu-Ebene-ICP IPE und IME hervor. Diese haben größtenteils einen RMSE-Median von weit unter 0.001 Einheiten. Wobei der KNG in den meisten Fällen den niedrigsten RMSE-Median aufweist. Dieser liegt teilweise sogar unter 0.0001 Einheiten. Zwar steigt der Fehler auch beim KNG ab einer Punktzahl von ca. 125,000 an, aber selbst bei über 300,000 Punkten pro Wolke ist die Genauigkeit des KNG vergleichbar bis höher zu den ICP-Varianten. Am schlechtesten schneiden IMP, IKP, IKE und IPP ab. Trotzdem registrieren diese ICP-Varianten mit einer hohen Genauigkeit von 0.1 Einheiten.

Anzumerken ist, dass alle Algorithmen erst ab ca. 6,000 Punkten pro Wolke einen RMSE-Median von unter 0.1 Einheiten erreichen. Das kann darauf zurückgeführt werden, dass die Punktwolken bei einer so geringen Auflösung nur wenig eindeutige Strukturen besitzen. Demzufolge fehlen allen Algorithmen aussagekräftige Punktkorrespondenzen. Solch ein geringer Informationsgehalt wirkt sich eindeutig negativ auf das Ergebnis der Registrierung aus.

### 4.1.3 C: Robustheit

Ein Registrierungsverfahren heißt robust, wenn es bei stark verrauschten oder fehlerhaften Punktwolkendaten eine hohe Genauigkeit erreicht. In diesem Experiment werden die Algorithmen hinsichtlich ihrer Robustheit analysiert. Hierfür wird ebenfalls ein weitverbreiteter Testdatensatz verwendet, die sogenannte Utah-Teekanne. Dieser synthetische Datensatz der Computergrafik wird aufgrund seiner hohen Akzeptanz von mehreren Punktwolkenbibliotheken bereitgestellt. In dieser Arbeit wird der Datensatz verwendet, der in MATLAB integriert ist.

Ursprünglich besteht die Punktwolke aus 41,472 Punkten und hat eine Dimension von ca. (129,80,63) Einheiten. Aus Komplexitätsgründen und der damit einhergehenden Gewährleistung der Fairness wurde die Anzahl Punkte sowohl für das Modell als auch für das Template auf jeweils 5,391 bzw. 3,318 Punkte reduziert. Die Wahl dieser Stichproben erfolgt zufällig, sodass auch hier davon auszugehen ist, dass in beiden Punktwolken nur

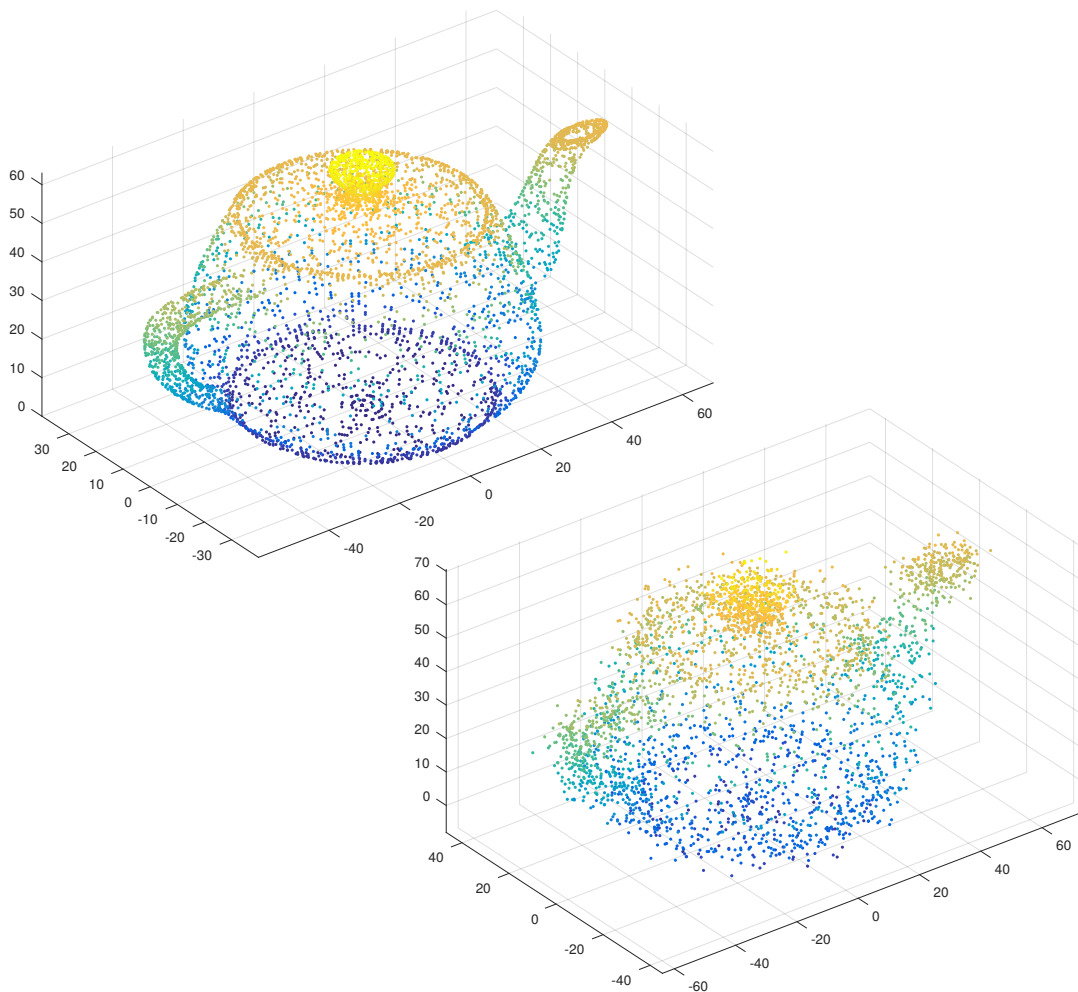


Abbildung 4.4: Utah-Teekanne ohne (links oben) und mit (rechts unten) künstlichem normalverteilten Rauschen. Die Varianz des Rauschens beträgt in diesem Fall  $\sigma^2 = 2.9$  pro Punkt.

wenig direkte Punkt-zu-Punkt-Korrespondenzen vorhanden sind.

Im weiteren Verlauf des Experiments wird das Template in 30 Durchläufen zunehmend stark verrauscht. Das Rauschen ist normalverteilt und wird für jeden einzelnen Punkt des Templates separat um alle drei Raumachsen modelliert. Beginnend mit  $\sigma^2 = 0.0$  wird in jedem Durchgang des Experiments die Varianz der Normalverteilung solange um  $\sigma^2 = 0.1$  erhöht, bis  $\sigma^2 = 2.9$ . Der Unterschied zwischen einer Punktwolke mit maximalen Rauschen und einer Wolke ohne ist in Abbildung 4.4 visualisiert.

In jedem Durchgang und demnach für jedes Rauschlevel werden jeweils 100 Versuche

durchgeführt. Da sich beide Punktwolken optimal überlagern, wird eine Kopie des Templates um eine zufällige Transformation verschoben. Hierbei sind Translationen in einem Bereich von  $[-50, +50]$  Einheiten und Rotation zwischen  $[-20, 20]^\circ$  erlaubt, jeweils um sämtliche Raumachsen. Schließlich muss jeder Algorithmus das verschobene Template auf das Modell registrieren. Danach kann der **RMSE** zwischen originalem Template und der registrierten Kopie gemessen werden.

In den Abbildungen 4.5 und 4.6 sind die Ergebnisse des Experiments dargestellt. In beiden Graphen sind auf der  $x$ -Achse die Varianzen der Normalverteilung und aufgrund dessen die Rauschlevel abgebildet. Auf der  $y$ -Achse sind Medianwerte des **RMSE** über die 100 Versuche pro Durchgang aufgetragen. In Abbildung 4.5 ist, zusätzlich zu den Genauigkeiten der Algorithmen, der **RMSE**-Erwartungswert (EW) für das jeweilige Rauschlevel eingezeichnet. Dieser gibt an, wie groß der **RMSE** zwischen originalen Template und dessen verrauschten Kopie ist. Da die Registrierungsgenauigkeit sämtlicher Algorithmen weit unter dem Erwartungswert liegt, zeigt die Abbildung 4.6 die Ergebnisse nochmals ohne Erwartungswert. Hier lassen sich die Unterschiede zwischen den einzelnen Algorithmen deutlicher ausmachen.

Im ersten Durchgang und ohne Rauschen ist der Erwartungswert des **RMSE** gleich Null. Es ist sehr unwahrscheinlich, dass irgendein Registrierungsalgorithmus diesen Wert erreicht. Jedoch zeigen die hier getesteten Algorithmen eine sehr hohe Genauigkeit. Außerdem ist erkennbar, dass die **RMSE**-Mediane sämtlicher Algorithmen zwar näherungsweise dem linearen Anstieg des Erwartungswerts folgen, aber weit unter diesem bleiben. Demzufolge können die Verfahren ausnahmslos als sehr robust eingestuft werden.

Die Unterschiede zwischen den einzelnen Algorithmen werden erst deutlicher, wenn der Erwartungswert ausgeblendet wird und der Ausschnitt vergrößert werden kann (siehe 4.6). Dann zeigt sich, dass der **IKE** am wenigsten robust ist. Die anderen Algorithmen zeigen eine annähernd ähnliche Robustheit. Es ist jedoch zu erkennen, dass die Punkt-zu-Ebenen-Varianten des **ICP** zunehmend ungenauer werden, je mehr das Rauschlevel steigt. Der Grund hierfür liegt in der hohen Rauschanfälligkeit der Ebenengenerierung. Die Normalenvektoren der Ebenen werden jeweils aus mehreren benachbarten Punkten generiert. Demnach wird auch das Rauschen mehrerer Punkte akkumuliert und die Ebene wird mit hoher Wahrscheinlichkeit noch mehr vom Ursprung abweichen, als das für einzelne Punkte der Fall ist. Daher sind sämtliche Punkt-zu-Punkt-Varianten des **ICP** robuster. Die Punkt-zu-Punkt-Korrespondenzen gleichen das normalverteilte Rauschen im Mittel aus.

Die **ICP**-Varianten bieten bei wenig Rauschen eine höhere Genauigkeit als der **KNG**.

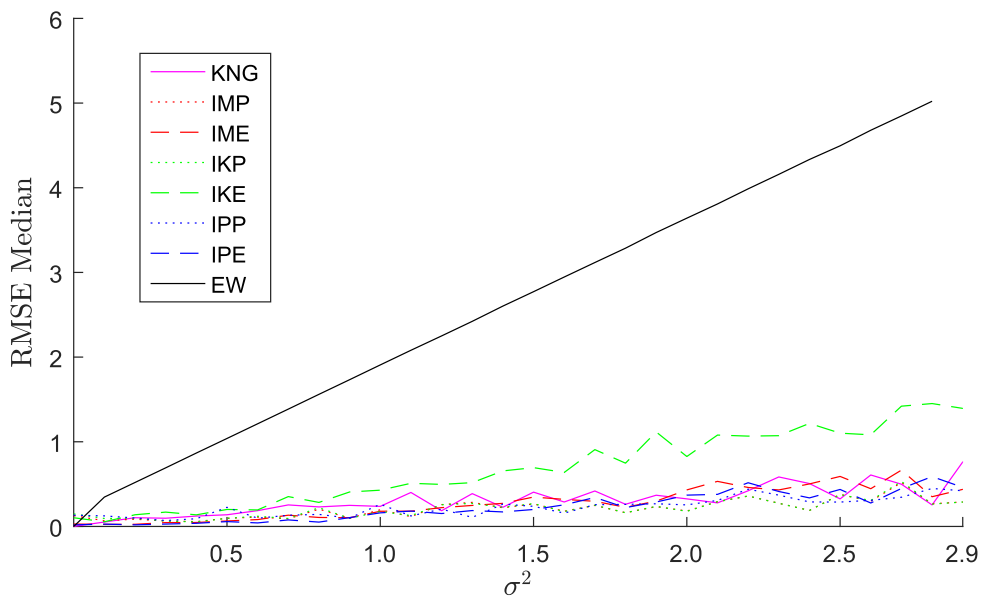


Abbildung 4.5: Ergebnisse zum Experiment C. Der Registrierungsfehler repräsentiert durch den RMSE-Median über 100 Registrierungen ist dem Rauschlevel  $\sigma^2$  gegenübergestellt. Zusätzlich ist der zu erwartende Registrierungsfehler (EW) aufgetragen.

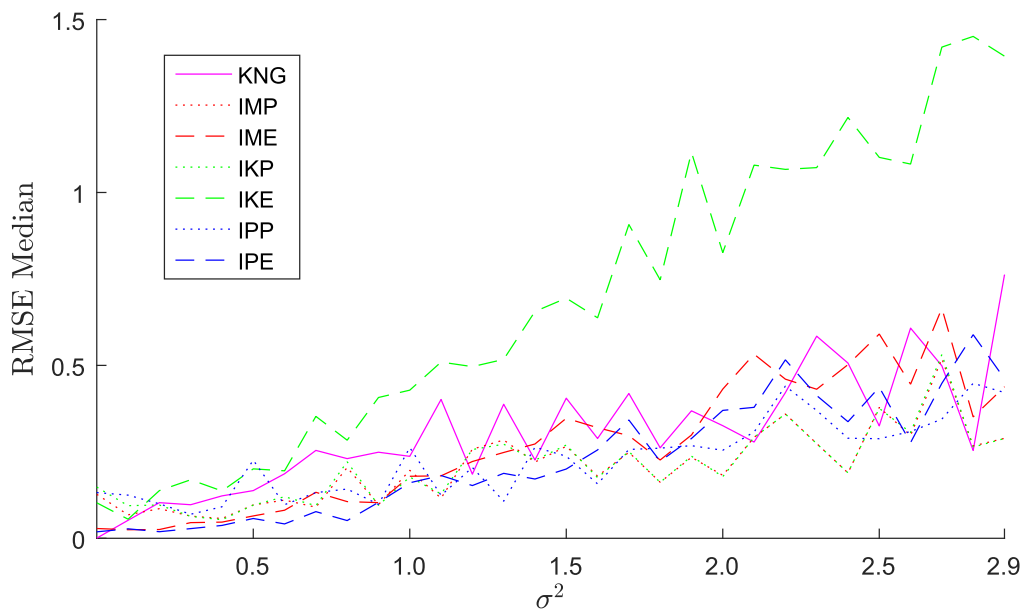


Abbildung 4.6: Ergebnisse zum Experiment C. Der Registrierungsfehler repräsentiert durch den RMSE-Median über 100 Registrierungen ist dem Rauschlevel  $\sigma^2$  gegenübergestellt.

Mit zunehmenden Rauschlevel gleichen sich die **RMSE**-Mediane jedoch allmählich an. Da beim **KNG** die Korrespondenzen über Kraftfelder berechnet werden, die wiederum indirekt den Abstand zwischen mehreren Template-Punkten zu vielen unterschiedlichen Modell-Punkten messen, wird auch hier das gleichverteilte Rauschen im Mittel ausgeglichen. Mit Ausnahme des **IKE** sind die getesteten Algorithmen ähnlich robust. Daraus lässt sich schließen, dass das neue Verfahren robust genug ist, um in der Punktwolkenregistrierung eingesetzt werden zu können.

#### 4.1.4 D: Merkmalerweiterung

Das letzte Experiment der Machbarkeitsstudie untersucht die Performanz der Algorithmen bei Punktwolken, die nur wenig räumliche Strukturen besitzen, aber dafür um zusätzliche Merkmale erweitert sind. Explizit dafür wurde ein besonders anspruchsvoller Punktwolkendatensatz generiert. Hierbei handelt es sich um die Oberfläche einer Halbkugel, die zusätzlich mit einem charakteristischen Muster eingefärbt ist. Die Halbkugel ist in Abbildung 4.7 a) dargestellt. Sie hat einen Radius von 50 Einheiten. Der Farbraum ist 1D und hat einen Wertebereich von  $[0, 1]$ .

Die anderen Darstellungen aus Abbildung 4.7 zeigen die einzelnen Schritte der Generierung des Datensatzes. Zunächst wurden auf der Halbkugel zwei unabhängige Bereiche segmentiert. Diese Segmente sind in Abbildung 4.7 b) visualisiert, jeweils in magenta und grün. Obgleich sich beide Bereiche an vielen Stellen überschneiden, sollte es einige räumliche und farbliche Unterschiede geben. Außerdem sollten die beiden quadratischen Ausschnitte nicht die gleiche Orientierung und Größe besitzen. Schließlich wurden jeweils beide Segmente unabhängig voneinander und zufällig gerastert. Das Resultat sind die beiden Punktwolken aus Abbildung 4.7 c) und 4.7 d). Somit wird auch hier gewährleistet, dass direkte Punkt-zu-Punkt-Korrespondenzen in den überlappenden Bereichen möglich sind, aber sehr unwahrscheinlich auftreten.

Die Punktwolke aus Abbildung 4.7 c) ist das Modell und besteht aus insgesamt 1,095 Punkten. Abbildung 4.7 d) zeigt das Template, das aus 971 Punkten besteht. Da sich beide Punktwolken optimal überlagern, wird in dem Experiment eine Kopie des Templates 1,000-Mal um eine zufällige Transformation verschoben:  $[-50, 50]$  Einheiten translatorisch und  $[-20, +20]^\circ$  rotatorisch je Raumachse. In jedem Versuch müssen die Algorithmen die Templatekopie auf das Modell registrieren. Der **RMSE** berechnet sich aus dem originalen Template und dessen registrierte Kopie.

Die Ergebnisse des Experiments sind, analog zu Experiment A, in Form einer Tabelle 4.2 dargestellt. Zusätzlich zu den bisher getesteten Registrierungsalgorithmen evaluiert

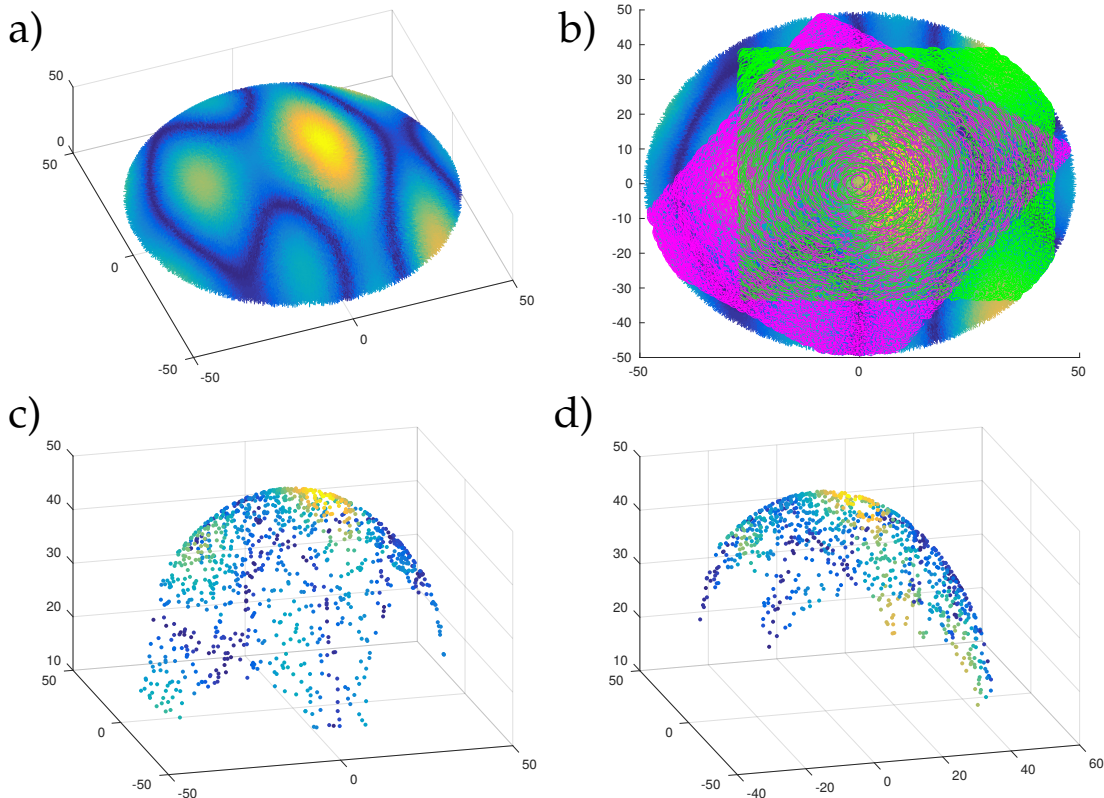
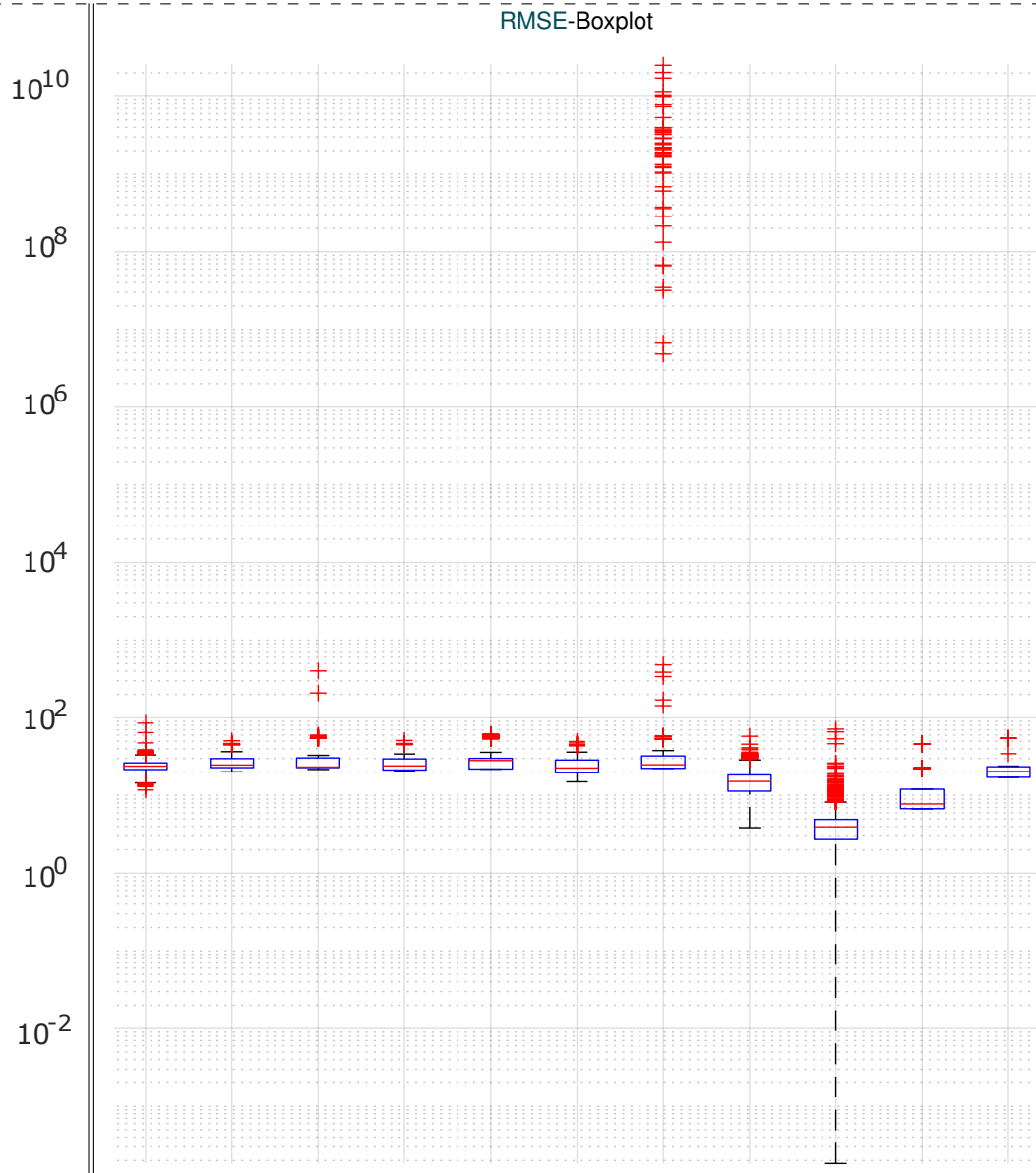


Abbildung 4.7: a) Synthetisch generierte Farbhalkugel aus der b) zwei sich teilweise überlappende Segmente gewählt wurden. c) und d) zeigen die Punktwolken, die sich aus der jeweils unabhängigen zufälligen Rasterung beider Segmente ergeben.

dieser Versuch ebenso diejenigen Verfahren, die Merkmalerweiterungen bei der Registrierung berücksichtigen. Das sind zum einen die beiden Varianten **KCO** und **KCM**, die in dieser Arbeit beschrieben werden. Beide registrieren merkmalsbehaftete Punktwolken mit Hilfe von Metriken, die dem Coulombschen Gesetz nachempfunden sind. **KCO** modelliert lediglich anziehende Kräfte, skaliert diese aber entsprechend der Merkmalsunterschiede. **KCM** berücksichtigt zudem abstoßende Kräfte, die entstehen, wenn der Abstand zwischen den Merkmalen korrespondierender Punkte zu groß ist. Zum anderen werden bei zwei erweiterten **ICP**-Varianten **IFP** und **IFE** berücksichtigt. Ersterer beruht auf dem Punkt-zu-Punkt-Fehlermaß und letzterer auf der Punkt-zu-Ebene-Metrik. In beiden Fällen werden die Korrespondenzen nicht ausschließlich über deren räumliche **3D**-Lage ermittelt, sondern zusätzlich über den entsprechend **1D**-Merkmalsraum. Das heißt,

Tabelle 4.2: Die Ergebnisse zum Experiment D. Zusammengefasst sind die durchschnittliche Laufzeit der Algorithmen in  $s$  und der Registrierungsfehler gemessen am RMSE. Der RMSE wiederum wird durch den Median, den Interquartilabstand, der Bandbreite und den logarithmisch skalierten Boxplot repräsentiert. Die Werte ergeben sich aus 1,000 Registrierungen der eingefärbten Halbkugelsegmente.

Alg.	KNG	IMP	IME	IKP	IKE	IPP	IPE	KCO	KCM	IFP	IFE
Durchschnittliche Laufzeit											
$t$ in $s$	0.80	0.12	0.83	0.44	0.54	0.05	0.04	1.01	0.79	0.62	0.75
RMSE											
Med	23.88	24.70	23.25	24.07	28.23	22.60	24.86	15.17	3.95	7.79	20.43
IQA	4.72	7.03	7.60	8.32	8.16	9.01	9.96	7.05	2.22	5.28	6.32
Bbr	73.67	30.53	379.46	30.61	39.01	34.57	$20^{10}$	54.03	71.52	39.28	37.42



die Korrespondenzen werden im vierdimensionalen Raum berechnet.

Die Ergebnisse zeigen deutlich, dass es bei Punktwolken mit wenig räumlichen Strukturen unbedingt notwendig ist, zusätzliche Merkmale zu berücksichtigen. In diesem konkreten Testszenario schneiden die Algorithmen, die diese Eigenschaft besitzen, am besten ab. Jedoch sind auch deren Genauigkeiten sehr unterschiedlich ausgeprägt. Einerseits sind die **RMSE**-Mediane von **KCM** und **IFP** im Vergleich zu den anderen mit 3.95 und 7.79 Einheiten besonders niedrig. Andererseits zeigen **KCO** und **IFE** mit einem **RMSE**-Median von 15.17 und 20.43 Einheiten den Vorteil der Merkmalerweiterungen an. Die Resultate sind aber nicht in dem Maße besser wie die von **KCM** und **IFP**. Die übrigen Algorithmen, die keine Merkmale berücksichtigen, haben einen **RMSE**-Median zwischen 22 und 29 Einheiten.

Im Verhältnis zum **RMSE**-Median ist der **IQA** aller Algorithmen relativ gering. Das deutet auf eine verhältnismäßig hohe Reproduzierbarkeit hin. Ausgenommen sind die besonders hohen, aber meist seltenen Ausreißer sämtlicher Verfahren. Obwohl die Reproduzierbarkeit der Algorithmen annähernd vergleichbar ist, stechen zwei Verfahren besonders hervor. Zum einen hat der **IPE** viele und vor allem hohe Ausreißer. Zum anderen ist eine hohe Zahl von Ausreißern des **KCM** weitaus niedriger als sein **RMSE**-Median.

Das bedeutet, Algorithmen die Merkmalerweiterungen berücksichtigen, registrieren Punktwolken mit wenig räumlichen Strukturen besser als solche, die das nicht können. Jedoch gibt es bei diesen auffällige Unterschiede. So hat die Merkmalsgewichtung bei der Punkt-zu-Punkt-Korrespondenz des **IFP** sehr viel mehr positiven Einfluss auf die Registrierung als bei der Punkt-zu-Ebene-Variante **IFE**. Grund hierfür ist die Tatsache, dass bei der Generierung von vierdimensionalen Hyperebenen die Merkmale benachbarter Punktepaare zusammengefasst werden. Das führt zu einer Interpolation bzw. Vermischung der beteiligten Merkmale pro Ebene und das wiederum zu weniger deutlich abgrenzbaren Punkt-zu-Ebenen-Korrespondenzen. Vor allem an härteren Farbübergängen oder bei kleineren Merkmalsstrukturen ist eine eindeutige Zuordnung von Punktepaaren beider Wolken oftmals unmöglich.

Der **KCM** ist bezüglich Genauigkeit und Reproduzierbarkeit der Testsieger. Obgleich es hier viele Ausreißer gibt, die sich schlecht auf die Genauigkeit und Reproduzierbarkeit auswirken, liegen die meisten Ausreißer weit unter dem **RMSE** Medianen der anderen Algorithmen. Das hängt offenbar mit den abstoßenden Kräften zusammen, die entstehen, wenn immer Merkmale zweier Punktepaare zu stark voneinander abweichen. Im Vergleich dazu kann **KCO** diese guten Ergebnisse nicht bestätigen, obwohl dieser ebenfalls auf dem Coulombschen Gesetz beruht. Aber hier werden die Kräfte bei ungleichen Punktepaaren lediglich abnehmend skaliert, aber nie in entgegengesetzte Richtung modelliert.

Ähnliches gilt auch für den *IFP*. Bei dieser *ICP*-Variante fließen die Merkmale indirekt als vierte Dimension gewichtend ein. Ähnlich dem *KCO* werden dadurch die Punkt-zu-Punkt-Korrespondenzen beider Wolken skaliert.

## 4.2 Parameteranalyse

Die Machbarkeitsstudie zeigt, dass sich der neu entwickelte Algorithmus nicht nur zur Punktwolkenregistrierung eignet, sondern auch konkurrenzfähig gegenüber gängigen *ICP*-Varianten und -Implementierungen ist. Die Genauigkeit in der Registrierung ist in vielen Fällen sogar höher als bei den *ICP*-Varianten. Das Gleiche gilt auch für die effiziente Registrierung von sehr hoch aufgelösten Punktwolken. Hier ist der neue Algorithmus nicht nur genau, sondern hat zudem eine nahezu konstante Laufzeit. Außerdem verhält sich das neue Verfahren bei verrauschten Punktwolken vergleichbar robust wie die meisten *ICP*-Algorithmen. Sind Punktwolken mit geringer räumlicher Struktur um zusätzliche Merkmale erweitert, bringen die Metriken analog zum Coulombschen Gesetz einen erheblichen Genauigkeitsvorteil.

Die Ergebnisse zeigen weiterhin, dass das neue Verfahren bestimmte Schwachstellen hat. Zum Beispiel ist die Varianz der Registrierungsergebnisse im Vergleich zum *ICP* relativ hoch und vereinzelte Ausreißer sind stark ausgeprägt. Zudem ist die Laufzeit der Registrierung, verglichen mit anderen Algorithmen, bei niedrig aufgelösten Punktwolken zu hoch.

In den folgenden Teilabschnitten werden Möglichkeiten untersucht, welche diese Probleme beheben können. So wird zunächst die Transformationsdarstellung des Algorithmus analysiert. Der *KNG* verwendet homogene Transformationsmatrizen, die zum einen numerisch instabil sind und zum anderen uneindeutige Rotationen beschreiben können. Zudem lassen sich mehrere Bewegungen lediglich durch komplexe Verfahren interpolieren und somit glätten. Die Bewegungsbeschreibung mit Dualen Quaternionen hat diese Einschränkung nicht. Im folgenden Teilabschnitt 4.2.1 wird daher untersucht, ob dieser Fakt für die Registrierung von Punktwolken einen Vorteil bringt.

Die Komplexität des neuen Verfahrens und folglich die Laufzeit hängt wiederum stark von den Stichprobengrößen des Monte-Carlo-Ansatzes ab. Aufgrund dessen wird im zweite Teilabschnitt 4.2.2 differenzierter betrachtet, inwiefern die Wahl der Stichprobengröße die Effizienz des Algorithmus beeinflusst.

Schließlich beeinflussen ebenfalls die frei wählbaren Parameter des Simulated Annealing die Genauigkeit und Geschwindigkeit des Algorithmus. Diese Parameter sind zum einen die Abkühlungskonstante  $c$  und zum anderen der Temperaturschwellwert  $\varepsilon$ .

Die Abkühlungskonstante gibt an, wie schnell das System abkühlt und wie groß die einzelnen Bewegungen sein können. Der Temperaturschwellwert beschreibt den Zeitpunkt, an dem der Algorithmus terminiert. Die Terminierung gibt also an, wie fein die Punktwolkenbewegungen am Ende der Registrierung sein können. Demnach haben beide Parameter eine indirekte Einwirkung auf die Anzahl der Iterationen und beeinflussen die Performanz des Algorithmus. Beide Parameter werden im letzten Teilabschnitt 4.2.3 betrachtet.

Vor allem die letzten beiden Teilabschnitte können als Leitfaden für weitere Untersuchungen des Algorithmus betrachtet werden, denn in beiden Fällen ist die Wahl der Parameter sowohl von den räumlichen Strukturen der Punktwolken und von deren Auflösung abhängig als auch von der spezifischen Anwendung. Daher sollte jedes Anwendungsszenario, bei dem das Verfahren eingesetzt wird, vorab analysiert werden. Erst dann kann der Algorithmus adäquat auf die Anwendung angepasst werden.

#### 4.2.1 Transformationsdarstellungen

Die Ergebnisse der Machbarkeitsstudie zeigen, dass das neue Paradigma zur Punktwolkenregistrierung eine höhere Genauigkeit erzielen kann als aktuelle Standardimplementierungen des ICP. Einzig die Reproduzierbarkeit ist mit einer hohen Varianz gekennzeichnet. Anzumerken ist, dass obwohl die meisten Variationen in den Registrierungsergebnissen sind, einige Ausreißer stark ausgeprägt sind. Im Gegensatz zum ICP ist die neue Methode nicht deterministisch. Das heißt, selbst wenn immer gleiche Voraussetzungen angenommen werden können, wie gleichbleibende Punktzahl, Form und Versatz der Wolken, ist es sehr unwahrscheinlich, dass die Ergebnisse eindeutig reproduziert werden. Dies ist nicht nur negativ zu beurteilen, da so lokale Optima vermieden werden, in denen ein ICP festhängen würde. Gleichwohl ist es unbedingt notwendig die Varianz auf ein Minimum reduzieren. Das gilt vor allem für die starken Ausreißer.

Ein Grund für diese hohe Varianz ist die Transformationsdarstellung durch homogene  $4 \times 4$ -Matrizen. Denn wie bereits in Kapitel 3.3.3 beschrieben, haben die Transformationsmatrizen den Nachteil, dass die Konvertierung zwischen Achse-Winkel-Darstellung und Rotationsanteil der Matrix numerisch instabil ist. Folglich entstehen Transformationen, die nicht mehr ausschließlich rigide Bewegungen abbilden. Das wiederum kann nur mit rechnerisch komplexen Verfahren verhindert werden. Bei der Akkumulation mehrerer fehlerhafter Matrizen kann der Registrierungsfehler zusätzlich stark ansteigen. Außerdem können zu große Kräfte in seltenen Fällen ein unverhältnismäßig starkes Überspringen verursachen, das mit dem Simulated Annealing nicht abgefangen werden kann. Daher ist

es sinnvoll, die Bewegung zu glätten und möglichst schlechte Transformationen bereits im Vorhinein zu filtern.

Numerisch stabiler als Transformationsmatrizen sind die Dualen Quaternionen. Eine Konvertierung zwischen Achse-Winkel-Darstellung und Dualer Quaternion ist mit weniger Rechenaufwand verbunden und verursacht weitaus kleinere Fehler als die Umwandlung in Rotationsmatrizen. Zudem lassen sich mittels Dualer Quaternionen Bewegungen leicht interpolieren und dadurch glätten. Durch die Mittlung einzelner Transformationen und durch das eventuelle Filtern von Ausreißern könnte die Reproduzierbarkeit und möglicherweise die Genauigkeit der Registrierung verbessert werden. Da die Ergebnisse zu Experiment A (Genauigkeit und Reproduzierbarkeit) in Kapitel 4.1.1 zeigen, dass über 99% der Ergebnisse genauer sind als die der anderen Algorithmen sollte bereits eine kleine Anzahl von unterschiedlichen Transformationen pro Iteration ausreichen, um die Varianz der Registrierung erheblich zu senken.

Im Folgenden wird in zwei Experimenten der Einfluss von Transformationsdarstellungen auf die Genauigkeit und Reproduzierbarkeit des Algorithmus untersucht. Das erste Experiment evaluiert die Bewegungsbeschreibung anhand von Matrizen und Dualer Quaternionen analog zum Experiment A des vorherigen Abschnitts 4.1.1. Das zweite evaluiert, inwiefern sich die unterschiedlichen Transformationsdarstellungen auf die Robustheit des Algorithmus auswirken (vgl. Experiment 4.1.3).

**Genauigkeit und Reproduzierbarkeit:** Die Analyse der Genauigkeit und Reproduzierbarkeit hat denselben Versuchsaufbau und Datensatz wie das Experiment A aus dem Teilabschnitt 4.1.1. Die zwei Punktwolken des Drachen-Datensatzes werden in 1.000 Versuchen registriert. Getestet wird zum einen die Bewegungsbeschreibung mit einer einzelnen Dualen Quaternion (DQ1) und zum anderen die Transformation mit Glättung. Dabei wird die Glättung zunächst mit jeweils zwei gemittelten Bewegungen pro Iteration ausgeführt (DQ2). Danach sind jeweils drei bis acht Einzeltransformationen pro Iteration beteiligt, von denen jeweils die schlechteste herausgefiltert und die restlichen erneut gemittelt werden (DQ3 bis DQ8). Die Ergebnisse dieses Experiments sind in Tabelle 4.3 zusammengetragen. Die Transformationsbeschreibungen DQ1 bis DQ8 sind dem bereits getesteten KNG aus der Machbarkeitsstudie gegenübergestellt.

Zunächst fällt auf, dass sich die Laufzeit im Vergleich zum KNG verschlechtert. Das ist zu erwarten, da pro Iteration nicht nur eine mögliche Kraftfeldberechnung durchgeführt wird, sondern bis zu acht. Hinzu kommen die weiteren Berechnungen zur Filterung und Mittlung der Einzelbewegungen. Außerdem benötigt die Transformationsbeschrei-

Tabelle 4.3: Die Ergebnisse zur Wiederholung des Experiments A (vgl. Teilabschnitt 4.1.1) mit unterschiedlichen Transformationsbeschreibungen. Zusammengefasst sind die durchschnittliche Laufzeit der Algorithmen in  $s$  und der Registrierungsfehler gemessen am RMSE. Der RMSE wiederum wird durch den Median, den Interquartilabstand, der Bandbreite und den logarithmisch skalierten Boxplot repräsentiert. Die Werte ergeben sich aus 1,000 Registrierungen zweier Punktwolken des Dragon-Datensatzes.

Alg.	KNG	DQ1	DQ2	DQ3	DQ4	DQ5	DQ6	DQ7	DQ8
Durchschnittliche Laufzeit									
$t$ in $s$	0.6328	0.9742	1.0195	1.0601	1.0939	1.1243	1.1573	1.2979	1.3222
RMSE									
Med	0.0728	0.0596	0.0585	0.0579	0.0578	0.0579	0.0574	0.0578	0.0576
IQA	0.0257	0.0139	0.0116	0.0106	0.0089	0.0083	0.0073	0.0068	0.0066
Bbr	44.2111	66.0228	57.6522	0.0544	0.0436	0.0426	0.0343	0.0299	0.0268

bung von Vektoren, also den einzelnen Punkten in der Wolke, durch Duale Quaternionen mehr Rechenoperationen als die Matrizen-Vektor-Multiplikationen.

Obwohl bis zu acht Kraftfeldberechnungen gleichzeitig ausgeführt werden, ist der Anstieg zwischen den Laufzeiten von DQ1 bis DQ8 relativ gering. Das liegt in der optimierten Implementierung begründet. Nach dem Erfolg der Machbarkeitsstudie wurde das Verfahren in C++ neu implementiert. Das ermöglicht eine performantere Kommunikation zwischen CPU und GPU sowie die Nutzung weiterer CUDA-spezifischer Funktionen, wie zum Beispiel die asynchron überlappenden CUDA Streams [17, 48, 73, 74]. Diese nutzen die Performanz der GPU weit effizienter aus, da versucht wird möglichst sämtliche Ressourcen der Grafikkarte jederzeit optimal zu nutzen und demnach die maximale Anzahl an parallelen Operationen zu gewährleisten. Viele solcher Funktionen können in Matlab derzeit noch nicht verwendet werden.

Weiterhin ist zu erkennen, dass die Transformationsdarstellung mittels Dualen Quaternionen zuverlässigere Ergebnisse liefert als der KNG. Jedoch kann nicht nur die Reproduzierbarkeit des Algorithmus gesteigert werden, sondern ebenfalls die Genauigkeit gegenüber der Matrizendarstellung. Sogar in der Implementierung DQ1, bei der die Bewegungen nicht geglättet werden, ist bereits eine deutliche Reduktion der Varianz erkennbar. Das beweist, dass die Transformationsbeschreibung durch Duale Quaternionen weitaus stabiler ist als die mit Matrizen. Das spiegelt sich bei der Anzahl und Verteilung der Ausreißer wieder. Lediglich 5 von 1,000 Registrierungsergebnissen haben einen RMSE von mehr als 0.1.

Die einfache Glättung des DQ2 bringt lediglich eine geringfügige Verbesserung mit sich. Im Vergleich zu DQ1 verringern sich RMSE-Median, IQA und Bandbreite geringfügig, doch die wenigen, aber großen Ausreißer bleiben erhalten. Da in der Implementierung DQ2 lediglich zwei Bewegungen pro Iteration gemittelt werden, scheinen die schlechten Transformationen noch immer stark ins Gewicht zu fallen und somit die Registrierungsergebnisse signifikant zu beeinflussen.

Eine erhebliche Verbesserung der Reproduzierbarkeit erfolgt erst durch das zusätzliche Filtern. Wenn die jeweils schlechteste Transformation pro Iterationsschritt eliminiert und die verbleibenden Bewegungen gemittelt sind, dann werden auch die starken Ausreißer beseitigt. Ab der Variante DQ3 liegen sämtliche Abweichungen vom RMSE-Median unter 0.1 Einheiten. Verglichen mit der Tabelle 4.1 aus dem Experiment A 4.1.1, bei dem die RMSE-Mediane sämtlicher ICP-Varianten über 0.2 liegen, ist das eine eindeutige Verbesserung.

Die Unterschiede in der Genauigkeit und Reproduzierbarkeit der Varianten DQ4 bis DQ8 sind kontinuierlich aber nicht mehr auffällig. Da die Laufzeit ebenfalls permanent

zunimmt, lässt sich daraus ableiten, dass für zukünftige Verwendungen des Algorithmus die Variante **DQ4** bevorzugt werden sollte. Schließlich zeigen die Ergebnisse, dass **DQ4** das beste Verhältnis zwischen Laufzeit und Genauigkeit sowie Reproduzierbarkeit garantiert.

**Robustheit** Durch die Transformationsbeschreibung Duale Quaternionen sowie der zusätzlichen Filterung und Glättung kann sowohl die Genauigkeit als auch die Reproduzierbarkeit der Registrierung erheblich gesteigert werden. Mit diesen Erkenntnissen stellt sich die Frage, ob dies ebenfalls auf die Robustheit des Algorithmus übertragbar ist.

Um das herauszufinden, wird das Experiment aus Abschnitt 4.1.3 wiederholt. In 30 Durchläufen werden auf dem Template der Utah-Teekanne normalverteilte Rauschlevel simuliert. Die Varianzen der Normalverteilungen sind diskret von  $\sigma^2 = 0.0$  bis  $\sigma^2 = 2.9$ , mit einem Abstand von 0.1, verteilt. In jedem Durchgang müssen die Implementierungen **DQ1** bis **DQ8** je 100 Registrierungen durchführen. Die 100 simulierten Transformationen zwischen Modell und Template-Kopie sind dieselben wie in Experiment 4.1.3. Die jeweiligen **RMSE**-Mediane sind in Abbildung 4.8 aufgetragen. Für den direkten Vergleich sind zusätzlich die Ergebnisse des **KNG** aus Experiment 4.1.3 beigefügt.

Es ist deutlich zu erkennen, dass die Transformationsdarstellung mit Dualen Quaternionen keinen Genauigkeitsvorteil gegenüber homogenen Matrizen mit sich bringt. Das gilt weder für die einfache Transformationsbeschreibung durch Duale Quaternionen (**DQ1**) noch für die einfache Mittlung zweier Bewegungen (**DQ2**) sowie für die zusätzliche Filterung (**DQ3-DQ8**). Auffällig ist, dass die Varianten **DQ1** bis **DQ8** wiederholbarer registrieren als der **KNG**, wobei **DQ1** und **DQ2** geringfügig schlechter abschneiden als die restlichen Implementierungen. Das heißt, auch in Anwendungen, bei denen verrauschte Punktwolken zu erwarten sind, sollte die Variante **DQ4** den anderen Transformationsbeschreibungen vorgezogen werden. Diese hat im Vergleich zur Laufzeit das beste Verhältnis zwischen Genauigkeit, Reproduzierbarkeit und Robustheit.

#### 4.2.2 Komplexität

Sämtliche vorhergegangene Experimente zeigen, dass die neue Kraftfeldmethode ein verlässliches Verfahren zur Punktwolkenregistrierung ist. Bei den meisten Versuchen zeigt die Kraftfeldmethode überdies eine höhere Genauigkeit als gängige **ICP**-Implementierungen. Durch die Erweiterung einer Transformationsbeschreibung, die auf Dualen Quaternionen basiert, ist es möglich mehrere Kraftfelder pro Iteration zu berechnen. Von

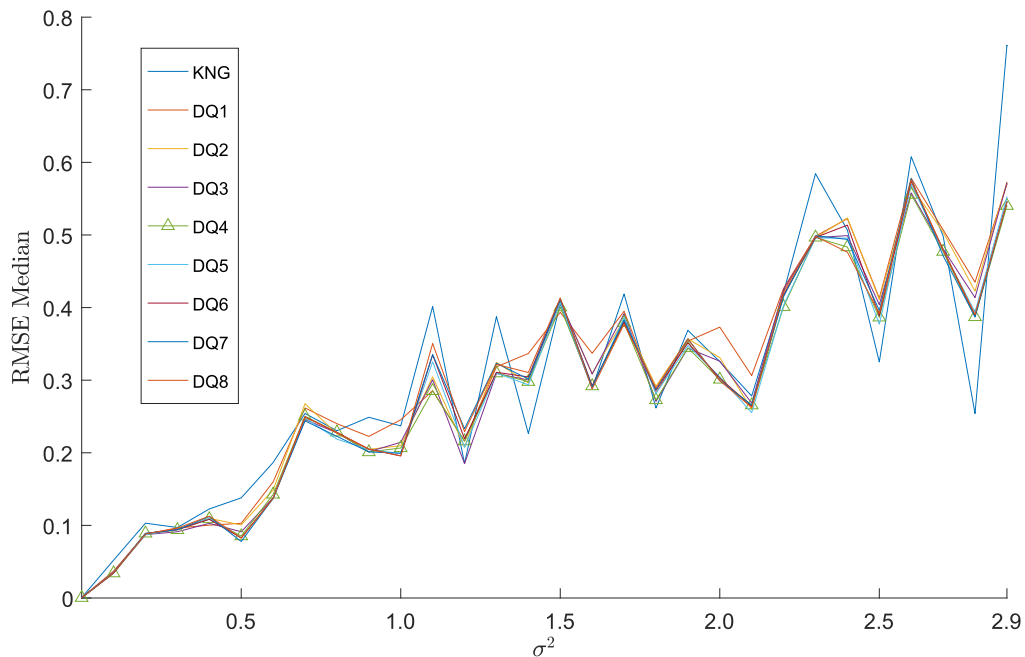


Abbildung 4.8: Die Ergebnisse zur Wiederholung des Experiments C (vgl. Teilabschnitt 4.1.3) mit unterschiedlichen Transformationsbeschreibungen. Der Registrierungsfehler repräsentiert durch den RMSE-Median über 100 Registrierungen ist dem Rauschlevel  $\sigma^2$  gegenübergestellt.

denen wird jeweils die schlechteste Bewegung gefiltert und die verbleibenden Transformationen zu einer gemittelt. Das führt zu einer erkennbaren Reproduzierbarkeit des Algorithmus, sodass die Methode konkurrenzfähig zum Goldstandard ist.

Die Experimente zeigen weiterhin, dass die Laufzeit des neuen Algorithmus annähernd konstant ist und vor allem bei hoch aufgelösten Punktwolken überzeugt. Hinsichtlich niedriger Punktauflösungen stellt die konstante Berechnungszeit ein Problem dar, denn der Algorithmus ist entscheidend langsamer als andere Registrierungsalgorithmen. Der **KNG** benötigt, unabhängig von der Punktwolkengröße, um die 0.6 s. Die Varianten mit geglätteter Bewegung benötigen überdies knapp über 1 s. Verglichen mit den **ICP**-Implementierungen im Effizienzexperiment 4.1.2 ist diese Laufzeit bei Auflösungen unter 30,000 Punkten verbesserungswürdig.

Schlussfolgernd lässt sich sagen, dass die Laufzeit des Verfahrens deutlich verringert werden muss. Hierfür sieht das neue Konzept die Anpassung der Stichprobengröße im Monte-Carlo-Ansatz vor. Wie in Kapitel 3.3.2 beschrieben hängt die Komplexität des Algorithmus hauptsächlich von den Stichproben  $\mathcal{M}$  und  $\mathcal{N}$  ab. Werden beide verringert, sinkt die Laufzeit des Algorithmus. Dabei ist zu beachten, dass eine solche Verringerung

der Stichprobengröße zu einer Verschlechterung der Registrierungsgenauigkeit führen kann.

Um das Verhalten des Algorithmus bei der Komplexitätsreduktion und somit die Abhängigkeiten von Stichprobengröße, Genauigkeit und Komplexität zu untersuchen, wird im Folgenden ein weiteres Experiment auf Grundlage des Drachen-Datensatzes durchgeführt (vgl. Experiment 4.1.1). Dabei werden sowohl  $\mathcal{M}$  als auch  $\mathcal{N}$  kontinuierlich erhöht und für jedes Wertepaar 100 Registrierungen durchgeführt. Als Grundlage dient die DQ4-Implementierung, die sich zuvor als adäquater Kompromiss aus Genauigkeit, Reproduzierbarkeit und Laufzeit erwiesen hat.  $\mathcal{M}$  und  $\mathcal{N}$  beginnen jeweils mit einer Größe von einem Punkt, gefolgt von 10 Punkten. Anschließend werden  $\mathcal{M}$  und  $\mathcal{N}$  solange um jeweils 10 Punkte erweitert, bis beide Stichproben 1,020 Punkte groß sind.

Die Ergebnisse dieses Experiments sind in Abbildung 4.9 zusammengetragen. Die Grafik kombiniert alle vier Abhängigkeiten. Auf zwei Achsen sind jeweils die Stichprobengrößen  $\mathcal{M}$  und  $\mathcal{N}$  aufgetragen. Die dritte Achse zeigt den Registrierungsfehler in Abhängigkeit von  $\mathcal{M}$  und  $\mathcal{N}$  angegeben durch den RMSE-Median. Aus Gründen der Übersichtlichkeit ist diese Achse logarithmisch skaliert. Infolgedessen setzen sich die diskreten RMSE-Mediane zu dem abgebildeten Gitternetz zusammen, wobei jedes Flächenelement des Gitters einen Datenpunkt repräsentiert. Ein RMSE-Median unter dem Wert von 0.07 Einheiten kann bei diesem Datensatz als Registrierungsergebnis mit sehr hoher Genauigkeit beurteilt werden (vgl. Experiment 4.1.1). Daher sind für sämtliche Datenpunkte bei denen der RMSE-Median kleiner als 0.07 Einheiten ist die entsprechenden Flächenelemente ausgefüllt dargestellt. Leere Flächenelemente symbolisieren Datenpunkte, deren RMSE-Medien größer oder gleich dem Grenzwert von 0.7 Einheiten sind. Der Farbwert, mit dem das Gitternetz gefärbt ist, ergibt sich aus den gemittelten Laufzeiten sämtlicher 100 Registrierungen, abhängig von  $\mathcal{M}$  und  $\mathcal{N}$ . Die dunkelblaue Farbe repräsentiert Laufzeiten unter 0.3 s. Rote bis dunkelrote Farben stehen für Berechnungszeiten über 1 s.

Zunächst zeigen die Ergebnisse, dass es eine direkte Abhängigkeit zwischen den Stichprobengrößen und der Genauigkeit sowie der Laufzeit gibt. Bezüglich der Genauigkeit ist der Einfluss von  $\mathcal{M}$  und  $\mathcal{N}$  jedoch weniger stark ausgeprägt. Es ist offensichtlich, dass die Stichproben nicht nur aus einem einzigen Punkt bestehen sollten. Allerdings sind bereits ab Werten von  $\mathcal{M} \geq 50$  sowie  $\mathcal{N} \geq 50$  die meisten RMSE-Mediane kleiner als 0.2 Einheiten. Ab diesen Werten sind die Registrierungsergebnisse mit denen der ICP-Implementierungen vergleichbar (vgl. Experiment 4.1.1). Die Laufzeiten liegen in diesem konkreten Fall bei 0.26 s. Auch dieser Wert ist mit den meisten ICP-Varianten

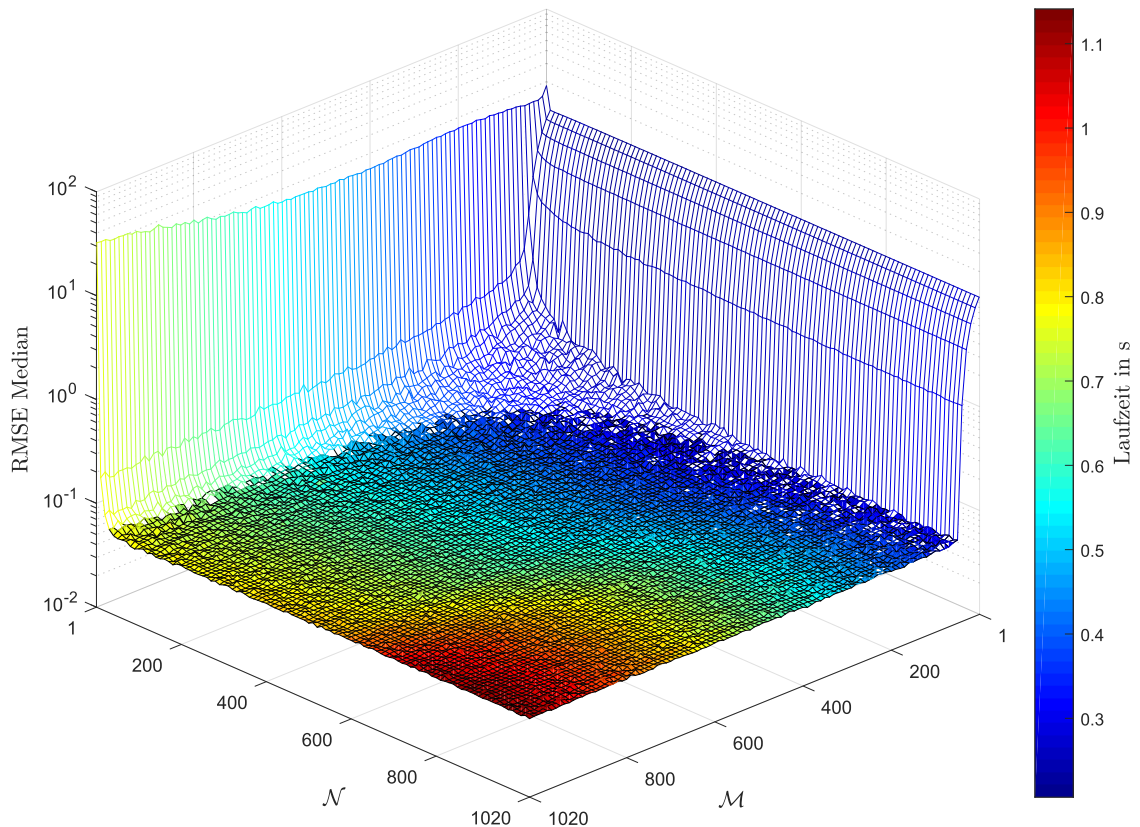


Abbildung 4.9: Die Ergebnisse zur Komplexitätsuntersuchung: analog zu Experiment 4.1.1 werden zwei Punktwolken des Dragon-Datensatzes für jeweils unterschiedliche  $\mathcal{M}$  und  $\mathcal{N}$  100-mal registriert. Gegenübergestellt sind der logarithmische Registrierungsfehler RMSE-Median und die durchschnittliche Laufzeit in  $s$ . Ausgefüllte Flächenelemente repräsentieren einen RMSE-Median von unter 0.07 Einheiten.

konkurrenzfähig.

Für höhere Genauigkeiten bei der Registrierung müssen andere Werte für  $\mathcal{M}$  und  $\mathcal{N}$  gewählt werden. Diese sind eindeutig aus der Abbildung 4.9 zu ermitteln. Der RMSE-Median hängt annähernd symmetrisch von der Größe beider Stichproben ab. Das gilt nicht für die Laufzeit, denn diese hängt stärker von  $\mathcal{M}$  als von  $\mathcal{N}$  ab. Das wiederum liegt an der Mehrkernimplementierung und Ausführung des Algorithmus auf der GPU. Es werden  $\mathcal{N}$  Kraftfeldberechnungen auf ebenso vielen parallel arbeitenden Recheneinheiten der Grafikkarte gestartet. Diese wiederum berechnen jeweils  $\mathcal{M}$  viele Einzelkräfte. Wie bereits in Kapitel 3.3.2 beschrieben, sind die Recheneinheiten der GPU darauf ausgelegt, kleine Berechnungen durchzuführen, von denen viele parallel sein sollen. Da mit steigenden  $\mathcal{M}$  die Komplexität der Berechnungen pro Recheneinheit stark zunimmt, kann dies

nicht mehr durch hohe Parallelität ausgeglichen werden. Die GPU muss warten, bis alle Recheneinheiten fertig sind. Ist  $\mathcal{M}$  jedoch klein, benötigen die Einheiten weniger Zeit für die Kraftberechnungen. Dann kann eine größere Anzahl an  $\mathcal{N}$  parallelen Berechnungen hinzugezogen werden, ohne die Laufzeit zu verschlechtern.

Beim Blick auf die Grafik 4.9 fällt auf, dass mehrere Bereiche gewählt werden könnten, in denen das Verhältnis von  $\mathcal{M}$  und  $\mathcal{N}$  zur Genauigkeit sowie Laufzeit ausgewogen ist. Ausreichend genau wäre beispielsweise ein RMSE-Median unter 0.07 Einheiten. Außerdem sollte nach Möglichkeit  $\mathcal{N}$  größer sein als  $\mathcal{M}$ , um die Laufzeit gering zu halten. In diesem Fall wäre zum Beispiel eine Kombination aus  $\mathcal{M} = 100$  und  $\mathcal{N} = 600$  ein guter Kompromiss aus Genauigkeit und Laufzeit, bei dem der RMSE-Median bei 0.06 Einheiten und die Laufzeit bei 0.3 s liegt. Verglichen mit den meisten ICP-Implementierungen im Test ist das vergleichbar bis besser (vgl. Experiment 4.1.1).

Anzumerken ist, dass diese Werte zu einem gewissen Grad von der Punktwolkengröße abhängen. Das heißt, vor späteren Verwendungen des Algorithmus sollte unbedingt ein ähnlicher Test wiederholt und die Wahl der Stichproben auf das spezifische Anwendungsproblem angepasst werden.

### 4.2.3 Regularisierung

Neben der Glättung von Punktwolken Transformationen und der Reduktion der Laufzeit muss noch ein letzter Bestandteil des neuen Verfahrens untersucht werden: die Regularisierung von Punktwolkenbewegungen durch Simulated Annealing (vgl. Kapitel 3.3.1). Der Prozess der simulierten Abkühlung hat drei Aufgaben. Erstens werden die Punktwolkenbewegungen pro Iteration zunehmend abgeschwächt, denn die Kraftfelder erzeugen während der Registrierung Schrittweiten, die direkt mit der Qualität der Überlagerung zunehmen oder zumindest konstant bleiben. Das heißt, die Abkühlung bewirkt, dass diese Schrittweiten kleiner werden können und der Algorithmus sich zunehmend dem Optimum nähert. Zweitens verhindert das Simulated Annealing, dass bei zu großen Kraftanstiegen zwischen aufeinanderfolgenden Iterationen unverhältnismäßig hohe Oszillationen um das Optimum entstehen. Drittens lässt das Simulated Annealing dennoch einige der Überschwinger zu. Dementsprechend wird verhindert, dass sich das Optimierungsverfahren in lokalen Optima festsetzt. Das steht im Gegensatz zu deterministischen Verfahren wie dem ICP, bei denen unter denselben Grundvoraussetzungen stets identische Ergebnisse zu erwarten sind.

Wie groß der Einfluss des Simulated Annealings auf die Registrierung ist, soll in diesem Abschnitt untersucht werden. Dabei ist vor allem interessant, wie stark die Genau-

igkeit der Ergebnisse von den frei wählbaren Parametern Abkühlungskonstante  $c$  und Temperaturschwellwert  $\varepsilon$  abhängt. Diese Analyse gibt Aufschluss darüber, wie die Parameter schlussendlich gewählt werden sollten. Doch zunächst wird der Einfluss der zufälligen Schrittweitenanpassungen bei zu hohen Energieübergängen untersucht. Denn möglicherweise ist es an dieser Stelle empfehlenswert, konstant abnehmende Schrittweiten zu verwenden statt diese aus den Kräftefeldern zu berechnen.

**Optimierungsrichtung und Schrittweiten** Zunächst soll geprüft werden, inwiefern die hohe Genauigkeit des Algorithmus von den Schrittweiten der Kraftfelder und den Übergangswahrscheinlichkeiten des Simulated Annealings abhängen. Hierfür wird erneut der Drachen-Datensatz aus dem Experiment 4.1.1 verwendet. Ausgehend von den zuvor gewonnenen Erkenntnissen wird die DQ4-Implementierung getestet. Das heißt, die Transformationsbeschreibung geschieht mittels Dualen Quaternionen, wobei in jeder Iteration aus vier verschiedenen Bewegungen die jeweils schlechteste herausgefiltert wird und die restlichen drei zu einer Gesamttransformation geglättet werden. Diese zeigte die höchste Genauigkeit und Reproduzierbarkeit im Verhältnis zur geringen Laufzeit. Die Stichprobengrößen orientieren sich an den Ergebnissen der Komplexitätsanalyse, mit  $\mathcal{M} = 100$  sowie  $\mathcal{N} = 600$ .

In diesem Experiment wird der DQ4-Algorithmus zweimal ausgeführt. Zum einen wird DQ4 für 1,000 Versuche bei standardmäßiger simulierter Abkühlung ausgeführt inklusive Schrittweitenmodellierung durch Kraftfelder und zufälliger Zustandsübergänge des Simulated Annealings (DQ4<sub>M</sub>). Zum anderen wird die Schrittweite zu Beginn der Registrierung auf 1 gesetzt (DQ4<sub>O</sub>). Damit ist diese weder von der Entfernung beider Punktwolken zueinander noch von den Zufallsfaktoren bei hohen Energieübergängen abhängig. Lediglich die Temperaturabnahme sorgt für eine kontinuierliche Reduktion der Schrittweite. Das gilt sowohl für den Winkel der Rotation als auch für die Länge des Translationsvektors.

Auch DQ4<sub>O</sub> liefert präzise Registrierungsergebnisse. Das zeigen die Testresultate in Tabelle 4.4. Obgleich der RMSE-Median deutlich höher ist als der des DQ4<sub>M</sub>, ist dieser Wert noch immer geringer als die der getesteten ICP-Varianten aus Experiment 4.1.1. Nur 4 von 1,000 Ausreißern sind größer als 0.21 Einheiten und überschreiten folglich den Fehler der besten ICP-Implementierungen. Infolgedessen iteriert das Verfahren unabhängig von Punktwolkenentfernungen und zufälligen Bewegungsänderungen in Richtung der besten Überlagerung. Das neue Konzept ist somit nicht zwingend von den zufälligen Energietransitionen des Simulated Annealings abhängig, was wiederum bedeutet, dass auch andere Regularisierungsverfahren für den Gradientenabstieg verwendet wer-

Tabelle 4.4: Die Ergebnisse zur Evaluierung der Schrittweitenmodellierung analog zu Experiment 4.1.1. Getestet wurden  $DQ4_M$ , der Schrittweiten aus dem Simulated Annealing und Kraftfeldberechnungen berechnet, und  $DQ4_O$ , der konstant abnehmende Schrittweiten verwendet. Verglichen wurden beide bezüglich des Registrierungsfehlers, repräsentiert durch RMSE-Median, IQA, Bandbreite und Boxplot.

Alg.	RMSE			RMSE-Boxplot				
	Med	IQA	Bbr	0.05	0.1	0.15	0.2	0.25
$DQ4_M$	0.0612	0.0188	0.0916					
$DQ4_O$	0.0952	0.0420	0.2104					

den könnten.

Darüber hinaus ist zu erkennen, dass  $DQ4_M$  eine weitaus höhere Genauigkeit und Reproduzierbarkeit erreicht. Ein Grund dafür könnten die möglicherweise zu kurzen Schrittweiten sein, die unabhängig von der eigentlichen Punktwolkendistanz verwendet werden. Eine weitere Ursache liegt in den lokalen Optima, auf die der Algorithmus hin iteriert, wenn keine Zustandsübergänge erlaubt sind und deshalb auch keine unverhältnismäßig hohen Schrittweiten wahrscheinlich sind. Auch wenn es bedeutet, dass es zu Oszillationen um das Optimum kommt und sich folglich die Reproduzierbarkeit des Algorithmus leicht verschlechtert, ist das Simulated Annealing ein adäquater Regularisierer für das neue Registrierungskonzept.

**Parameteranalyse:** Nachdem gezeigt wurde, dass sowohl die Glättung der Bewegung mittels Dualen Quaternionen als auch die Regularisierung der Punktwolkenbewegung zulässige Verfahren für die Punktwolkenregistrierung sind, müssen die frei wählbaren Parameter des Simulated Annealings geprüft werden. Das sind zum einen die Abkühlungskonstante  $c$  und zum anderen das Abbruchkriterium  $\varepsilon$ . Beide werden im Folgenden in Abhängigkeit von der Registrierungsgenauigkeit untersucht. Hierfür dient wieder der Drachen-Datensatz aus Experiment 4.1.1. Getestet wird anhand der  $DQ4$ -Implementierung mit den Stichprobengrößen  $\mathcal{M} = 100$  und  $\mathcal{N} = 600$ .

Für das Abbruchkriterium des Registrierungsalgorithmus werden in diesem Experiment sechs verschiedene Werte  $\varepsilon = \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$  analysiert. Die Abkühlungskonstante ist in diskrete Werte zwischen  $c = 0.750$  und  $c = 0.999$  unterteilt mit einer Schrittweite von 0.001. Aus Komplexitätsgründen, und da sich der  $DQ4$  als sehr genau und reproduzierbar erwiesen hat, wird für jedes Parameterpaar  $c$  und  $\varepsilon$  nur eine

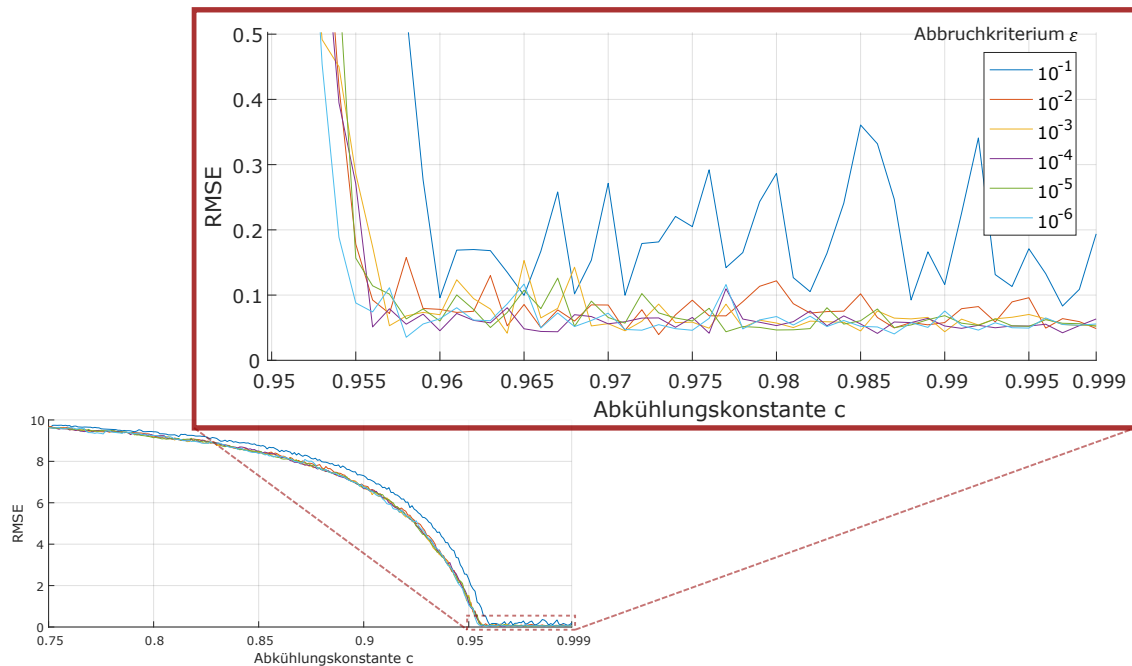


Abbildung 4.10: Ergebnisse zur Analyse des Registrierungsfehlers RMSE in Abhängigkeit der frei wählbaren Parameter Abkühlungskonstante  $c$  und Abbruchkriterium  $\varepsilon$ . Oben rechts und rot eingerahmt ist der Ausschnitt mit dem höchsten Informationsgehalt vergrößert dargestellt.

einzigste Registrierung berechnet. Die Registrierungsfehler und Laufzeiten der einzelnen Durchläufe sind in Abbildungen 4.10 und 4.11 dargestellt.

In Abbildung 4.10 ist unten links der RMSE in Abhängigkeit der beiden Parameter über alle Werte der Abkühlungskonstante aufgetragen. Da bei einem kleinen  $c$  zu hohe Registrierungsfehler auftreten, ist der interessante Bereich für Werte  $c > 0,95$  vergrößert dargestellt (rot eingerahmter Kasten oben rechts). Für  $c \geq 0,955$  fällt der RMSE sehr stark ab. Danach konvergieren die Registrierungsfehler. Ab einer Abkühlungskonstante von  $c = 0,98$  sind die Registrierungsergebnisse gleichbleibend gut. Außerdem ist erkennbar, dass ein Temperaturschwellwert von  $\varepsilon \leq 10^{-4}$  gewählt werden sollte. Dieser konvergiert am schnellsten gegen hohe Genauigkeiten und hat eine sehr kleine Varianz. Größere Werte von  $\varepsilon = 10^{-1}$  sind hingegen zu ungenau, konvergieren zu langsam bzw. variieren zu stark. Für  $\varepsilon < 10^{-4}$  sind dagegen kaum Unterschiede auszumachen.

Wird das Wertepaar  $\varepsilon = 10^{-4}$  und  $c = 0,98$  mit den Laufzeiten in Abbildung 4.11 abgeglichen, zeigt sich, dass auch die Berechnungszeiten in einem akzeptablen Bereich

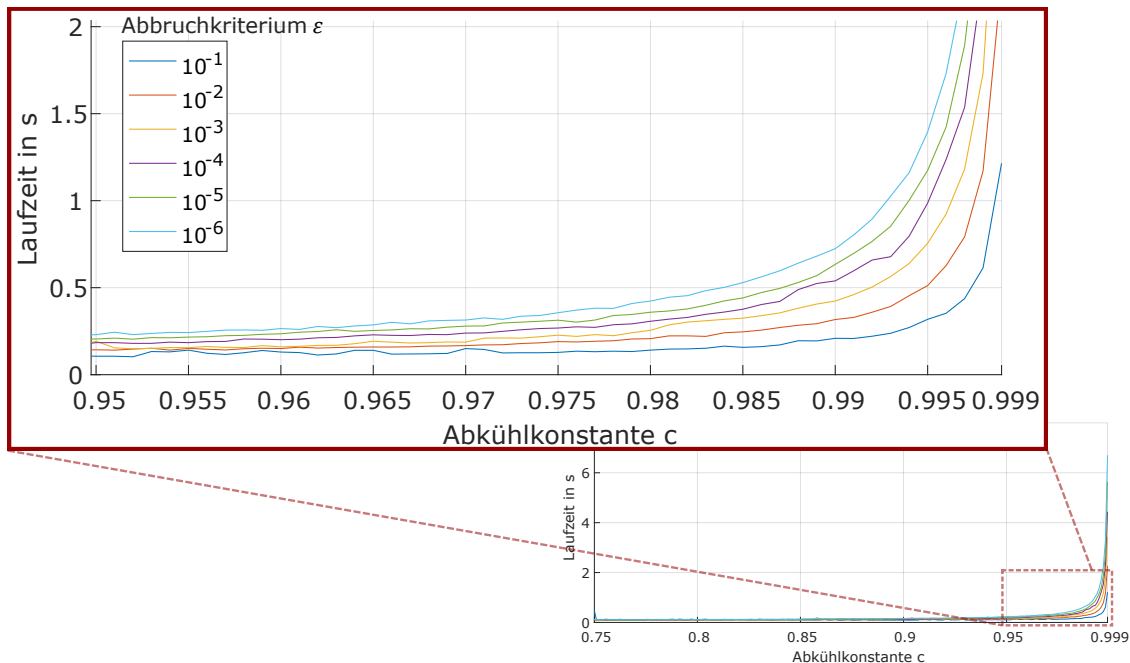


Abbildung 4.11: Ergebnisse zur Analyse der Laufzeit in Abhängigkeit der frei wählbaren Parameter Abkühlkonstante  $c$  und Abbruchkriterium  $\varepsilon$ . Oben links und rot eingerahmt ist der Ausschnitt mit dem höchsten Informationsgehalt vergrößert dargestellt.

von um die 0.3 s liegen. Höhere Werte für  $c$  bzw. kleinere  $\varepsilon$  sind nach der Abbildung nicht empfehlenswert, da deren Laufzeiten exponentiell ansteigen. Kleine  $c$  könnten ebenfalls gewählt werden, was wiederum Auswirkungen auf eine bessere Laufzeit hätte. Doch sollten diese in einem adäquaten Verhältnis zur Genauigkeit und Reproduzierbarkeit stehen.

Schlussendlich lassen sich die Parameter identifizieren, die für die Simulierte Abkühlung besonders geeignet sind. Der Schwellwert  $\varepsilon = 10^{-4}$  ist in diesem Anwendungsbeispiel besonders eindeutig. Größere Werte führen zu höheren Ungenauigkeiten. Kleinere Werte indessen haben nur wenig Auswirkung auf die Genauigkeit, wirken sich aber stark negativ auf die Laufzeit des Algorithmus aus. Bei der Abkühlkonstante sind die Grenzen ab  $c > 0.955$  weniger stark ausgeprägt. Hier sind Werte um  $c \approx 0.98$  zu wählen. Denn sowohl die Genauigkeit als auch die Reproduzierbarkeit sind in diesem Bereich sehr hoch und ändern sich kaum.

Diese beiden Parameter hängen jedoch stark von der jeweiligen Anwendung und von den zu registrierenden Punktwolken ab. So beeinflusst die räumliche Ausdehnung der Wolken den Schwellwert  $\varepsilon$ . Für kleine Punktwolken sollte demnach ein kleiner Wert  $\varepsilon$  angenommen werden. Die Wahl der Abkühlkonstante ist vom Abstand der Punkt-

wolken zueinander abhängig. Ist der zu erwartende Abstand zwischen zwei zu registrierenden Punktwolken groß, muss ebenfalls  $c$  entsprechend größer gewählt werden. Damit steigt die Laufzeit, aber der Algorithmus verwendet mehr Iterationen, in denen potentiell größere Transformationen stattfinden.  $c$  beeinflusst nicht nur die Anzahl der Iterationen und somit die Laufzeit, sondern auch die Größe der Schrittweitenreduktion.

Genauso wie es für andere Registrierungsalgorithmen üblich ist, sollte für das neue Verfahren immer dann eine Parameteranalyse durchgeführt werden, wenn es zum Wechsel der spezifischen Anwendung kommt. Je nach Anwendungsszenario bzw. -anforderung kann das entsprechende Verhältnis zwischen Laufzeit, Genauigkeit und Reproduzierbarkeit ermittelt werden. Die zu wählenden Parameter lassen sich bei dieser Untersuchung eindeutig ablesen und ermöglichen eine maximal hohe Effizienz des Algorithmus.



## 5 Fazit

Die Motivation dieser Dissertation umfasst zwei Fragestellungen. Erstens gilt es herauszufinden, ob es möglich ist, das Problem der Punktwolkenregistrierung in einer Art und Weise zu formulieren, dass es mit Hilfe von physikalischen Gesetzen gelöst werden kann.

Hierzu wurde im Rahmen dieser Arbeit ein neues Konzept entwickelt, das Punktwolken analog zu Modellen der klassischen Kinematik beschreibt. Inspiriert von den Mehrteilchensystemen der Physik wirken Kräfte auf sämtliche Elemente einer Punktwolke, die wiederum in ihrem Massenschwerpunkt zusammengefasst werden. Die gebündelte Krafteinwirkung hat eine Bewegungsänderung des Gesamtsystems Punktwolke zur Folge. Da die Registrierung rigide ist, leitet sich die Punktwolkenbewegung aus dem kinematischen Modell der Starrkörpertransformation ab. Folglich kann die zu registrierende Punktwolke (Template) in Richtung des statischen Modells bewegt werden. Dieser Prozess ist iterativ, sodass sich das Template schrittweise auf das Modell zubewegt, bis eine möglichst optimale Überlagerung stattfindet.

Die Bewegungsrichtung des Templates ergibt sich aus Kraftfeldern, die zwischen den beiden Punktwolken berechnet werden und auf den Massenschwerpunkt des Templates einwirken. Indessen repräsentieren Kraftfelder die Metrik des neuen Registrierungskonzeptes. Die Modellierung der Kräfte kann frei gewählt werden und im Gegensatz zu den drei Beispielen, die in dieser Arbeit beschrieben werden, müssen sich die Metriken nicht zwangsläufig von der Physik ableiten. Von den drei vorgeschlagenen Metriken ist eine dem Newtonschen Gravitationsgesetz nachempfunden und beschreibt die Kräfte anhand der räumlichen Punktverteilung beider Wolken. Je geringer der Abstand zwischen Punktpaaren ist, umso stärker wirken die Anziehungskräfte. Demnach ist diese Metrik eine gedämpfte Version des ICP-Verfahrens. Da der ICP ausschließlich die jeweils am nächsten benachbarten Punktpaare betrachtet, führt das zu einer direkten eins-zu-eins Punktkorrespondenz zwischen den Wolken. Mit Kraftfeldern als Abstandsmaß wird hingegen ein Punkt des Templates von mehreren Modell-Punkten gleichzeitig beeinflusst.

Die anderen beiden Beispielmetriken sind vom Coulombschen Gesetz inspiriert. Punktwolken können um zusätzliche Merkmale erweitert werden, wie zum Beispiel Helligkeitswerte oder Farben. Analog zur Elektrostatik können diese Merkmale als Ladungen in-

terpretiert werden. Aus diesem Grund sind die Kräfte zwischen zwei Punkten nicht nur von deren Abstand abhängig, sondern darüber hinaus von deren Merkmalen. Die erste Metrik skaliert die Anziehungskräfte entsprechend den Merkmalsunterschieden. Wenn die Merkmale korrespondierender Punktpaare annähernd gleich sind, dann wird der Abstand zwischen den Punkten stark gewichtet. Bei unterschiedlichen bis ungleichen Merkmalen tendiert der Kraftbetrag gegen Null. Die zweite Metrik wiederum skaliert nicht nur den Betrag der Kraft, sondern modelliert zusätzlich abstoßende Kräfte. Abstoßungen treten immer dann auf, wenn die Merkmale entsprechender Punktpaare wenig bis gar nicht zueinander passen.

Dieses neue Konzept fordert, dass die Kraftbeträge mit zunehmender Qualität der Registrierung wachsen oder zumindest konstant bleiben. Daher müssen die Schrittweiten der Punktwolkenbewegungen entsprechend angepasst werden. Andernfalls kommt es zu nie endenden Oszillationen um das Optimum bis hin zu Punktwolken, die sich zunehmend voneinander entfernen. Um dieses Verhalten zu regularisieren, wird ein Simulated Annealing Verfahren angewandt. Wie die Methoden zuvor leitet sich auch diese Methode von physikalischen Realweltphänomenen ab und beschreibt den Optimierungsprozess als thermodynamisches System. Es skaliert die Punktwolkenbewegung über eine stetig abnehmende Temperatur. Daher können trotz steigender Kraftbeträge die Schrittweiten solange abnehmen, bis der iterative Algorithmus an einem bestimmten Temperaturschwellwert terminiert. Zusätzlich akzeptiert das Simulated Annealing bestimmte Zustandsübergänge in Abhängigkeit der aktuellen Systemtemperatur und Energieunterschiede. Die Wahrscheinlichkeit des Zustandsübergangs ergibt sich aus der Boltzmann-Gibbs-Verteilung. Zu Beginn der Registrierung und bei hohen Temperaturen sind große Sprünge in der Punktwolkenbewegung sehr wahrscheinlich. Zum Ende hin nehmen die Temperaturen und folglich auch die Schrittweiten ab. Größere Ausreißer in den Bewegungsbeträgen werden mit hoher Wahrscheinlichkeit unterdrückt, können aber auftreten. Obgleich feine Schrittweiten am Ende der Registrierung notwendig sind, ermöglicht dieses Vorgehen weiterhin eine gewisse Anzahl an größeren Bewegungen, um die Registrierung in lokale Optima zu vermeiden.

Die hohe Laufzeitkomplexität des Algorithmus wird durch parallel laufende Kraftfeldberechnungen auf Mehrkernrechnerarchitekturen gelöst. Das ist möglich, da die Kraftfelder, welche die einzelnen Template-Punkte beeinflussen, vollständig unabhängig voneinander berechnet werden. Lediglich die resultierende Kraft muss mittels des Superpositionsprinzips im Massenschwerpunkt der Wolke zusammengefasst werden. Zudem beruht die Kraftfeldberechnung auf dem Monte-Carlo-Prinzip. Anstatt die Kraftfelder für sämtliche Punkte beider Wolken zu berechnen, werden lediglich Stichproben betrachtet. Das ver-

schaft dem Algorithmus nicht nur eine annähernd konstante Laufzeitkomplexität, sondern ermöglicht zusätzlich den effizienten Einsatz von GPUs.

Schließlich verwendet der Algorithmus eine für Registrierungen unübliche Transformationsbeschreibung. Unter Zuhilfenahme von Dualen Quaternionen ist es möglich, mehrere Punktwolkenbewegungen in einer Iteration auszuführen und anschließend eine geglättete Bewegung zu berechnen. Dazu werden sämtliche Rotationen einer Iteration gemittelt und die Transformation, deren Rotation am weitesten vom Mittel abweicht, eliminiert. Die übrigen Bewegungen werden erneut zu einer resultierenden Gesamttransformation gemittelt. Es zeigt sich, dass dieses Verfahren die Fehlervarianz erheblich senkt. Vor allem die wenigen, aber extrem hohen Ausreißer in der Registrierungsgenauigkeit werden durch die Glättung gefiltert. Zudem sind Duale Quaternionen numerisch stabiler als homogene Transformationsmatrizen, was sich ebenfalls positiv auf die Reproduzierbarkeit der Registrierung auswirkt. Zwar steigt der Rechenaufwand durch den Glättungsprozess, doch ist dieser Laufzeitanstieg nicht wesentlich.

Die umfangreiche Evaluierung des Algorithmus zeigt, dass das Zusammenspiel sämtlicher Komponenten ein funktionsfähiges Registrierungsverfahren ergibt. Der Algorithmus registriert Punktwolken unterschiedlicher Form und Auflösungen effizient. Die Genauigkeit der Registrierungsergebnisse ist reproduzierbar hoch. Laufzeitkomplexität und Punktwolkenauflösung sind unabhängig voneinander, sodass der Algorithmus stets mit einer annähernd konstanten Rechenzeit registriert. Sogar die Registrierung von Punktwolken mit wenig räumlicher Struktur, die um zusätzliche Merkmale erweitert sind, können relativ zuverlässig registriert werden.

Somit kann die erste Fragestellung eindeutig bestätigt werden. Es ist möglich einen Algorithmus mit Hilfe von physikalischen Gesetzen so zu formulieren, dass dieser das Problem der Punktwolkenregistrierung löst. Folglich galt es in der zweiten Leitfrage herauszufinden, ob dieser neue Ansatz konkurrenzfähig zu bestehenden Standards ist.

Im Vergleich mit diversen Varianten des ICP bestätigt das neue Konzept größtenteils eine weitaus höhere Performanz. Die Experimente zur Genauigkeit und Reproduzierbarkeit zeigen, dass der Registrierungsfehler des neuen Algorithmus im RMSE-Median bis zu 70 % geringer ist als der der besten ICP-Varianten im Test. Obwohl das neue Verfahren im Gegensatz zum ICP nicht deterministisch ist, liegen sämtliche Fehlervarianzen bis zu über 50 % unter dem Fehler des genauesten ICP. Außerdem zeigt das neue Verfahren eine vergleichbar hohe Robustheit bei verrauschten oder fehlerbehafteten Punktwolken.

Die Laufzeit des neuen Registrierungsalgorithmus ist unabhängig von der Punktwolkengröße und demnach annähernd konstant. Das ist vor allem bei sehr hoch aufgelösten Punktwolken von Vorteil. In verschiedenen Experimenten konnte belegt werden, dass

selbst Wolken mit mehr als 300,000 Punkten effizienter registriert werden können als mit Standard ICP-Implementierungen. Dabei kann die Laufzeit des neuen Verfahrens bis auf 0.3 s reduziert werden. Die getesteten ICP-Varianten benötigen demgegenüber bereits ab 125,000 Punkten mehr als 15 s, jedoch unter der Voraussetzung, dass keine weiteren Vorverarbeitungsschritte vorgenommen werden. Ungeachtet des hohen Geschwindigkeitsvorteils bleibt die Genauigkeit der Registrierung erhalten und ist weitestgehend höher als die der getesteten ICPs.

Auch wenn die Testergebnisse größtenteils positiv ausfallen, offenbart die Evaluierung überdies Potenzial für zukünftige Erweiterungen und Verbesserungen. Beispielsweise sollte die Laufzeitkomplexität des Algorithmus einer gewissen Dynamik unterliegen. Eine Rechenzeit von konstant 0.3 s stellt für Punktwolken mit sehr hoher Auflösung ein hervorragendes Ergebnis dar, besonders da keine weiteren Vorverarbeitungsschritte und Vorregistrierungen notwendig sind. Doch bei niedrigen Punktzahlen ist diese Rechenzeit im Vergleich zu einigen ICP-Implementierungen zu hoch. In zukünftigen Modifikationen des Algorithmus könnte zum Beispiel eine dynamische Anpassung der Stichprobengröße vorgesehen werden, sodass sich diese automatisch und entsprechend der Auflösung anpasst.

Weisen Punktwolken nur wenige räumliche Strukturen auf, sind dafür aber um zusätzliche Merkmale erweitert, ist der RMSE Median ebenfalls um ca. 49 % geringer als der der anderen Verfahren im Test. Dabei schneidet die Metrik am besten ab, die dem Coulombschen Gesetz der Elektrostatik nachempfunden ist und sowohl anziehende als auch abstoßende Kräfte modelliert. Dieses Ergebnis lässt infolgedessen zwei Rückschlüsse zu: Erstens scheint es notwendig, dass sich Punktwolken im Registrierungsprozess nicht nur anziehen, sondern auch abstoßen. Das ermöglicht eine schnellere Konvergenz und vermeidet lokale Optima. Zweitens sollten weitere Metriken entwickelt und untersucht werden. Verglichen mit den anderen Experimenten in dieser Arbeit sind die Registrierungsfehler von rund 3.39 Einheiten bei einer Punktwolkenausdehnung von 100 Einheiten noch immer viel zu groß.

Vor allem für spezifische Anwendungsfälle könnten speziell angepasste Metriken entwickelt und analysiert werden. Die hohe Modularität des Algorithmus ermöglicht solche Erweiterungen oder Modifikationen mit wenig Aufwand. Zum Beispiel könnte der Einsatz differenzierter Regularisierungsverfahren untersucht werden. Obgleich das Simulated Annealing ein sehr gut abgestimmtes Verfahren für den neuen Algorithmus darstellt und es in den Experimenten herausragende Ergebnisse liefert, könnten andere Methoden die Genauigkeit, Reproduzierbarkeit oder Konvergenz und somit die Laufzeit des Algorithmus verbessern.

Abschließend kann gesagt werden, dass das neue Verfahren im Vergleich mit dem nahezu dreißigjährigen Entwicklungsvorsprung des ICP eine realistische Alternative in der Punktwolkenregistrierung darstellt. Das physikalisch motivierte Registrierungskonzept bietet eine reproduzierbar hohe Genauigkeit, läuft äußerst effizient bei sehr hoch aufgelösten Punktwolken und ist ausreichend robust bei verrauschten Punktwolkendaten.



## Literaturverzeichnis

- [1] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, sep 1987.
- [2] M. Attia, Y. Slama, and M. A. Kamoun. On performance evaluation of registration algorithms for 3D point clouds. In *2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGIV)*, pages 45–50. IEEE, mar 2016.
- [3] M. A. Audette, F. P. Ferrie, and T. M. Peters. An algorithmic overview of surface registration techniques for medical imaging. *Medical Image Analysis*, 4(3):201–217, sep 2000.
- [4] S. Bauer, J. Wasza, F. Lugauer, D. Neumann, and J. Hornegger. Real-time RGB-d mapping and 3-d modeling on the GPU using the random ball cover. In *Consumer Depth Cameras for Computer Vision*, pages 27–48. Springer London, 2013.
- [5] B. Bellekens, V. Spruyt, R. Berkvens, R. Penne, and M. Weyn. A benchmark survey of rigid 3D point cloud registration algorithms. *International Journal on Advances in Intelligent Systems*, 8(1&2):118–127, 2015.
- [6] B. Bellekens, V. Spruyt, and R. B. Maarten Weyn. A survey of rigid 3D pointcloud registration algorithms. In *Fourth International Conference on Ambient Computing, Applications, Services and Technologies, Proceedings*, pages 8–13. IARA, 2014.
- [7] F. Bellocchio, N. A. Borghese, S. Ferrari, and V. Piuri. *3D Surface Reconstruction*. Springer New York, 2013.
- [8] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multi-view registration technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):540–547, may 1996.
- [9] P. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, feb 1992.
- [10] P. Biber and W. Strasser. The normal distributions transform: a new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 3, pages 2743–2748 vol.3. IEEE, oct 2003.
- [11] S. R. Buss and J. P. Fillmore. Spherical averages and applications to spherical splines and interpolation. *ACM Transactions on Graphics*, 20(2):95–126, apr 2001.
- [12] V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, jan 1985.
- [13] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, apr 1992.
- [14] H. Chui and A. Rangarajan. A feature registration framework using mixture models. In *Proceedings IEEE Workshop on Mathematical Methods in Biomedical Image Analysis. MMBIA-2000 (Cat. No.PR00737)*, MMBIA '00, page 190, Washington, DC, USA, 2000. IEEE Comput. Soc.

- [15] J. L. B. Claraco. Development of scientific applications with the mobile robot programming toolkit. *Machine Perception and Intelligent Robotics Laboratory, University of Málaga. MRPT reference book, Málaga*, 2008.
- [16] W. K. Clifford. *Mathematical papers*. Macmillan and Company, 1882.
- [17] S. Cook. *CUDA Programming*. Elsevier LTD, Oxford, 2012.
- [18] J. Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 58(15-16):1–35, 2006.
- [19] Y. Díez, F. Roure, X. Lladó, and J. Salvi. A qualitative review on 3D coarse registration methods. *ACM Computing Surveys*, 47(3):1–36, feb 2015.
- [20] L. Douadi, M. Jose Aldon, and A. Crosnier. Pair-wise registration of 3D/color data sets with ICP. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 663–668. IEEE, oct 2006.
- [21] R. M. Dreizler and C. S. Lüdde. *Theoretical Mechanics*. Springer Berlin Heidelberg, 1 edition, 2010.
- [22] S. Druon, M. Aldon, and A. Crosnier. Color constrained ICP for registration of large unstructured 3d color data sets. In *2006 IEEE International Conference on Information Acquisition*, pages 249–255. IEEE, aug 2006.
- [23] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive Theory of Functions of Several Variables*, pages 85–100. Springer Berlin Heidelberg, 1977.
- [24] D. Eberly. Rotation representations and performance issues. *Magic Software, Inc., Chapel Hill, NC*, 2002.
- [25] B. Eckart, K. Kim, A. Troccoli, A. Kelly, and J. Kautz. MLMD: Maximum likelihood mixture decoupling for fast and accurate point cloud registration. In *2015 International Conference on 3D Vision*. IEEE, oct 2015.
- [26] I. Fischer. *Dual-number methods in kinematics, statics and dynamics*. CRC Press, 1998.
- [27] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, jun 1981.
- [28] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2012.
- [29] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness. New algorithms for 2D and 3D point matching. *Pattern Recognition*, 31(8):1019–1031, aug 1998.
- [30] V. Golyanik, S. A. Ali, and D. Stricker. Gravitational approach for point set registration. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016.
- [31] M. Greenspan, G. Godin, and J. Talbot. Acceleration of binning nearest neighbor methods. *Proceedings of Vision Interface 2000*, pages 337–344, 2000.
- [32] M. Greenspan and M. Yurick. Approximate k-d tree search for efficient ICP. In *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, pages 442–448. IEEE, IEEE, 2003.
- [33] F. Hartmann and A. Schlaefer. Feasibility of touch-less control of operating room lights. *International Journal of Computer Assisted Radiology and Surgery*, 8(2):259–268, jul 2012.

- [34] E. Heim, T. Kilgus, S. Haase, J. Iszatt, A. M. Franz, A. Seitel, M. Müller, M. Fangerau, J. Hornegger, H.-P. Meinzer, and L. Maier-Hein. GPGPU-beschleunigter anisotroper ICP zur registrierung von tiefendaten. In *Informatik aktuell*, pages 24–29. Springer Berlin Heidelberg, 2014.
- [35] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke. Registration with the point cloud library: A modular framework for aligning in 3-d. *IEEE Robotics & Automation Magazine*, 22(4):110–124, dec 2015.
- [36] L. Howes and D. Thomas. Efficient random number generation and application using cuda. *GPU gems*, 3:805–830, 2007.
- [37] J. Janai, F. Güney, A. Behl, and A. Geiger. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *Arxiv*, 2017.
- [38] P. Jauer, I. Kuhlemann, R. Bruder, A. Schweikard, and F. Ernst. Efficient registration of high-resolution feature enhanced point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, submitted dec 2016 (under review).
- [39] P. Jauer, I. Kuhlemann, F. Ernst, and A. Schweikard. GPU-based real-time 3D workspace generation of arbitrary serial manipulators. In *2016 2nd International Conference on Control, Automation and Robotics (ICCAR)*, pages 56–61. IEEE, apr 2016.
- [40] A. E. Johnson and S. B. Kang. Registration and integration of textured 3D data. *Image and Vision Computing*, 17(2):135–147, feb 1999.
- [41] J. H. Joung, K. H. An, J. W. Kang, M. J. Chung, and W. Yu. 3D environment reconstruction using modified color ICP algorithm by fusion of a camera and a 3D laser range finder. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3082–3088. IEEE, oct 2009.
- [42] B. Jüttler. Visualization of moving objects using dual quaternion curves. *Computers & Graphics*, 18(3):315–326, may 1994.
- [43] L. Kavan, S. Collins, C. O’Sullivan, and J. Zara. Dual quaternions for rigid transformation blending. *Trinity College Dublin, Tech. Rep. TCD-CS-2006-46*, 2006.
- [44] L. Kavan and J. Žára. Spherical blend skinning. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games - SI3D '05, I3D '05*, pages 9–16, New York, NY, USA, 2005. ACM Press.
- [45] B. Kenwright. A beginners guide to dual-quaternions. In *WSCG Communication proceedings*, pages 1–10. Union Agency, 2012.
- [46] B. Kenwright. *Dual-Quaternions: From Classical Mechanics to Computer Graphics and Beyond*, oct 2012.
- [47] H. Kim and A. Hilton. Influence of colour and feature geometry on multi-modal 3D point clouds data registration. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 202–209. IEEE, dec 2014.
- [48] D. B. Kirk and W. mei W. Hwu. *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann, 2010.
- [49] S. Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 34(5-6):975–986, mar 1984.
- [50] H. M. Kjer and J. Wilm. Evaluation of surface registration algorithms for pet motion correction. Bachelor’s Thesis, Technical University of Denmark, 2010.

- [51] M. Kjer and J. Wilm. *Iterative Closest Point*. Technical University of Denmark, <https://mathworks.com/matlabcentral/fileexchange/27804-iterative-closest-point>, 2012.
- [52] D. E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley Pub, 2 edition, 1981.
- [53] S. Köhnen, J. Ehrhardt, A. Schmidt-Richberg, and H. Handels. CUDA optimierung von nicht-linearer oberflächen- und intensitätsbasierter registrierung. In *Bildverarbeitung für die Medizin 2011*, pages 99–103. Springer Berlin Heidelberg, 2011.
- [54] M. Korn, M. Holzkothen, and J. Pauli. Color supported generalized-ICP. In *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 3, pages 592–599, jan 2014.
- [55] S. Krig. *Computer Vision Metrics*. Apress, 2014.
- [56] D. P. Landau and K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge University Press, Cambridge, 3 edition, 009 2009.
- [57] D. Lee, H. Kim, and H. Myung. GPU-based real-time RGB-d 3D SLAM. In *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. IEEE, nov 2012.
- [58] M. Levoy, J. Gerth, B. Curless, and K. Pull. *The Stanford 3D scanning repository*. <http://www-graphics.stanford.edu/data/3dscanrep>, 2005.
- [59] K.-L. Low. Linear least-squares optimization for point-to-plane ICP surface registration. *Chapel Hill, University of North Carolina*, 4, 2004.
- [60] M. Magnusson, A. Lilienthal, and T. Duckett. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10):803–827, 2007.
- [61] M. Magnusson, N. Vaskevicius, T. Stoyanov, K. Pathak, and A. Birk. Beyond points: Evaluating recent 3D scan-matching algorithms. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3631–3637. IEEE, may 2015.
- [62] S. Marsland. *Machine learning: an algorithmic perspective*. CRC press, 2011.
- [63] The MathWorks, Inc., Natick, Massachusetts, USA. *Computer Vision System Toolbox Reference (r2016a)*, revised for version 7.1 (release r2016a) edition, Mar. 2016.
- [64] H. Men, B. Gebre, and K. Pochiraju. Color point cloud registration with 4D ICP algorithm. In *2011 IEEE International Conference on Robotics and Automation*, pages 1511–1516. IEEE, may 2011.
- [65] H. Men and K. Pochiraju. Hue-assisted automatic registration of color point clouds. *Journal of Computational Design and Engineering*, 1(4):223–232, oct 2014.
- [66] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, dec 2010.
- [67] A. Myronenko, X. Song, and M. Á. Carreira-Perpiñán. Non-rigid point set registration: Coherent point drift. In P. B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1009–1016. MIT Press, 2007.
- [68] J. Neumann and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.
- [69] H. Nguyen. *GPU Gems 3*. Addison-Wesley Professional, 2007.

- [70] W. Nolting. *Grundkurs Theoretische Physik 1*. Springer Berlin Heidelberg, 10 edition, 2013.
- [71] A. Nuchter, K. Lingemann, and J. Hertzberg. Cached k-d tree search for ICP algorithms. In *Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007)*, pages 419–426. IEEE, aug 2007.
- [72] A. Nuchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6D SLAM with approximate data association. In *ICAR '05. Proceedings., 12th International Conference on Advanced Robotics, 2005.*, pages 242–249. IEEE, jul 2005.
- [73] NVIDIA. *CUDA RUNTIME API*, 2016.
- [74] NVIDIA. *CUDA C Programming Guide*, 2017.
- [75] S.-Y. Park and M. Subbarao. An accurate and fast point-to-plane registration technique. *Pattern Recognition Letters*, 24(16):2967–2976, dec 2003.
- [76] E. Pennestri and R. Stefanelli. Linear algebra and numerical algorithms using dual numbers. *Multibody System Dynamics*, 18(3):323–344, aug 2007.
- [77] E. Pennestri and P. Valentini. Dual quaternions as a tool for rigid body motion analysis: A tutorial with an application to biomechanics. *Archive of Mechanical Engineering*, LVII(2):187–205, jan 2010.
- [78] N. Ploskas and N. Samaras. *GPU Programming in MATLAB*. Elsevier LTD, Oxford, 2016.
- [79] F. Pomerleau, F. Colas, and R. Siegwart. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends in Robotics*, 4(1):1–104, may 2015.
- [80] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, feb 2013.
- [81] F. Pomerleau, S. Magnenat, F. Colas, M. Liu, and R. Siegwart. Tracking a depth camera: Parameter exploration for fast ICP. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3824–3829. IEEE, IEEE, sep 2011.
- [82] H. Pottmann, Q.-X. Huang, Y.-L. Yang, and S.-M. Hu. Geometry and convergence analysis of algorithms for registration of 3D shapes. *International Journal of Computer Vision*, 67(3):277–296, mar 2006.
- [83] K. Pulli. Multiview registration for large data sets. In *Second International Conference on 3-D Digital Imaging and Modeling (Cat. No.PR00062)*, pages 160–168. IEEE, IEEE Comput. Soc, 1999.
- [84] D. Qiu, S. May, and A. Nüchter. GPU-accelerated nearest neighbor search for 3D registration. In *Lecture Notes in Computer Science*, pages 194–203. Springer Berlin Heidelberg, 2009.
- [85] A. Rangarajan, H. Chui, E. Mjolsness, S. Pappu, L. Davachi, P. Goldman-Rakic, and J. Duncan. A robust point-matching algorithm for autoradiograph alignment. *Medical Image Analysis*, 1(4):379–398, sep 1997.
- [86] J. Razlaw, D. Droschel, D. Holz, and S. Behnke. Evaluation of registration methods for sparse 3D laser scans. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–7. IEEE, IEEE, sep 2015.
- [87] M. Rezaei and R. Klette. *Computer Vision for Driver Assistance*. Springer International Publishing, 2017.
- [88] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152. IEEE Comput. Soc, 2001.

- [89] R. B. Rusu and S. Cousins. 3D is here: Point cloud library (PCL). In *2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, may 2011. IEEE.
- [90] J. Salvi, C. Matabosch, D. Fofi, and J. Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25(5):578–596, may 2007.
- [91] J. Santamaría, O. Córdón, and S. Damas. A comparative study of state-of-the-art evolutionary image registration methods for 3D modeling. *Computer Vision and Image Understanding*, 115(9):1340–1354, sep 2011.
- [92] P. Schneider and D. H. Eberly. *Geometric tools for computer graphics*. Morgan Kaufmann, 2002.
- [93] A. Schweikard and F. Ernst. *Medical Robotics*. Springer International Publishing, 2015.
- [94] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Robotics: Science and Systems*, volume 2, 2009.
- [95] K. Shoemake. Animating rotation with quaternion curves. *ACM SIGGRAPH Computer Graphics*, 19(3):245–254, jul 1985.
- [96] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot modeling and control*, volume 3. Wiley New York, 2006.
- [97] T. Taketomi, H. Uchiyama, and S. Ikeda. Visual SLAM algorithms: a survey from 2010 to 2016. *IPSV Transactions on Computer Vision and Applications*, 9(1), jun 2017.
- [98] G. K. L. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1199–1217, jul 2013.
- [99] T. Tamaki, M. Abe, B. Raychev, and K. Kaneda. Softassign and EM-ICP on GPU. In *2010 First International Conference on Networking and Computing*. IEEE, nov 2010.
- [100] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In T. Pajdla and J. Matas, editors, *Lecture Notes in Computer Science*, pages 558–569. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [101] G. Turk and B. Mullins. *Large Geometric Models Archive*. Georgia Institute of Technology, [http://www.cc.gatech.edu/projects/large\\_models/](http://www.cc.gatech.edu/projects/large_models/), 2007.
- [102] T. Wissel, P. Stüber, B. Wagner, R. Bruder, C. Erdmann, C.-S. Deutz, B. Sack, J. Mani, A. Schweikard, and F. Ernst. Enhanced optical head tracking for cranial radiation therapy: Supporting surface registration by cutaneous structures. *International Journal of Radiation Oncology\* Biology\* Physics*, 95(2):810–817, jun 2016.
- [103] P. Zanuttigh, G. Marin, C. D. Mutto, F. Dominio, L. Minto, and G. M. Cortelazzo. 3d scene reconstruction from depth camera data. In *Time-of-Flight and Structured Light Depth Cameras*, pages 231–251. Springer International Publishing, Cham, 2016.
- [104] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, oct 1994.
- [105] T. Zinßer, J. Schmidt, and H. Niemann. Performance analysis of nearest neighbor algorithms for ICP registration of 3-d point sets. In *VMV*, pages 199–206, 2003.

## Abbildungsverzeichnis

2.1	Exemplarische Darstellung einer Punktwolke verglichen mit einer Farbaufnahme des gleichen Objektes. . . . .	9
2.2	Flussdiagramm des Iterative-Closest-Point-Algorithmus. . . . .	16
2.3	Exemplarische Konstruktion eines $k$ D-Baums. . . . .	18
2.4	2D-Beispiel der Punkt-zu-Punkt-Variante des ICP. . . . .	20
2.5	2D-Beispiel der Punkt-zu-Ebene-Variante des ICP. . . . .	22
3.1	Flussdiagramm zum neuen Registrierungsalgorithmus. . . . .	28
3.2	Das Flussdiagramm des neuen Registrierungsalgorithmus erweitert um zwei neue Bestandteile. . . . .	36
3.3	2D Minimalbeispiel, das die Notwendigkeit von Merkmalerweiterungen bei der Punktwolkenregistrierung demonstriert. . . . .	41
3.4	Das Flussdiagramm des neuen Registrierungsalgorithmus erweitert um die Kraftfeldmodellierung. . . . .	43
3.5	Das Flussdiagramm des neuen Registrierungsalgorithmus erweitert um einen Regularisierer. . . . .	51
3.6	Das Flussdiagramm des neuen Registrierungsalgorithmus erweitert um das Monte-Carlo-Verfahren. . . . .	60
3.7	Das abschließende Flussdiagramm des neuen Registrierungsalgorithmus erweitert um die Bewegungsbeschreibung durch Duale Quaternionen. . . . .	70
4.1	Dragon-Datenansatz aus Experiment A: Genauigkeit und Reproduzierbarkeit. . . . .	76
4.2	Sehr hoch aufgelöste Punktwolken mit jeweils 327,323 Punkten. . . . .	80
4.3	Ergebnisse zum Effektivitätsexperiment B: Effizienz. . . . .	82
4.4	Utah-Teekanne ohne und mit künstlichem normalverteilten Rauschen. . . . .	84
4.5	Ergebnisse zum Experiment C: Robustheit, mit Erwartungswert. . . . .	86
4.6	Ergebnisse zum Experiment C: Robustheit, ohne Erwartungswert. . . . .	86
4.7	Synthetisch generierte Datensatz für das Experiment D: Merkmalerweiterung. . . . .	88

4.8	Die Ergebnisse zur Wiederholung des Experiments C: Robustheit mit unterschiedlichen Transformationsbeschreibungen. . . . .	97
4.9	Die Ergebnisse zur Komplexitätsuntersuchung analog zu Experiment A: Genauigkeit und Reproduzierbarkeit. . . . .	99
4.10	Ergebnisse zur Analyse des Registrierungsfehlers RMSE in Abhängigkeit der frei wählbaren Parameter Abkühlungskonstante $c$ und Abbruchkriterium $\varepsilon$ . . . . .	103
4.11	Ergebnisse zur Analyse der Laufzeit in Abhängigkeit der frei wählbaren Parameter Abkühlungskonstante $c$ und Abbruchkriterium $\varepsilon$ . . . . .	104

## Tabellenverzeichnis

4.1	Die Ergebnisse zum Experiment A: Genauigkeit und Reproduzierbarkeit. . .	78
4.2	Die Ergebnisse zum Experiment D: Merkmalerweiterung. . . . .	89
4.3	Die Ergebnisse zur Wiederholung des Experiments A: Genauigkeit und Reproduzierbarkeit mit unterschiedlichen Transformationsbeschreibungen.	94
4.4	Die Ergebnisse zur Evaluierung der Schrittweisenmodellierung analog zu Experiment A: Genauigkeit und Reproduzierbarkeit. . . . .	102



## Abkürzungsverzeichnis

<b>1D</b>	Eindimensional
<b>2D</b>	Zweidimensional
<b>3D</b>	Dreidimensional
<b>Bbr</b>	RMSE Bandbreite
<b>CPU</b>	Central Processing Unit
<b>CUDA</b>	Compute Unified Device Architecture
<b>DQ</b>	Physikalisch motivierter Algorithmus zur Punktwolkenregistrierung mit der Metrik analog zum Newtonschen Gravitationsgesetz; Transformationen sind durch Duale Quaternionen repräsentiert, mit: <b>DQ1</b> : nur eine Transformation pro Iteration <b>DQ2</b> : zwei Transformationen pro Iteration werden gemittelt <b>DQ3 bis DQ8</b> : drei bis acht Transformationen pro Iteration werden berechnet, die schlechteste gefiltert und der Rest gemittelt
<b>GPU</b>	Graphics Processing Unit
<b>ICP</b>	Iterative Closest Point
<b>IFE</b>	Punkt-zu-Ebene-Implementierung des ICP von Kjer und Wilm [50, 51] mit der Merkmalerweiterung nach [40, 54]
<b>IFP</b>	Punkt-zu-Punkt-Implementierung des ICP von Kjer und Wilm [50, 51] mit der Merkmalerweiterung nach [40, 54]
<b>IKE</b>	Punkt-zu-Ebene-Implementierung des ICP von Kjer und Wilm [50, 51]
<b>IKP</b>	Punkt-zu-Punkt-Implementierung des ICP von Kjer und Wilm [50, 51]
<b>IME</b>	Punkt-zu-Ebene-Implementierung des ICP aus MATLAB R2016a [63]

<b>IMP</b>	Punkt-zu-Punkt-Implementierung des ICP aus MATLAB R2016a [63]
<b>IPE</b>	Punkt-zu-Ebene-Implementierung des ICP aus der PCL [89]
<b>IPP</b>	Punkt-zu-Ebene-Implementierung des ICP aus der PCL [89]
<b>IQA</b>	Interquartilabstand
<b>KCO</b>	Physikalisch motivierter Algorithmus zur Punktwolkenregistrierung mit der Metrik analog zum Coulombschen Gesetz der Elektrostatik, bei dem lediglich Anziehungskräfte zwischen den Punktwolken anhand von Merkmalsübereinstimmungen modelliert werden
<b>KCM</b>	Physikalisch motivierter Algorithmus zur Punktwolkenregistrierung mit der Metrik analog zum Coulombschen Gesetz der Elektrostatik, bei dem Anziehungs- und Abstoßungskräfte zwischen den Punktwolken anhand von Merkmalsübereinstimmungen modelliert werden
<b>KNG</b>	Physikalisch motivierter Algorithmus zur Punktwolkenregistrierung mit der Metrik analog zum Newtonschen Gravitationsgesetz
<b>LCG</b>	Linear Congruential Generator
<b>Med</b>	RMSE Median
<b>PCL</b>	Point Cloud Library [89]
<b>RMSE</b>	Root Mean Square Error

## Danksagung

An erster Stelle möchte ich mich bei meiner Frau Rahel bedanken. Durch ihre uneingeschränkte Unterstützung, besonders in stressigen und schwierigen Zeiten, gab sie mir stets unverzichtbaren Rückhalt. Auch meinen Eltern Corina und Frank sowie meinen Großeltern Renate und Karl gilt besonderer Dank. Zeit meines Lebens waren und sind sie eine wichtige Säule, auf die ich mich immer verlassen kann. Diese Arbeit wäre nicht ohne die Unterstützung meiner Familie entstanden.

Zudem möchte ich meinem Doktorvater Herrn Prof. Dr.-Ing. Achim Schweikard danken, dass er mir die Möglichkeit gab, an diesem interessanten Thema zu arbeiten und mich dabei unterstützt hat, meine Ideen frei zu verfolgen. Gleichmaßen möchte ich mich bei meinen Kollegen bedanken, mit denen ich in den letzten sechs Jahren am Institut für Robotik und Kognitive Systeme arbeiten konnte. Die konstruktiven Kritiken und fachlichen Diskussionen sowie das hilfreiche Korrigieren haben einen erheblichen Teil zum Gelingen der Doktorarbeit beigetragen. Es war und ist mir eine Ehre in diesem aufgeschlossenen und kreativen Team zu arbeiten.

Zu guter Letzt danke ich meinen Freunden für die gelegentliche Ablenkung, um den Kopf von der Arbeit frei zu bekommen. Denn „tritt eine Idee in ein hohlen Kopf, so füllt sie ihn völlig aus – weil keine andere da ist, die ihr den Rang streitig machen könnte“ (Charles-Louis de Montesquieu).



## Lebenslauf



### Persönliche Daten

Name Philipp Jauer  
Geburtsdatum 20. September 1984  
Geburtsort Wilhelm-Pieck-Stadt Guben

### Ausbildung

2010–2012 **Master of Science**, Universität zu Lübeck, Studium der Informatik mit dem Anwendungsfach Robotik und Automation  
*Thema der Masterarbeit: Multimodaler transkranieller 4D-Ultraschallatlas*

2006–2010 **Bachelor of Science**, Universität zu Lübeck, Studium der Informatik mit dem Anwendungsfach Robotik und Automation  
*Thema der Bachelorarbeit: Fusion statischer CT- und 4D-Ultraschall-Aufnahmen*

2005 **Allgemeine Hochschulreife**, Europaschule „Marie & Pierre Curie“ Guben